

UNCLASSIFIED
T.O. 31P2-2FSQ7-112

THEORY OF PROGRAMMING

FOR

AN/FSQ-7

COMBAT DIRECTION CENTRAL

15 November 1956

The work reported in this document was performed under a government contract; information contained herein is of a proprietary nature. ALL INFORMATION CONTAINED HEREIN SHALL BE KEPT IN CONFIDENCE. No information shall be divulged to persons other than IBM employees authorized by the nature of their duties to receive such information or individuals or organizations who are authorized in writing by the Department of Engineering or its appointee to receive such information. GOVERNMENT RELEASE MUST BE OBTAINED THROUGH THE IBM PATENT DEPARTMENT BEFORE THIS INFORMATION MAY BE USED FOR COMMERCIAL APPLICATIONS.

MILITARY PRODUCTS DIVISION

INTERNATIONAL BUSINESS MACHINES CORPORATION

KINGSTON, NEW YORK

UNCLASSIFIED

COPY NO.

UNCLASSIFIED
T.O. 31P2-2FSQ7-112

Reproduction for non-military use of the information or illustrations contained in this publication is not permitted without specific approval of the issuing service (BuAer or USAF). The policy for use of classified publications is established for the Air Force in AFR 205-1 and for the Navy in Navy Regulations, Article 1509.

—LIST OF REVISED PAGES—

INSERT LATEST REVISED PAGES. DESTROY SUPERSEDED PAGES.

NOTE: The portion of the text affected by the current revision is indicated by a vertical rule in the left margin of a left-hand page and in the right margin of a right-hand page.

*The asterisk indicates pages revised, added or deleted by the current revision.

CONTENTS

<i>Heading</i>	<i>Page</i>
PART 1 EQUIPMENT INTRODUCTION	1
CHAPTER 1 INTRODUCTION	1
SECTION 1 SCOPE AND CONTENTS	1
SECTION 2 DIGITAL COMPUTERS	2
2.1 Advantages	2
2.2 Design Requirements	2
2.2.1 Basic Operations	2
2.2.2 Coded Instructions	2
2.2.3 Internal Storage	2
2.3 Programming	2
2.3.1 Introduction	2
2.3.2 Program Definition	3
2.3.3 Examples	3
2.3.4 Iterative Programming	3
2.4 Needs of Programmer	3
SECTION 3 AN/FSQ-7 COMBAT DIRECTION CENTRAL	5
3.1 Purpose of Equipment	5
3.1.1 Air Defense Problem	5
3.1.2 Iterative Air Defense Program	5
3.1.3 Sample Program Routine	5
3.2 Equipment Functions	6
3.2.1 Component Systems	6
3.2.2 Central Computer System	6
3.2.2.1 Logical Divisions	6
3.2.2.2 Element Functions	6
3.2.2.3 System Information Flow	7
3.2.3 Input System	8
3.2.4 Drum System	8
3.2.5 Display System	10

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.2.6 Output System	10
3.2.7 Warning Light System	11
3.3 Duplexing	11
CHAPTER 2 NUMBER SYSTEMS	13
SECTION 1 GENERAL	13
1.1 The Concept of Number	13
1.2 Decimal Number System	13
1.2.1 General Properties	13
1.2.2 Positional Notation of Magnitude	13
1.2.3 Radix	14
1.2.4 Modulus	14
1.3 General Number System	14
SECTION 2 BINARY NUMBER SYSTEM	15
2.1 Introduction	15
2.2 Conversion from Decimal to Binary Notation	15
2.2.1 Conversion by Definition	15
2.2.2 Systematic Conversion	15
2.2.2.1 Integral Decimal Numbers	15
2.2.2.2 Fractional Decimal Numbers	16
2.2.2.3 Mixed Decimal Numbers	16
2.2.2.4 Justification	16
2.3 Conversion from Binary to Decimal Notation	16
SECTION 3 BINARY ARITHMETIC	17
3.1 General	17
3.2 Arbitrary Limitations	17
3.2.1 Design Limitations	17
3.2.2 Sign Convention	17
3.2.3 Fixed Binary Point	17
3.3 Binary Addition	18
3.4 Binary Subtraction	18
3.4.1 Direct Subtraction	18
3.4.2 Complements	18

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.4.3 Application of Complements	19
3.4.3.1 Interpretation	19
3.4.3.2 Generation	19
3.4.3.3 Subtraction Using 2's Complement	19
3.4.3.4 Subtraction Using 1's Complement	20
3.4.3.5 Uses of 2's and 1's Complements	20
3.4.4 Zero	20
3.4.5 Overflow	20
3.5 Binary Multiplication	21
3.5.1 Direct Multiplication	21
3.5.2 Multiplication by Addition and Shifting	21
3.5.3 Shifting	22
3.5.4 Round-Off	22
3.6 Binary Division	22
3.6.1 Principles	22
3.6.2 Direct Division	22
3.6.3 Non-Restoring Method	23
3.7 Number Conversion Using Binary Arithmetic	24
SECTION 4 OCTONARY NUMBER SYSTEM	25
CHAPTER 3 CENTRAL COMPUTER SYSTEM	27
SECTION 1 INTRODUCTION	27
1.1 System Characteristics	27
1.2 Description of Characteristics	27
1.2.1 Information Form	27
1.2.2 Handling Methods	27
1.2.2.1 Parallel Operation	27
1.2.2.2 Dual Operation	27
1.2.2.3 Computer Word	27
1.2.3 Internal Memory	28
1.2.4 Instructions	28
1.2.4.1 Significance	28
1.2.4.2 Instruction Function	28
1.2.4.3 Single-Address Instruction	28

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
1.2.5 Programs	29
1.2.6 Real-Time Operation	29
1.2.7 General Applicability	29
1.3 Duplexing	29
SECTION 2 INTERNAL MEMORY OPERATION	30
2.1 Core Memory	30
2.1.1 Introduction	30
2.1.2 Principles of Operation	30
2.1.3 Coincident Current Selection	30
2.1.3.1 Core Array	30
2.1.3.2 Word Storage	30
2.1.3.3 Word Reading	32
2.1.4 Register Addressing	32
2.1.5 Function of Memory Buffer Register	33
2.1.6 Memory Cycles	34
2.1.6.1 Timing	34
2.1.6.2 Read Cycle	34
2.1.6.3 Write Cycle	34
2.2 Test Memory	35
2.3 Real-Time Clock	35
2.4 Memory Addresses	36
SECTION 3 SYSTEM OPERATION	37
3.1 Introduction	37
3.1.1 General	37
3.1.2 Types of Operations	37
3.1.3 Timing	37
3.1.3.1 Machine Cycles	37
3.1.3.2 Cycle Timing	37
3.1.3.3 Instruction Cycles	39
3.1.4 Operating Units	40
3.1.4.1 Functional Grouping	40
3.1.4.2 Computing Registers	40
3.1.4.3 Internal Control Registers	41

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.1.4.4 IO Control Registers	42
3.2 Instructions	43
3.2.1 Classification	43
3.2.2 Decoding	43
3.2.3 Indexing	45
3.3 Add Class Instructions	45
3.3.1 Principles of Execution	45
3.3.2 Typical Instruction	48
3.3.3 Variations	48
3.3.4 Overflow Control	49
3.4 Multiply Class Instructions	49
3.4.1 General	49
3.4.2 <i>Multiply</i>	49
3.4.3 <i>Divide</i>	51
3.4.4 Variations	53
3.5 Store Class Instructions	53
3.5.1 Principles of Execution	53
3.5.2 Typical Instruction	53
3.5.3 Similar Variations	53
3.5.4 Different Variations	54
3.5.4.1 <i>Add One</i>	54
3.5.4.2 <i>Exchange</i>	55
3.5.4.3 <i>Deposit</i>	55
3.6 Shift Class Instructions	56
3.6.1 Principles of Execution	56
3.6.2 Typical Instruction	58
3.6.3 Shift Variations	58
3.6.4 Cycle Variations	58
3.7 Branch Class Instructions	59
3.7.1 Principles of Execution	59
3.7.2 Typical Instruction	60
3.7.3 Similar Variations	61
3.7.4 Different Variations	61
3.7.4.1 <i>Branch on Zero</i>	61

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.7.4.2 <i>Sense</i>	61
3.7.4.3 <i>Branch and Index</i>	63
3.8 <i>IO Class Instructions</i>	63
3.8.1 <i>General</i>	63
3.8.2 <i>Variations</i>	65
3.9 <i>Reset Class Instructions</i>	67
3.9.1 <i>General</i>	67
3.9.2 <i>Variations</i>	67
3.10 <i>Miscellaneous Class Instructions</i>	67
3.10.1 <i>General</i>	67
3.10.2 <i>Program Stop</i>	67
3.10.3 <i>Extract</i>	67
3.10.4 <i>Operate</i>	68
3.10.5 <i>Clear and Subtract Word Counter</i>	68
3.10.6 <i>Shift Left and Round</i>	68
3.10.7 <i>Load B Registers</i>	70
SECTION 4 INCLUDED IO UNITS	71
4.1 <i>Introduction</i>	71
4.2 <i>Punched Cards and Card Forms</i>	71
4.2.1 <i>General</i>	71
4.2.2 <i>Instruction Card</i>	72
4.2.3 <i>Binary Card</i>	73
4.2.4 <i>Card Image</i>	74
4.3 <i>Card Reader</i>	77
4.3.1 <i>Operation</i>	77
4.3.2 <i>Controls and Indicators</i>	77
4.3.3 <i>Information Transfer</i>	78
4.4 <i>Card Punch</i>	79
4.4.1 <i>Operation</i>	79
4.4.2 <i>Controls and Indicators</i>	79
4.4.3 <i>Information Transfer</i>	79
4.5 <i>Line Printer</i>	82
4.5.1 <i>Operation</i>	82

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
4.5.2 Controls and Indicators	84
4.5.2.1 Control Panel	84
4.5.2.2 Manual Controls and Indicators	85
4.5.3 Information Transfer	86
4.6 Tape Element	87
4.6.1 Introduction	87
4.6.2 Tape Drive Unit	89
4.6.2.1 Manual Operations	89
4.6.2.2 Programmed Operation	90
4.6.3 Tape Adapter Unit	91
4.6.4 Programming	94
4.6.4.1 Pertinent Instructions	94
4.6.4.2 General Rules	95
4.6.4.3 Read Operation	95
4.6.4.4 Write Operation	96
4.6.4.5 Backspace	97
4.6.4.6 Rewind	97
4.7 IO Register	98
SECTION 5 APPLICATIONS	100
5.1 Introduction	100
5.2 Straight-Line Programming	100
5.2.1 Basic Arithmetic Operations	100
5.2.1.1 Addition	100
5.2.1.2 Subtraction	100
5.2.1.3 Multiplication	101
5.2.1.4 Division	102
5.2.2 Combined Operations	103
5.2.2.1 Co-ordinate Conversion	103
5.2.2.2 Evaluation of a Function	104
5.2.3 Logical Operations	105
5.2.4 IO Operations	105
5.3 Iterative Programming	107
5.3.1 General	107

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
5.3.2 Indexing	107
5.3.3 Program Comparisons	108
5.3.3.1 Straight-Line Program	108
5.3.3.2 Non-indexed Iterative Program	108
5.3.3.3 Indexed Iterative Program	110
5.3.4 Applications of Indexing	111
5.3.4.1 Indexed Number-Sorting Program	111
5.3.4.2 Indexed Function Evaluation Program	113
5.3.5 Program Precautions in Indexing	115
5.3.5.1 Number of Cycles of Iterated Routine	115
5.3.5.2 Use of Zero Address with Indexable Instructions	116
5.4 Program Preparation	117
5.4.1 Program Organization	117
5.4.1.1 General	117
5.4.1.2 Master Programs	117
5.4.1.3 Subroutines	117
5.4.1.4 Compiler Programs	118
5.4.1.5 Utility Programs	118
5.4.1.6 Parameters	119
5.4.2 Program Coding	119
5.4.2.1 Absolute Programming	119
5.4.2.2 Symbolic Programming	120
5.4.2.3 Assembly Program	120
5.4.2.4 Assembly Program Card Punching	120
5.4.2.5 Printout of Assembled Program	123
5.4.3 Program Testing	123
5.4.3.1 General	123
5.4.3.2 Program Monitoring	123
5.4.3.3 Test Programs	124
5.4.3.4 Simulation Programs	124
CHAPTER 4 DRUM SYSTEM	125
SECTION 1 SYSTEM DESCRIPTION	125
1.1 System Function	125

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
1.2 System Data	126
1.2.1 Physical Description	126
1.2.2 Logical Description	126
1.2.3 Logical Definitions	127
1.2.4 Timing	128
1.3 System Logic	129
SECTION 2 SYSTEM OPERATION	132
2.1 General	132
2.2 Time Buffer Operation	137
2.2.1 Writing	137
2.2.1.1 Status	138
2.2.1.2 Modified Status	138
2.2.1.3 Marker Status	139
2.2.2 Reading	139
2.2.2.1 Status Identification	139
2.2.2.2 Status	140
2.3 Auxiliary Memory Operation	140
2.3.1 Writing	140
2.3.1.1 Address	140
2.3.1.2 Interleave	141
2.3.2 Reading	142
2.3.2.1 Address	142
2.3.2.2 Interleave	143
2.3.2.3 Precession	143
SECTION 3 INFORMATION FLOW	145
3.1 OD Information Transfers	145
3.1.1 Input System Information	145
3.1.2 Output System Information	145
3.1.3 Display System Information	145
3.2 CD Information Transfers	147
3.2.1 Auxiliary Memory Information	147
3.2.2 Input Information	148
3.2.3 Output Information	149

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.2.4 Intercommunication Information	151
3.3 Test Operations	152
3.3.1 General	152
3.3.2 Parity Check	153
3.3.3 Computer Tests	155
3.3.4 Manual Drum Tests	156
CHAPTER 5 INPUT SYSTEM	159
SECTION 1 LONG-RANGE RADAR INPUT ELEMENT	159
1.1 Introduction	159
1.2 LRI Element Description	160
1.2.1 Element Function	160
1.2.2 Element Logic	160
1.2.3 LRI Data	161
1.2.3.1 Data Sources	161
1.2.3.2 Data Handling	162
1.2.3.3 Data Forms	163
1.3 Element Operation	164
1.3.1 General	164
1.3.2 Read-In	164
1.3.3 Checks	167
1.3.4 Transfer to Drum Field	167
1.3.5 Alarm Conditions	168
1.3.6 LRI Monitor	169
1.4 Processed-Data Forms	169
1.5 Duplexing	170
SECTION 2 GAP-FILLER INPUT ELEMENT	173
2.1 Element Description	173
2.1.1 Element Function	173
2.1.2 Element Logic	173
2.1.3 Data Source	173
2.2 Element Operation	175
2.2.1 General	175

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.2.2 Data Conversion Receiver	175
2.2.3 Mapper Section	175
2.2.4 Counter Section	177
2.2.5 GFI Common Section	177
2.2.6 GFI Monitor	177
2.3 Information Form	178
2.4 Duplexing	178
SECTION 3 CROSSTELL INPUT ELEMENT	181
3.1 Element Description	181
3.1.1 Element Function	181
3.1.2 Element Logic	181
3.1.3 Data Form	182
3.2 Element Operation	182
3.2.1 General	182
3.2.2 Read-In	182
3.2.3 Checks	183
3.2.4 Transfer to Drum Field	184
3.2.5 Alarm Conditions	186
3.3 Information Form	186
3.4 XTL Duplexing	188
SECTION 4 MANUAL DATA INPUT ELEMENT	189
4.1 Drum Entry Section	189
4.1.1 Introduction	189
4.1.2 Section Description	189
4.1.2.1 Function	189
4.1.2.2 Computer Entry Punch Data Sources	190
4.1.2.3 Display Data Sources	191
4.1.3 Section Operation	191
4.1.3.1 General	191
4.1.3.2 Computer Entry Punch Information Transfer	193
4.1.3.3 Light Gun Information Transfer	193
4.1.3.4 Area Discriminator Information Transfer	193
4.1.4 Information Forms	194

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
4.1.5 Duplexing	194
4.2 Direct IO Buffer Entry Section	195
4.2.1 Section Description	195
4.2.1.1 Function	195
4.2.1.2 Keyboard Messages	196
4.2.2 Section Operation	197
4.2.2.1 Message Insertion	197
4.2.2.2 Core Matrix	198
4.2.2.3 Transfer to Central Computer System	198
4.2.3 Duplexing	199
CHAPTER 6 DISPLAY SYSTEM	201
SECTION 1 SYSTEM DESCRIPTION	201
1.1 System Function	201
1.2 System Logic	201
1.2.1 Overall	201
1.2.2 Duplexing	203
1.3 Display Tubes	204
1.3.1 General	204
1.3.2 Situation Display Tube	204
1.3.3 Digital Display Tube	206
SECTION 2 SITUATION DISPLAY SUBSYSTEM OPERATION	208
2.1 Introduction	208
2.2 Situation Display Messages	208
2.2.1 Radar Data Messages	208
2.2.1.1 Information Contents	208
2.2.1.2 Effects of Programming on Display	209
2.2.1.3 Radar Data Message Drum Layout	210
2.2.2 Track Data Messages	210
2.2.2.1 General	210
2.2.2.2 Tabular Messages	211
2.2.2.3 Tabular Message Drum Layout	214
2.2.2.4 Vector Messages	214

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.2.2.5 Vector Message Drum Layout	215
2.3 Situation Display Generator Element	217
2.3.1 Element Function	217
2.3.2 Situation Display Cycle	218
2.3.3 Element Operation	218
2.4 Situation Display Indicator Element	221
2.4.1 General	221
2.4.2 Display Selection	221
2.4.3 Expansion and Offcentering	222
2.4.4 Message Position Addressing	224
2.5 Situation Display Camera	225
2.6 Large Board Display Element	225
SECTION 3 DIGITAL DISPLAY SUBSYSTEM OPERATION	227
3.1 Introduction	227
3.2 Digital Data Messages	227
3.2.1 General	227
3.2.2 Message Contents	227
3.2.3 Drum Layout	230
3.3 Digital Display Generator Element	230
3.3.1 Element Function	230
3.3.2 Digital Display Cycle	230
3.3.3 Element Operation	231
3.4 Digital Display Indicator Element	233
3.5 Maintenance Display Console	234
SECTION 4 INFORMATION FLOW	235
4.1 Introduction	235
4.2 Relation to Manual Data Input Element	235
4.2.1 Situation Display Subsystem	235
4.2.1.1 General	235
4.2.1.2 Light-Gun Operation	237
4.2.1.3 Area Discriminator Operation	237
4.2.2 Digital Display Subsystem	238
4.3 Duplexed Information Flow	239

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
CHAPTER 7 OUTPUT SYSTEM	241
7.1 System Discussion	241
7.1.1 System Function	241
7.1.2 System Logic	241
7.1.3 Information Forms	241
7.1.3.1 Bursts	241
7.1.3.2 Output Drum Word	243
7.1.3.3 Output Messages	244
7.2 System Operation	244
7.2.1 Output Control Element	244
7.2.1.1 Introduction	244
7.2.1.2 Checking and Sorting	245
7.2.1.3 Output Alarms	247
7.2.1.4 Output Computer Section	248
7.2.2 Output Storage Element	249
7.2.2.1 Ground-to-Air Output Storage Section	249
7.2.2.2 Ground-to-Ground Output Storage Element	251
7.2.2.3 Teletype Output Storage Section	254
7.2.3 Output Test Section	255
7.3 Information Flow	255
7.3.1 Burst Preparation	255
7.3.1.1 Ground-to-Air Bursts	255
7.3.1.2 Ground-to-Ground Bursts	257
7.3.1.3 Teletype Bursts	259
7.3.2 Delivery to Drum System	260
CHAPTER 8 WARNING LIGHT SYSTEM	263
8.1 System Description	263
8.1.1 System Function	263
8.1.2 System Logic	263
8.2 System Operation	263
8.2.1 General	263
8.2.2 Analysis	264
8.3 Information Flow	265

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
CHAPTER 9 DUPLEX FACILITIES	267
9.1 Equipment Description	267
9.1.1 Introduction	267
9.1.2 Duplex Philosophy	267
9.2 Status	269
9.2.1 Active Status	269
9.2.2 Standby Status	269
9.2.3 Other	269
9.2.4 Status Control	269
9.3 Duplex Switching Consoles	269
9.3.1 Introduction	269
9.3.2 Duplex Switching Controls	270
9.3.2.1 General	270
9.3.2.2 Duplex ACTIVE Switch	270
9.3.2.3 INTERLOCK BYPASS Switch	270
9.3.2.4 TEST OPERATE OVERRIDE BYPASS Switch	270
9.3.3 Duplex Switching Indicators	271
9.3.4 Power Switching Controls	271
9.3.4.1 General	271
9.3.4.2 Simplex Power ACTIVE Switch	272
9.3.4.3 INTERLOCK BYPASS Switch	272
9.3.5 Simplex Power Switching Indicators	272
9.3.6 Other Facilities	272
9.4 Unit Status Switches	273
CHAPTER 10 MAINTENANCE FACILITIES	274
SECTION 1 INTRODUCTION	274
1.1 General	274
1.2 Central Maintenance Facilities	274
1.2.1 Duplex Maintenance Consoles	274
1.2.2 Simplex Maintenance Console	274
1.2.3 Marginal Checking	276

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
SECTION 2 DUPLEX MAINTENANCE CONSOLE	277
2.1 General	277
2.2 Central Scope and Probes Panel	277
2.3 Marginal Checking Control Panel	277
2.3.1 General	277
2.3.2 POWER GROUP Controls and Indicators	278
2.3.3 NOT READY Indicators	278
2.4 Central Computer System Control Panel	279
2.4.1 General	279
2.4.2 POWER ALARMS	279
2.4.3 ALARMS Indicators	281
2.4.4 Alarm Control Switches	281
2.4.4.1 General	281
2.4.4.2 Alarm Activation Switches	283
2.4.4.3 STOP-BRANCH Switch	284
2.4.4.4 Applications	284
2.4.5 ALARMS Neon Group	284
2.4.6 CONDITION LIGHT Neon Group	284
2.4.7 TPD AND CYCLE CONTROL Neon Group	284
2.4.8 INSTRUCTION Neon Group	285
2.4.9 DIVIDE TPD Neons Group	291
2.4.10 ARITHMETIC Neon Group	291
2.4.11 MEMORY 1 and MEMORY 2 Neon Groups	294
2.4.12 DRUMS Neon Group	294
2.4.13 CARD MACHINES Neon Group	294
2.4.14 MISC SELECTION Neon Group	294
2.4.15 TAPE Neon Group	296
2.4.16 Test and SENSE Control Switches	296
2.4.16.1 General	296
2.4.16.2 OPERATE COMPUTER-TEST Switch	296
2.4.16.3 SENSE Switches	296
2.4.16.4 UNASSIGN-ASSIGN Switch	296
2.4.16.5 BRANCH NORMAL-BRANCH TO ZERO MEM Switch	297
2.4.16.6 CAMERA INDEX-SUPPRESS Switch	297

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.4.16.7 NORMAL-REVERSED Switch	297
2.4.17 Loading Preparation Controls	297
2.4.17.1 General	297
2.4.17.2 CLEAR MEMORY Pushbutton	297
2.4.17.3 RESET FLIP-FLOPS Pushbutton	297
2.4.17.4 CLEAR ALARMS Pushbutton	298
2.4.17.5 RESET AUDIBLE ALARM Pushbutton	298
2.4.17.6 MASTER RESET Pushbutton	298
2.4.18 Program Loading Controls	299
2.4.18.1 General	299
2.4.18.2 PROGRAM STOP Pushbutton	299
2.4.18.3 PROGRAM CONTINUE Pushbutton	299
2.4.18.4 LOAD FROM CARD READER Pushbutton	299
2.4.18.5 LOAD FROM AM DRUMS Pushbutton	300
2.4.18.6 START FROM TEST MEMORY Pushbutton	300
2.4.19 Program Loading Control and Test Indicators	301
2.4.20 Servicing Controls	301
2.4.20.1 General	301
2.4.20.2 INST STEP Pushbutton	301
2.4.20.3 MEMORY CYCLE Pushbutton	301
2.4.20.4 SINGLE PULSE Pushbutton	301
2.4.20.5 COMPLEMENT Pushbutton	302
2.4.21 Miscellaneous Controls and Indicators	302
2.4.21.1 General	302
2.4.21.2 AIR COND TROUBLE ACKNOW Pushbutton	302
2.4.21.3 RESET AIR COND AUD ALARM Pushbutton	302
2.4.21.4 SELECT TEST MEMORY Pushbutton	302
2.4.21.5 READY IO UNITS Pushbuttons	302
2.4.21.6 START CAMERA MODE 1 and 2 Pushbuttons	302
2.4.22 COMPLEMENT Lever Switch	302
2.4.23 COMPUTER STATUS Indicators	302
2.4.24 Cyclic Program Controls	303
2.5 Register Neons Panel	304
2.6 Test Memory	304

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.6.1 Capacity	304
2.6.2 Register Characteristics	304
2.6.3 Test Memory Reading	306
2.6.4 Test Memory Selection and Addressing	307
2.7 Tape Indicators and Controls	307
2.8 Main Drum Group Indicator Panel	309
2.8.1 General	309
2.8.2 COMPUTER TO DRUMS Neons Group	310
2.8.3 INPUT DRUMS Neon Group	313
2.8.4 DISPLAY FIELDS Neons Group	317
2.8.5 Test Indicators and Controls	318
2.8.5.1 General	318
2.8.5.2 TEST Neon Group	318
2.8.5.3 DRUM SELECT Controls and Indicators	319
2.9 AUXILIARY DRUMS Indicator Panel	320
2.9.1 General	320
2.9.2 Test Controls and Indicators	321
2.9.3 TEST Neon Group	323
2.9.4 COMPUTER TO DRUMS Neon Group	323
2.9.5 Auxiliary Drums Test Register Neons	324
2.10 Display Tester Control Panel	325
2.10.1 General	325
2.10.2 BIT STORAGE CONTROL Lever Switches	326
2.10.3 Control Switches and Indicators	326
2.10.3.1 Uses	326
2.10.3.2 OPERATE-TEST DD Switch	326
2.10.3.3 OPERATE-TEST SD Switch	326
2.10.3.4 DISP DD CONT-DISP 1 CYC DD Switch	326
2.10.3.5 TD-RD Switch	327
2.10.3.6 DISP RD BRIGHT-DISP RD DIM Switch	327
2.10.3.7 STARTER TESTER Pushbutton	327
2.10.3.8 Indicators	327
2.11 Display Generators Indicators	328
2.11.1 General	328

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.11.2 SITUATION DISPLAY GENERATOR Neon Group	328
2.11.3 Display Registers Neon Group	329
2.11.4 DISPLAY TESTER Neon Group	329
2.11.5 DIGITAL DISPLAY GENERATOR Neon Group	330
2.11.6 CAMERA CONTROL Neon Group	332
2.12 WARNING LIGHTS Indicators	332
2.13 MANUAL INPUTS Indicators	332
2.14 PHONE LINE INPUTS Indicators and Controls	335
2.14.1 General	335
2.14.2 LRI DUPLEX EQUIPMENT Neons and Control	336
2.14.3 GFI DUPLEX EQUIPMENT Neons	337
2.14.4 XTL DUPLEX EQUIPMENT Neons and Control	337
2.15 OUTPUT SYSTEM Indicators	338
2.15.1 General	338
2.15.2 Counter and Output Parity Generator Neons Group	338
2.15.3 COMMON CONTROLS Neon Group	339
2.15.4 G/A CONTROLS Neon Group	339
2.15.5 G/G CONTROLS Neon Group	341
2.15.6 TTY CONTROLS Neon Group	342
2.16 OUTPUT TEST CONTROL PANEL	342
2.16.1 General	342
2.16.2 OUTPUT ALARMS Neon Group	342
2.16.3 G/A LOOPED TO LRI Controls and Indicators	343
2.16.3.1 General	343
2.16.3.2 SELECT G/A Controls and Indicators	343
2.16.4 SELECT G/G Controls and Indicators	344
2.16.5 TTY LOOPED TO TTY RCVR Controls and Indicators	344
2.16.6 UNIT LOOP TEST Neon Group	344
2.16.7 OUTPUT Controls and Indicators	345
2.16.8 G/A, TTY, G/G OUTPUT DATA Neon Groups	346
SECTION 3 SIMPLEX MAINTENANCE CONSOLE	347
3.1 General	347
3.2 Central Scope and Probes Panel	347

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
3.3 Simplex Marginal Checking Control Panel	347
3.4 Power Supply Control Panels	347
3.5 Computer Entry Punch Panel	350
3.5.1 General	350
3.5.2 Air Conditioning Controls	350
3.5.3 Computer Entry Punches Indicators	351
3.6 GFI Channel Control Panels	351
3.6.1 General	351
3.6.2 GFI ALARMS Control and Indicators	351
3.6.3 GFI Unit Status Rotary Switch	351
3.6.4 GFI POWER Indicators	351
3.6.5 GFI DATA CIRCUITS Controls and Indicators	351
3.6.6 CHANNEL SELECTOR Rotary Switch	351
3.6.7 GFI Channel Input Section Neons	351
3.7 XTL Channel Control Panels	351
3.7.1 General	351
3.7.2 XTL ALARMS Controls and Indicators	355
3.7.3 XTL Unit Status Rotary Switch	356
3.7.4 XTL POWER Indicators	356
3.7.5 XTL DATA CIRCUITS Controls and Indicators	356
3.7.6 Channel Selector Rotary Switch	357
3.7.7 XTL Channel Input Section Neons	358
3.8 LRI Channel Control Panels	358
3.8.1 General	358
3.8.2 LRI ALARMS Controls and Indicators	358
3.8.3 LRI Unit Status Rotary Switch	358
3.8.4 LRI Power Indicators	358
3.8.5 LRI DATA CIRCUITS Controls and Indicators	360
3.8.6 Channel Selector Rotary Switch	360
3.8.7 LRI Channel Input Section Neons	360
3.9 GFI, XTL, LRI Test Pattern Generator Control Panel	360
3.9.1 General	360
3.9.2 GFI Pattern Generator Controls and Indicators	362
3.9.3 LRI Pattern Generator Controls and Indicators	362
3.9.4 XTL Pattern Generator Controls and Indicators	362

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
SECTION 4 MARGINAL CHECKING	363
4.1 Organization	363
4.2 Modes of Operation, Duplex	364
4.2.1 Manual Mode	364
4.2.2 Satellite Mode	368
4.2.3 Calculator Mode	368
4.3 Modes of Operation, Simplex	370
4.4 Code Assignments	373
4.4.1 Marginal Checking	373
4.4.2 Circuit Groups and Lines	373
4.4.3 Indirect Checks	374
4.5 Mimic Panel	374
PART 2 PROGRAMMING APPLICATIONS	375
CHAPTER 1 OPERATIONAL PROGRAM ORGANIZATION	375
SECTION 1 INTRODUCTION	375
1.1 Purpose	375
1.2 Direction Center Functions	375
1.2.1 General	375
1.2.2 Air Surveillance	375
1.2.2.1 Overall	375
1.2.2.2 Radar Inputs and Mapping	375
1.2.2.3 Track Detection and Initiation	375
1.2.2.4 Track Monitoring	376
1.2.2.5 Crosstelling	376
1.2.2.6 Forward-telling	376
1.2.2.7 Height Finding	376
1.2.2.8 Identification	377
1.2.2.9 Raid Forming	377
1.2.3 Weapons Direction	377
1.2.3.1 Overall	377
1.2.3.2 Weapons Assignment	377
1.2.3.3 Intercept Direction	377

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
1.2.3.4 Antiaircraft Direction	378
1.2.4 Manual Data Inputs	378
1.2.5 Weather	378
1.2.6 Subsector Command Post	378
1.2.7 Training and Battle Simulation	379
1.2.8 Standby Operation	379
1.2.9 Recording and Analysis	379
1.3 Response Time	379
1.3.1 Introduction	379
1.3.2 Timing and Sequencing of Programs	380
1.4 Direction Center Programs	380
1.4.1 General	380
1.4.2 Operational Active Program	380
1.4.3 Operational Standby Program	380
1.4.4 Operational Supporting Program	380
1.4.5 Utility Supporting Programs	381
1.4.6 Maintenance Programs	381
SECTION 2 OPERATIONAL ACTIVE PROGRAM	382
2.1 Program Design	382
2.1.1 Introduction	382
2.1.2 Central Group Programs	382
2.1.2.1 Sequence Control Program	282
2.1.2.2 Keyboard Message Interpretation Program Group	383
2.1.2.3 Situation Display Program Group	383
2.1.2.4 Digital Display Program Group	384
2.1.2.5 Input-Output Program Group	384
2.1.2.6 Central Bookkeeping Program Group	384
2.1.3 Operational Program Group	384
2.1.3.1 Tracking Program Group	384
2.1.3.2 Weapons Direction Program Group	385
2.1.3.3 Miscellaneous Program Group	385
2.2 Program Timing	386
2.2.1 General	386
2.2.2 Program Classes	386

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.2.3 Program Sequence	387
2.2.4 Effect of Program Sequence on Response Times	388
SECTION 3 OPERATIONAL STANDBY PROGRAM	389
3.1 Introduction	389
3.2 Air Defense Requirements	389
3.2.1 General	389
3.2.2 Startover	389
3.2.3 Safe Data Storage	389
3.2.4 Active Computer Monitoring	389
3.2.5 Standby Computer Monitoring	389
3.3 Maintenance Requirements	389
3.3.1 General	389
3.3.2 Standby Modes of Operation	390
3.3.2.1 Duplex Mode	390
3.3.2.2 Simplex Mode	390
3.3.3 Operational Standby Program, Duplex Mode	390
3.3.3.1 Sequence Control Program	390
3.3.3.2 Duplex Switching Console Programs	390
3.3.3.3 Active Computer Monitoring Program	390
3.3.3.4 Safe Data Storage Program	390
3.3.3.5 Startover Program	390
CHAPTER 2 PROGRAMMING DATA	391
SECTION 1 INTRODUCTION	391
1.1 General	391
1.2 Numbers and Arithmetic	391
1.2.1 Representation of Numbers	391
1.2.2 Addition and Subtraction	392
1.2.3 Overflow	392
1.2.4 Division	393
1.3 Sequence of Instruction Execution	393
1.3.1 Program Counter	393
1.3.2 Indexing	393

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
1.4 Internal Memory Addressing	393
1.4.1 General	393
1.4.2 Core Memory	394
1.4.3 Test Memory	394
1.4.4 Clock Register	395
SECTION 2 INSTRUCTIONS	396
2.1 General	396
2.2 Miscellaneous Class	396
2.2.1 Program Stop (HLT) Instruction	396
2.2.2 Extract (ETR) Instruction	396
2.2.3 Operate (PER) Instruction	396
2.2.4 Clear and Subtract Word Counter (CSW) Instruction	397
2.2.5 Shift Left and Round (SLR) Instruction	397
2.2.6 Load B Register (LDB) Instruction	397
2.3 Add Class	397
2.3.1 Clear and Add (CAD) Instruction	397
2.3.2 Add (ADD) Instruction	397
2.3.3 Twin and Add (TAD) Instruction	397
2.3.4 Add B Registers to Accumulators (ADB) Instruction	398
2.3.5 Clear and Subtract (CSU) Instruction	398
2.3.6 Subtract (SUB) Instruction	398
2.3.7 Twin and Subtract (TSU) Instruction	398
2.3.8 Clear and Add Magnitudes (CAM) Instruction	398
2.3.9 Difference in Magnitudes (DIM) Instruction	398
2.4 Multiply Class	398
2.4.1 Multiply (MUL) Instruction	398
2.4.2 Twin and Multiply (TMU) Instruction	398
2.4.3 Divide (DVD) Instruction	398
2.4.4 Twin and Divide (TDV) Instruction	398
2.5 Store Class	399
2.5.1 Store (FST) Instruction	399
2.5.2 Left Store (LST) Instruction	399
2.5.3 Right Store (RST) Instruction	399

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.5.4 <i>Store Address (STA) Instruction</i>	399
2.5.5 <i>Add One (AOR) Instruction</i>	399
2.5.6 <i>Exchange (ECH) Instruction</i>	399
2.5.7 <i>Deposit (DEP) Instruction</i>	399
2.6 <i>Shift Class</i>	399
2.6.1 <i>Shift Left (DSL) Instruction</i>	399
2.6.2 <i>Shift Right (DSR) Instruction</i>	399
2.6.3 <i>Shift Accumulators Left (ASL) Instruction</i>	400
2.6.4 <i>Shift Accumulators Right (ASR) Instruction</i>	400
2.6.5 <i>Left Element Shift Right (SLR) Instruction</i>	400
2.6.6 <i>Right Element Shift Right (RSR) Instruction</i>	400
2.6.7 <i>Cycle Left (DCL) Instruction</i>	400
2.6.8 <i>Cycle Accumulators Left (FCL) Instruction</i>	400
2.7 <i>Branch Class</i>	400
2.7.1 <i>Branch and Index (BPX) Instruction</i>	401
2.7.2 <i>Sense (BSN) Instruction</i>	401
2.7.3 <i>Branch on Zero (BFZ) Instruction</i>	402
2.7.4 <i>Branch on Minus (BFM) Instruction</i>	402
2.7.5 <i>Branch on Left Minus (BLM) Instruction</i>	402
2.7.6 <i>Branch on Right Minus (BRM) Instruction</i>	402
2.8 <i>IO Class</i>	403
2.8.1 <i>Load Input-Output Address Counter (LDC) Instruction</i>	403
2.8.2 <i>Select Drum (SDR) Instruction</i>	403
2.8.3 <i>Select (SEL) Instruction</i>	406
2.8.4 <i>Read (RDS) Instruction</i>	406
2.8.5 <i>Write (WRT) Instruction</i>	407
2.9 <i>Reset Class</i>	407
2.9.1 <i>Reset Index Register (XIN) Instruction</i>	407
2.9.2 <i>Reset Index Register from Right Accumulator (XAC)</i> <i>Instruction</i>	407
2.9.3 <i>Add Index Register (ADX) Instruction</i>	407
2.10 <i>Illegal Instructions</i>	407
2.10.1 <i>General</i>	407
2.10.2 <i>Illegal Instruction, Type 0</i>	407

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.10.3 Illegal Instruction, Type 1	407
2.10.4 Illegal Instruction, Type 2	407
2.10.5 Illegal Instruction, Type 3	407
2.10.6 Illegal Instruction, Type 4	407
2.10.7 Illegal Instruction, Type 5	407
2.10.8 Illegal Instruction, Type 6	408
2.10.9 Illegal Instruction, Type 7	408
SECTION 3 OTHER PERTINENT DATA	409
3.1 IO Process	409
3.1.1 IO Transfers	409
3.1.2 IO Interlock	409
3.1.3 IO Word Counter	410
3.1.4 IO Address Counter	410
3.1.5 Interleave	410
3.2 Alarms and Checks	421
3.2.1 Alarm Actions	423
3.2.2 Parity Checking	423
3.3 Miscellaneous Operating Controls	423
3.3.1 Automatic Branch to Zero	423
3.3.2 Test and Operate Modes	423
3.3.3 Duplex Switching	424
CHAPTER 3 PROGRAMMING TECHNIQUES	425
SECTION 1 GENERAL	425
1.1 Introduction	425
1.2 Alternate Routine Selection	425
1.2.1 Operational Programs	425
1.2.2 Maintenance Programs	425
1.3 Checking	426
SECTION 2 SCALING	427
2.1 Purpose and Definition	427
2.2 Notation	427
2.3 Consideration	427

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.4 Procedure	428
2.5 Examples	428
2.5.1 Multiplication Scaling	428
2.5.2 Division Scaling	428
2.5.3 Accumulation Scaling	429
2.5.3.1 General Procedure	429
2.5.3.2 Simple Addition	429
2.5.3.3 Compound Multiplication and Division	429
2.5.3.4 Function Evaluation	430
CHAPTER 4 IO PROGRAMMING	433
4.1 Introduction	433
4.2 IO Units	433
4.2.1 Card Reader	433
4.2.2 Card Punch	433
4.2.3 Line Printer	433
4.2.4 IO Register	434
4.2.5 Manual Input Matrix	434
4.2.6 Warning Lights Registers	434
4.2.7 Magnetic Tapes	434
4.2.8 Burst Time Counters	435
4.3 Drum Fields	435
4.3.1 Auxiliary Memory Fields	435
4.3.2 Time Buffer Fields	435
4.3.2.1 Reading	435
4.3.2.2 Writing	436
4.3.2.3 Number of Words Transferred	436
CHAPTER 5 INPUT OPERATIONS	437
5.1 Information Forms	437
5.1.1 Introduction	437
5.1.2 LRI Message Word Format	437
5.1.2.1 General	437
5.1.2.2 Search Radar	437
5.1.2.3 IFF	437
5.1.2.4 Height Reply	437

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
5.1.3 GFI Message Word Format	437
5.1.4 XTL Message Word Format	437
5.1.5 MDI Message Word Formats	439
5.1.5.1 General	439
5.1.5.2 Computer Entry Punch	439
5.1.5.3 Radar Data Word	439
5.1.5.4 Track Display Word	439
5.1.5.5 MDI Matrix Core Assignments	439
5.2 Transfer to Central Computer System	441
5.2.1 General	441
5.2.2 Reading by Status Identification	441
5.2.3 Sorting	441
5.2.3.1 LRI Element Information	441
5.2.3.2 GFI Element Information	442
5.2.3.3 XTL Element Information	442
5.2.3.4 MDI Drum Entry Section Information	442
5.2.4 Reading by Status	442
5.3 Information Processing	442
5.3.1 Radar Data Inputs	442
5.3.1.1 General	442
5.3.1.2 Mapping	443
5.3.1.3 Radar Orientation	444
5.3.1.4 Masking	444
5.3.1.5 Quality Analysis	444
5.3.1.6 Radar Set Status Control	444
5.3.1.7 Co-ordinate Conversion and Transformation	445
5.3.2 Correlation	445
5.3.2.1 Principles of Tracking	445
5.3.2.2 Search Areas	445
5.3.2.3 Multiple Radar Coverage	445
5.3.2.4 Track Sorting	445
5.3.2.5 Crossing Track Situations	446
5.3.2.6 Smoothing and Prediction	446
5.3.3 XTL Information Processing	446

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
5.3.4 MDI Information Processing	446
5.3.4.1 Automatic Initiation	446
5.3.4.2 Manual Initiation	447
5.3.4.3 Card Processing	447
5.3.4.4 Light Gun and Keyboard Messages	447
CHAPTER 6 DISPLAY OPERATIONS	449
SECTION 1 SITUATION DISPLAYS	449
1.1 Introduction	449
1.2 Message Types	449
1.2.1 Major Classifications	449
1.2.2 Radar Data Messages	449
1.2.3 Track Display Messages	449
1.2.3.1 General	449
1.2.3.2 Tabular Messages	449
1.2.3.3 Vector Messages	449
1.3 Information Contents	449
1.3.1 Radar Data Messages	449
1.3.2 Vector Messages	449
1.3.3 Tabular Track Messages	450
1.3.3.1 Point Feature	450
1.3.3.2 Vector Feature	450
1.3.3.3 E Feature	450
1.3.3.4 C Feature	450
1.3.3.5 D Feature	450
1.3.3.6 B Feature	452
1.3.3.7 A Feature	452
1.3.3.8 Examples	453
1.3.4 Tabular Information Messages	454
1.4 Message Routing	454
1.4.1 General	454
1.4.2 Categories	455
1.4.3 Display Assignment Bits	456
1.4.4 Special Test Lines	459

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
SECTION 2 DIGITAL DISPLAYS	460
2.1 General	460
2.2 Character Forming Matrix	460
2.3 Message Format	460
2.4 Message Distribution	460
2.5 Provisions for Expansion of Display Capacity	465
 CHAPTER 7 OUTPUT OPERATIONS	 467
7.1 Information Forms	467
7.1.1 General	467
7.1.2 Ground-to-Air Information	467
7.1.3 Ground-to-Ground Information	467
7.1.4 Teletype Information	468
7.2 Message Preparation	468
7.2.1 General	468
7.2.2 Section and Register Address	469
7.2.3 Burst Number Assignment	469
7.2.3.1 General	469
7.2.3.2 Number of Stored Bursts	470
7.2.3.3 Writing Time	471
7.3 Message Delivery	472
7.3.1 General	472
7.3.2 Input-Output Transfers to the Output Buffer Fields	472
 CHAPTER 8 WARNING LIGHTS OPERATION	 475
8.1 General	475
8.2 Warning Lights Programming	475
8.3 Information Flow	476
 CHAPTER 9 DUPLEX OPERATIONS	 479
9.1 Introduction	479
9.2 Intercommunication	479
9.3 Other Computer Alarms	479
9.4 Unit Status Evaluation	479
9.5 Switching Console Operation	480

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
CHAPTER 10 MAINTENANCE OPERATIONS	481
SECTION 1 MAINTENANCE PROGRAMS	481
1.1 Introduction	481
1.2 Reliability Programs	481
1.2.1 Definition	481
1.2.2 Types	481
1.2.3 Interpretation	481
1.3 Diagnostic Programs	481
1.3.1 Definition	481
1.3.2 Diagnostic Techniques	482
1.3.2.1 Increasing Area Check	482
1.3.2.2 Decreasing Area Check	482
1.3.2.3 Overlapping Area Check	482
1.3.2.4 Large Area Localization	482
1.3.2.5 Individual Circuit Check	482
1.3.3 Marginal Checking with Diagnostic Programs	482
1.4 Utility Programs	482
1.5 Maintenance Program Identification Codes	482
SECTION 2 MARGINAL CHECKING	484
2.1 Introduction	484
2.1.1 General	484
2.1.2 Basic Reliability Testing	484
2.1.3 Definition of Terms	484
2.1.4 Checking for Imminent Failure	484
2.2 Calculator-Controlled Marginal Checking	485
2.2.1 General	485
2.2.2 Testing Sequence	488
2.2.2.1 Basic Reliability Programs	488
2.2.2.2 Reliability Programs with Marginal Checking	488
2.2.2.3 Advantages of Programmed Marginal Checking	488
2.2.3 Programmed Marginal Checking	488
2.2.3.1 Excursion Application	488
2.2.3.2 Excursion Removal	489

CONTENTS (cont'd)

<i>Heading</i>	<i>Page</i>
2.2.3.3 Excursion Detection	489
2.2.3.4 Use of LS Bit	489
2.2.3.5 Restarts	489
2.2.3.6 Time Duration	490
2.2.3.7 Polarity	490
2.2.3.8 Safe Limit	490
2.2.3.9 Excursion Magnitude	490
2.3 Marginal Checking Breakdown Charts	490
2.4 Special Considerations	491

LIST OF ILLUSTRATIONS

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-1	Radar Co-ordinate Conversion	6
1-2	AN/FSQ-7 (DC-1) Combat Direction Central Information Flow	6
1-3	Central Computer System, Element Block Diagram	7
1-4	Internal Information Flow	8
1-5	IO Information Flow	8
1-6	Input System Information Flow	8
1-7	Drum System Information Flow	9
1-8	Display System Information Flow	10
1-9	Output System Information Flow	10
1-10	Duplexing, Simplified Block Diagram	11
1-11	Computer Word	28
1-12	Core Memory Unit	31
1-13	Hysteresis Loop of Ferrite Core	32
1-14	Core Memory Plane (55-611)	33
1-15	Core Array, Simplified	34
1-16	Memory Cycles	35
1-17	Machine and Instruction Cycles	38
1-18	Arithmetic Element Information Flow	39
1-19	Internal Control Information Flow	40
1-20	IO Control Information Flow	42
1-21	Instruction Flow	43
1-22	Instruction Decoding	44
1-23	Indexed Address Modification	45
1-24	Adder Circuit	46
1-25	Inherent Shift Right	47
1-26	Add Class Execution	47
1-27	<i>Difference in Magnitudes (DIM)</i> Execution	48
1-28	<i>Multiply (MUL, TMU)</i> Execution	50
1-29	<i>Divide (DVD, TDV)</i> Execution	52
1-30	Store Class Execution	54
1-31	<i>Add (AOR)</i> Execution	54

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-32	<i>Exchange (ECH) Execution</i>	55
1-33	<i>Deposit (DEP) Execution</i>	55
1-34	<i>Shift Class Execution</i>	57
1-35	<i>Cycle Variations Execution</i>	58
1-36	<i>Branch Class Execution</i>	59
1-37	<i>Branch on Zero (BFZ) Execution</i>	60
1-38	<i>Sense (BSN) Execution</i>	61
1-39	<i>Branch and Index (BPX) Execution</i>	64
1-40	<i>IO Class Execution</i>	65
1-41	<i>Reset Class Execution</i>	67
1-42	<i>Extract (ETR) Execution</i>	68
1-43	<i>Clear and Subtract Word Counter (CSW) Execution</i>	68
1-44	<i>Shift Left and Round (SLR) Execution</i>	70
1-45	<i>Basic IBM Card Showing Hollerith Code Zones</i>	71
1-46	<i>Field Division on Punch Cards</i>	72
1-47	<i>Instruction Card</i>	73
1-48	<i>Special Punching on Binary Card</i>	74
1-49	<i>Relation of Card Image to IBM Card</i>	75
1-50	<i>Card Reader, Type 713</i>	76
1-51	<i>Card Feed Unit, Card Reader</i>	77
1-52	<i>Card Reader Controls and Indicators</i>	77
1-53	<i>Card Reader Information Flow</i>	79
1-54	<i>Card Punch, Type 723</i>	80
1-55	<i>Card Feed Unit, Card Punch</i>	81
1-56	<i>Card Punch Controls and Indicators</i>	81
1-57	<i>Card Punch Information Flow</i>	82
1-58	<i>Line Printer, Type 718</i>	83
1-59	<i>Type Wheel, Pictorial Diagram</i>	84
1-60	<i>Line Printer Control Panel</i>	85
1-61	<i>Line Printer Controls and Indicators</i>	86
1-62	<i>Tape Word Bit Positions</i>	88
1-63	<i>Tape Reel Information and Control Units</i>	88
1-64	<i>Tape Drive Unit</i>	90
1-65	<i>Tape Element Information Flow</i>	92

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-66	Tape Adapter Unit Information Flow	93
1-67	Number Sorting Program, Flow Chart Form	105
1-68	Indexed Number Sorting Program, Flow Chart Form	112
1-69	Use of Closed Subroutine in Larger Program	118
1-70	Instruction Card	121
1-71	Relation of Drum System to Other Systems	125
1-72	Intercommunication in AN/FSQ-7 Combat Direction Central	126
1-73	The OD and CD Sides of the Drum System	126
1-74	Magnetic Drum	127
1-75	Magnetic Drum and Heads	128
1-76	Drum System Information Units	129
1-77	Timing Chart, Drum (CD Side) and Central Computer System	130
1-78	Drum System Logical Divisions	131
1-79	Drum and Field Selection	133
1-80	Writing by Status	138
1-81	Reading by Status-Identification	139
1-82	Writing by Address	141
1-83	Reading by Address	142
1-84	Input System Information Flow (OD Side)	146
1-85	Output System Information Flow (OD Side)	147
1-86	Display System Information Flow (OD Side)	148
1-87	Auxiliary Information Flow	149
1-88	Auxiliary Drum Group Information Flow	150
1-89	Input Information Flow (CD Side)	151
1-90	Output Information Flow (CD Side)	152
1-91	Intercommunication Information Flow	153
1-92	Computer Test	156
1-93	Manual Drum Tester Panel	157
1-94	Auxiliary Drum Tester Panel	158
1-95	Input System Information Flow	159
1-96	LRI Element Information Flow	160
1-97	LRI Duplexing	160
1-98	P Site Information Flow	161
1-99	P Site Data-Handling Equipment	163

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-100	Height-Range Indicator, Semiautomatic Height-Finder	164
1-101	LRI Telephone Line Message Layouts	165
1-102	LRI Channel Input Section	166
1-103	LRI Common Input Section	167
1-104	LRI Monitor Console	169
1-105	LRI Message Drum Field Layouts	170
1-106	LRI Element, Simplex and Duplex Switching	171
1-107	Gap-Filler Input Element	173
1-108	GFI Element, Duplexing	174
1-109	Quantization of GFI Radar Coverage Area, Simplified	174
1-110	Video Integration of GFI Radar Data	175
1-111	GFI Mapper Console	176
1-112	GFI Mapper Section, Simplified Diagram	176
1-113	GFI Counter Section, Simplified Diagram	177
1-114	GFI Common Section, Simplified Diagram	177
1-115	GFI Monitor Console	178
1-116	GFI Message Drum Field Layout	178
1-117	GFI Element, Simplex and Duplex Switching	179
1-118	Crosstelling	181
1-119	Crosstell Input Element	182
1-120	Crosstell Duplexing	182
1-121	Crosstell Message, Interleaved Telephone Line Layout	183
1-122	XTL Channel Input Section	184
1-123	XTL Common Section	185
1-124	Crosstell Message, Drum Field Layout	186
1-125	Crosstell Element, Simplex and Duplex Switching	187
1-126	Manual Data Input Element	189
1-127	Computer Entry Punch, Type 026 Modified	190
1-128	MDI Data Card	191
1-129	MDI Drum Entry Section	192
1-130	MDI Message Drum Field Layouts	195
1-131	MDI Keyboard Mounted on Situation Display Console	196
1-132	MDI Keyboard	197
1-133	MDI Direct IO Buffer Entry Section	198

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-134	MDI Element, Simplex and Duplex Switching	199
1-135	Display System, Logical Divisions	201
1-136	Display Console	202
1-137	Display System Relation to Manual Input Element	203
1-138	Situation Display Console Duplexing	204
1-139	Typical Situation Display	205
1-140	Character Matrix, Situation Display Tube	206
1-141	Situation Display Tube, Simplified Diagram	206
1-142	Typical Digital Display	207
1-143	Character Forming Matrix, Digital Display Tube	207
1-144	Digital Display Tube, Simplified Diagram	207
1-145	Typical Radar Data Message Displays	210
1-146	Radar Data Message Drum Layout	210
1-147	Octonary Character Addressing, Situation Display Tube	211
1-148	Tabular Message Display Format	212
1-149	Typical Tabular Message Displays	212
1-150	Tabular Message Drum Layout	213
1-151	Effect of Position Bit on Character Format	214
1-152	Vector Message Display Format and Character Positioning	215
1-153	Typical Vector Message Displays	215
1-154	Vector Message Drum Layout	216
1-155	Situation Display Generator Element	220
1-156	Situation Display Indicator Element	221
1-157	SD Console Controls	223
1-158	SD CRT and Subsector Co-ordinate Relations	224
1-159	Situation Display Camera	226
1-160	Digital Display Message Format	228
1-161	Octonary Character Addressing, Digital Display Tube	229
1-162	Digital Data Message Drum Layout	229
1-163	Digital Display Generator Element	231
1-164	Auxiliary Console	232
1-165	Maintenance Display Console	233
1-166	Maintenance Display Console Input Signal Control Panel	234
1-167	Information Return from Display System via Manual Data Input Element	236

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-168	Light Gun	237
1-169	Area Discriminator Console	238
1-170	Display System Duplexed Information Flow	239
1-171	Output System, Information Flow	241
1-172	Output System, Element Block Diagram	242
1-173	Output System Duplex Switching	242
1-174	Output System Timing	243
1-175	Output Drum Word Layout	244
1-176	Transmitted G/A Information Block	245
1-177	Transmitted G/G Message	245
1-178	Teletype Character Composition	245
1-179	Transmitted TTY Burst	245
1-180	Output Control Element, Simplified	246
1-181	Output Computer Section, Simplified	248
1-182	Burst Counts As Read by Central Computer System	249
1-183	G/A Output Storage Section, Simplified	250
1-184	G/G Output Storage Section, Simplified	252
1-185	TTY Output Storage Section, Simplified	253
1-186	Teletype Printer with Keyboard Used As TTY Monitor	256
1-187	G/A Drum Word Layout	257
1-188	G/G Information Flow	257
1-189	Height-Request Message Layout	258
1-190	TTY Information Flow	260
1-191	Warning Light Control and Storage Elements, Simplified Block Diagram	264
1-192	Duplex and Simplex Switching Facilities	268
1-193	Duplex Switching Console	269
1-194	Central Maintenance Area Facilities	275
1-195	Duplex Maintenance Console	277
1-196	POWER GROUP and NOT READY Controls and Indicators	277
1-197	Central Computer System Control Panel	279
1-198	POWER ALARMS Indicators	280
1-199	ALARMS Indicators	282
1-200	Alarm Activation Switches	282

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1-201	ALARMS and CONDITION LIGHTS Neons	285
1-202	TPD AND CYCLE CONTROL and INSTRUCTION Neons	286
1-203	ARITHMETIC Neons	289
1-204	MEMORY 1 and MEMORY 2 Neons	290
1-205	DRUMS, CARD MACHINES, MISC SELECTION and TAPES Neons	291
1-206	Test and Sense Control Switches	297
1-207	Operation Controls and Indicators	298
1-208	Register Neons	304
1-209	Test Memory Toggle Switch Registers	305
1-210	Test Memory Control Panel	306
1-211	Tape Element Controls and Indicators	308
1-212	Main Drum Group Controls and Indicators	310
1-213	Auxiliary Drum Group Controls and Indicators	321
1-214	Display Tester Controls and Indicators	325
2-215	Display System Neons	327
1-216	Warning Lights Register Neons	333
1-217	Manual Data Input Neons	333
1-218	Common Input System Controls and Indicators	335
1-219	Output System Indicators	338
1-220	Output Test Controls and Indicators	493, foldout
1-221	Simplex Maintenance Console	349
1-222	Power Supply Controls and Indicators	350
1-223	Computer Entry Punch Controls and Indicators	350
1-224	GFI Channel Controls and Indicators	352
1-225	XTL Channel Controls and Indicators	356
1-226	LRI Channel Controls and Indicators	359
1-227	GFI, XTL, LRI Pattern Generators Panel	362
1-228	Marginal Checking Equipment Divisions	363
1-229	Duplex Marginal Checking Controls and Indicators	364
1-230	Simplex Marginal Checking Controls and Indicators	371
2-1	Program Frame Organization and Timing	386
2-2	LRI Element Message Formats	438
2-3	GFI Element Message Format	438

LIST OF ILLUSTRATIONS (cont'd)

<i>Figure</i>	<i>Title</i>	<i>Page</i>
2-4	Radar Data Flow Through Input Programs	443
2-5	Quality Analysis and Set Status Control Flow Diagram	444
2-6	Tabular Track Data Message Displays	453
2-7	Teletype Message, Word Format	468
2-8	Burst Storage Determination	470
2-9	Burst Number Determination	471
2-10	Probability of Failure Curve	485

LIST OF TABLES

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-1	Integral Decimal to Binary Conversion	15
1-2	Fractional Decimal to Binary Conversion	16
1-3	Rules of Binary Addition	18
1-4	Rules of Direct Binary Subtraction	18
1-5	Rules of Binary Multiplication	21
1-6	Binary Multiplication	21
1-7	Binary Division	23
1-8	Binary-Coded Decimal Numbers	24
1-9	Binary-Coded Octonary Numbers	25
1-10	Memory Addresses	36
1-11	Instruction Classes	43
1-12	Add Class Instructions	49
1-13	Multiply Class Instructions	53
1-14	Deposit Information Changes	56
1-15	Store Class Instructions	56
1-16	Shift Class Instructions	59
1-17	Branch on Zero Test Operations	61
1-18	Sense Unit Selection Codes	62
1-19	Branch Class Instructions	64
1-20	Select Unit Codes	66
1-21	IO Class Instructions	66

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-22	Reset Class Instructions	68
1-23	Operate Unit Selection Codes	69
1-24	Miscellaneous Class Instructions	70
1-25	Hollerith Code for Punched Cards	72
1-26	Binary Form of Hollerith Code	74
1-27	Card Reader Controls and Indicators	77
1-28	Card Reader Program	78
1-29	Card Punch Controls and Indicators	81
1-30	Card Punch Program	81
1-31	Line Printer Controls and Indicators	87
1-32	Line Printer Program	87
1-33	Tape Information and Control Units	89
1-34	Tape Driver Unit Operating Specifications	89
1-35	Tape Drive Unit Controls	90
1-36	Tape Drive Unit Control Light Functions	91
1-37	Tape Drive Unit Operation Timing	91
1-38	Ready Versus Not-Ready Conditions	92
1-39	Not-Prepared Conditions	92
1-40	Tape Element Instructions	94
1-41	Program for Read Operation	95
1-42	Program for Write Operation	96
1-43	Program for Backspace Operation	97
1-44	Program for Rewind Operation after Reading End-of-File	98
1-45	Program to Clear Specific Memory Locations	99
1-46	Program to Completely Clear Core Memory	99
1-47	Programmed Addition	100
1-48	Programmed Subtraction	101
1-49	Alternate Subtraction Program	101
1-50	Multiplication Program	102
1-51	Division Program	102
1-52	Co-ordinate Conversion Program	103
1-53	Function Evaluation Program	104
1-54	Straight-Line Addition	108
1-55	Nonindexed Iterative Program	109

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-56	Indexed Iterative Program	110
1-57	Data Storage for Indexed Number Sorting Program	111
1-58	Indexed Function Evaluation Program	114
1-59	Control of Loop Iterations	116
1-60	Zero Repetition Indexed Routine	116
1-61	Number Sorting Program, Symbolic Form	120
1-62	Assembly Direction Card Punching	122
1-63	Printout of Assembled Program	123
1-64	Group, Drum, Field, and Mode Selection Codes	133
1-65	Delays in Drum IO Operation	137
1-66	Drum System Information Flow	154
1-67	Computer Test Operations Which Utilize Computer Test Element	155
1-68	Display System Layout of Situation Display Messages	208
1-69	Display System Instructions	209
1-70	Feature and Character Position Nomenclature for Tabular Track Data Messages	212
1-71	Message Component Nomenclature for Vector Track Data Messages	215
1-72	Control-Bit Functions, Situation Display Messages	217
1-73	Situation Display Cycle Timing	219
1-74	Control Bit Functions, Digital Data Messages	227
1-75	Digital Display Cycle Timing	230
1-76	Output Section Address Codes	243
1-77	Output Register Address Codes	244
1-78	Burst Count Read Program	249
1-79	Teletype Code	259
1-80	Sample Teletype Message	261
1-81	Warning Light Information Transfer	265
1-82	Duplex and Simplex Equipment	267
1-83	Duplex Switch Panel Indicators	271
1-84	Simplex Power Switch Panel Indicators	272
1-85	Power Group Controls and Indicators	278
1-86	NOT READY Indicators	278
1-87	POWER ALARMS Indicators	280
1-88	ALARMS Indicators	282

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-89	Alarm Neons	283
1-90	TPD AND CYCLE CONTROL Neons	287
1-91	INSTRUCTION Neons	287
1-92	ARITHMETIC Neons	288
1-93	MEMORY 1 and MEMORY 2 Neons	290
1-94	DRUMS Neons	292
1-95	CARD MACHINES Neons	293
1-96	MISC SELECTION Neons	294
1-97	TAPE Neons	296
1-98	Assigned Test Memory Addresses	296
1-99	CLEAR MEMORY Pushbutton Equivalent Program	298
1-100	Loading Preparation Controls	299
1-101	LOAD FROM CARD READER Equivalent Program	300
1-102	Program Loading Controls	301
1-103	Program Loading Control and Test Indicators	301
1-104	Cyclic Program Controls	303
1-105	Register Neons	305
1-106	Tape Neons	308
1-107	Tape Indicator Lights and Controls	309
1-108	COMPUTER TO DRUMS Neons	311
1-109	INPUT DRUMS Neons	313
1-110	DISPLAY FIELDS Neons	317
1-111	TEST Neons	318
1-112	DRUM SELECT Controls and Indicators	319
1-113	AUXILIARY DRUMS Control and Indicator Group	321
1-114	TEST Neons	323
1-115	COMPUTER TO DRUMS Neons	323
1-116	Auxiliary Drums Test Register Neons	324
1-117	Display Tester Control Panel Switches and Indicators	326
1-118	SITUATION DISPLAY GENERATOR Neons	328
1-119	DISPLAY TESTER Neons	329
1-120	DIGITAL DISPLAY GENERATOR Neons	330
1-121	CAMERA CONTROL Neons	332
1-122	MANUAL INPUTS Neons	333

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-123	LRI DUPLEX EQUIPMENT Neons	336
1-124	GFI DUPLEX EQUIPMENT Neons	337
1-125	XTL DUPLEX EQUIPMENT Neons	337
1-126	Counter and Output Parity Generator Neons	338
1-127	COMMON CONTROLS Neons	339
1-128	G/A CONTROLS Neons	340
1-129	G/G CONTROLS Neons	341
1-130	TTY CONTROLS Neons	342
1-131	OUTPUT ALARMS Neons	342
1-132	G/A LOOPED TO LRI Neons	344
1-133	UNIT LOOP TEST Neons	344
1-134	OUTPUT Controls and Indicators	345
1-135	G/A, TTY, G/G OUTPUT DATA Neons	346
1-136	Power Supply Controls and Indicators	347
1-137	Computer Entry Punch Controls Indicators	351
1-138	GFI ALARMS Controls and Indicators	351
1-139	GFI Unit Status Rotary Switch Positions	353
1-140	GFI POWER Indicators	353
1-141	GFI DATA CIRCUITS Controls and Indicators	354
1-142	GFI Input Section Neons	354
1-143	XTL ALARMS Controls and Indicators	355
1-144	XTL Unit Status Rotary Switch Positions	355
1-145	XTL POWER Indicators	355
1-146	XTL DATA CIRCUITS Controls and Indicators	356
1-147	XTL Channel Input Section Neons	357
1-148	LRI ALARMS Controls and Indicators	358
1-149	LRI Unit Status Rotary Switch Positions	358
1-150	LRI Power Indicators	358
1-151	LRI DATA CIRCUITS Controls and Indicators	360
1-152	LRI Channel Input Section Neons	361
1-153	GFI Pattern Generator Controls and Indicators	362
1-154	LRI Pattern Generator Controls and Indicators	362
1-155	XTL Pattern Generator Controls and Indicators	362
1-156	Duplex Marginal Checking Controls and Indicators	365

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
1-157	Marginal Checking Control Word Bit Assignments	368
1-158	Simplex Marginal Checking Controls and Indicators	370
1-159	Mimic Panel Indicators	374
2-1	Operational Active Programs	387
2-2	Instruction Codes	391
2-3	Overflow Alarm Control	392
2-4	Useful Illegal Divisions	393
2-5	Selection of Internal Memory Registers	394
2-6	Selection of Internal Memory Units	394
2-7	Instruction Code Summary	495, foldout
2-8	Operate Unit Codes	396
2-9	Sense Codes	401
2-10	Group, Drum, Field, and Mode Selection Codes	403
2-11	IO Unit Selection Codes	406
2-12	Programming an IO Transfer	409
2-13	Interleave Code	410
2-14	Octonary-Decimal Integer Conversion	410
2-15	Alarms Which May Influence Program	422
2-16	Card Punch Not-Ready Sensing Routine	434
2-17	Tape Element Instructions	435
2-18	Light Gun Codes and Core Assignments	439
2-19	Action Bit Core Assignments	440
2-20	Unit Status Core Assignments	440
2-21	Information Core Assignments	440
2-22	LRI Site Identification Codes	441
2-23	LRI Message Label Codes	442
2-24	GFI Site Identification Codes	442
2-25	XTL Site Identification Codes	442
2-26	MDI Message Label Codes	442
2-27	E Feature Symbology	450
2-28	C Feature Symbology	450
2-29	D ₁ Character Position Symbology	450
2-30	D ₂ Character Position Symbology	451
2-31	B ₄ and B ₅ Character Position Symbology	451

LIST OF TABLES (cont'd)

<i>Table</i>	<i>Title</i>	<i>Page</i>
2-32	Track Age Measurement	452
2-33	Interceptor Mission Symbology	452
2-34	Radar Data Categories	454
2-35	Tabular and Vector Message Categories	455
2-36	Display Assignment Bits	457
2-37	Special Test Lines	459
2-38	Control Bit Functions, Digital Display Messages	460
2-39	DD Slot Assignments	461
2-40	G/A Message Types and Values	467
2-41	Type of Height-Finder Request	468
2-42	Burst Number Assignment	472
2-43	OB Field Writing, Sample Program	473
2-44	IO Program for Warning Light System	475
2-45	Loading of Each Bit Position	476
2-46	Warning Light and Audible Alarm Bit Assignments	476
2-47	Maintenance Program Identification Codes	483
2-48	Marginal Checking Control Word Bit Assignments	486

LIST OF RELATED MANUALS*Manual**Title***TABLE OF CONTENTS AND INDEX**

T.O. 31P2-2FSQ7-1	Contents
T.O. 31P2-2FSQ7-11	Index

THEORY OF OPERATION

T.O. 31P2-2FSQ7-2	Basic Theory of Computers AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-12	Introduction to AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-22	Basic Circuits for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-32	Theory of Operation of Central Computer for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-42	Theory of Operation of Drum System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-52	Theory of Operation of Input System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-62	Theory of Operation of Display System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-72	Theory of Operation of Output System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-82	Theory of Operation of Power Supply System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-92	Theory of Operation of Marginal Checking for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-102	Theory of Operation of Warning Light System for AN/FSQ-7, Combat Direction Central

INSTALLATION

T.O. 31P2-2FSQ7-5	Installation of AN/FSQ-7, Combat Direction Central
-------------------	---

OPERATING PROCEDURE

T.O. 31P2-2FSQ7-21	Operating Procedure for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-122	Operating Procedure for Maintenance of AN/FSQ-7, Combat Direction Central

LIST OF RELATED MANUALS (cont'd)*Manual**Title***MAINTENANCE**

T.O. 31P2-2FSQ7-132	Introduction and Philosophy of Maintenance for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-142	Maintenance Techniques and Procedures of Central Computer for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-152	Maintenance Techniques and Procedures of Drum System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-162	Maintenance Techniques and Procedures of Input System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-172	Maintenance Techniques and Procedures of Output System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-182	Maintenance Techniques and Procedures of Display System for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-192	Maintenance Techniques and Procedures of Power Supply and Marginal Checking for AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-202	Maintenance Techniques and Procedures of Warning Light System for AN/FSQ-7, Combat Direction Central

SCHEMATICS

T.O. 31P2-2FSQ7-212	Schematics for Central Computer of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-222	Schematics for Drum System of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-232	Schematics for Input System of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-242	Schematics for Output System of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-252	Schematics for Display System of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-262	Schematics for Power Supply and Marginal Checking of AN/FSQ-7, Combat Direction Central
T.O. 31P2-2FSQ7-272	Schematics for Warning Lights of AN/FSQ-7, Combat Direction Central

LIST OF RELATED MANUALS (cont'd)*Manual**Title***PLUGGABLE UNITS**

T.O. 31P2-2FSQ7-282 Pluggable Units for AN/FSQ-7,
 Combat Direction Central

PARTS CATALOG

T.O. 31P2-2FSQ7-4 Illustrated Parts Catalog for AN/FSQ-7,
 Combat Direction Central

SPECIAL TEST EQUIPMENT

T.O. 31P2-2FSQ7-31	Test Set, Memory Driver Panel (TS-986/FSQ)
T.O. 31P2-2FSQ7-41	Test Set, Plug In Units (TS-985/FSQ)
T.O. 31P2-2FSQ7-51	Power Supply PP-1581/FSQ
T.O. 31P2-2FSQ7-61	Test Set, Cathode-Ray Tube (TS-987/FSQ)
T.O. 31P2-2FSQ7-71	Test Set, Amplifier (TS-988/FSQ)
T.O. 31P2-2FSQ7-81	Dummy Load (DA-155/FSQ)
T.O. 31P2-2FSQ7-91	Test Set, Metallic Rectifier (TS-989/FSQ)
T.O. 31P2-2FSQ7-101	Test Set, Diode Semi-conductor Device (TS-990/FSQ)
T.O. 31P2-2FSQ7-111	Calibrator, Oscilloscope (FR-112/FSQ)
T.O. 31P2-2FSQ7-121	Marginal Check Control, Remote (C-2022/FSQ)
T.O. 31P2-2FSQ7-131	Test Set, Electron Tube (TV-11/FSQ)
T.O. 31P2-2FSQ7-141	Distribution Box (J-779/FSQ)

PART 1

EQUIPMENT INTRODUCTION

CHAPTER 1

INTRODUCTION

SECTION 1

SCOPE AND CONTENTS

This manual presents the information necessary for understanding the program-controlled operation of AN/FSQ-7 Combat Direction Central. The manual is divided into two parts: the first covers information basic to an understanding of programming; the second summarizes and suggests applications of this information to the programming of this equipment.

The first two chapters of Part 1 review the meaning of programming for digital computers and the basic concepts of number systems, with particular emphasis on the binary number system and the arithmetic of that system. The purpose and the function of the systems within AN/FSQ-7 Combat Direction Central are also described briefly. In Chapter 3, the computer portion of the equipment is described, first in general terms, then in terms of programmed operations and some of their applications. Chapter 4 describes the system through which most information passes to enter or leave the computer portion of the equipment. Each of the remaining chapters of Part 1 describes one of the special systems within AN/FSQ-7 Combat Direction Central on a block diagram level, pro-

viding programmers with a knowledge of the operation of each system. More detailed descriptions of each system's operation are available in the manuals on these systems.

Part 2 summarizes all information pertinent to the programming of AN/FSQ-7 Combat Direction Central and discusses the approaches that can be used in programming many of the required operations. Part 2 can therefore be used conveniently as a reference source for the preparation or interpretation of programs. Chapter 1 of Part 2 presents a general description of the overall program for AN/FSQ-7 Combat Direction Central and of each major block within that program. Chapters 2, 3, and 4 present (in summary form) the instructions for the equipment, the techniques and considerations useful in programming, and the programming for various input-output equipment. Each of the remaining chapters of Part 2 investigates a large block of the overall program and the means of accomplishing the desired result within each block.

SECTION 2

DIGITAL COMPUTERS

2.1 ADVANTAGES

Electronic digital computers are used in military applications because they offer the advantage of rapid calculation. A manually controlled calculator is limited in speed of calculation to the rate at which its operator can direct its operations. To escape this limitation, an electronic digital computer must be capable of controlling its own operations to achieve a result independent of manual control.

In order to operate automatically, an electronic digital computer must be capable of:

- a. Performing a relatively small number of simple operations.
- b. Executing the intermediate steps within each simple operation under its own control, in response to a code signal arbitrarily designating which of the possible operations is to be performed.
- c. Storing within itself the series of code signals that will designate the sequence of simple operations to accomplish the desired mathematical result.

2.2 DESIGN REQUIREMENTS

2.2.1 Basic Operations

Most mathematical operations, even those with a high degree of complexity, can be reduced to a series of simple arithmetic operations. Therefore, a digital computer that can perform the four basic arithmetic operations (addition, subtraction, multiplication, and division) can solve almost any mathematical equation.

2.2.2 Coded Instructions

The use of code signals to direct the digital computer in the solution of a problem is easily accomplished. For example, a desk calculator is directed to add, subtract, multiply, or divide by the depression of one of four pushbuttons. The ADD pushbutton directs the desk calculator to perform an addition. The pushbutton itself carries no meaning. The calculator is designed to respond to the depression of that pushbutton by performing an addition. The pushbutton is, in effect, a code signal to the calculator.

A digital computer can be designed to respond to each of a set of code signals by performing a different operation for each signal. These signals, known as instructions, must be separately recognizable by the digi-

tal computer. Since the digital computer must be capable of handling numbers, each instruction can be identified by assigning it a code number. Code numbers designating instructions to be performed by the digital computer have no quantitative significance. Instead, the code numbers identify instructions to the digital computer in the same manner as telephone numbers identify particular telephones in a dial telephone system.

2.2.3 Internal Storage

The third feature required for automatic computation is implicit in the nature of a high-speed digital computer. Without some means of storing internally both instruction and numbers to be processed, a digital computer is limited to manual insertion rates. This internal storage, generally referred to as memory, must be capable of delivering instructions and operands (numbers to be processed) at rates comparable to the rate at which the digital computer can execute these instructions. In general, the computation rate of a digital computer is determined by the random access time of its memory; i.e., the average time required for a transfer into or out of a random memory location.

2.3 PROGRAMMING

2.3.1 Introduction

Although a digital computer with the features required for automatic operation may be designed, some problems arise in its use. A digital computer is limited to the direct execution of a simple arithmetic operation. Some specialized knowledge is required in adapting mathematical expressions for solution in terms of these operations. Once the solution is developed, it must be formulated as a program, a series of instructions, for presentation to the digital computer. Personnel trained in meeting these needs are called programmers.

A programmer must prepare a program in terms of possible digital computer operations, guarding against the possibility that the digital computer will attempt a meaningless operation within a program, thus rendering useless the results of the program. For example, if a problem requires the division of the result of one calculation by the result of another calculation, the divisor may be zero, making the division meaningless mathematically. Unless the program provides for some check on this possibility and for an alternate course of calculation if the possibility occurs, the digital computer will execute the division operation and use the result in fur-

ther calculation. Therefore, the program must direct every operation and every choice of alternatives which the digital computer is to perform.

2.3.2 Program Definition

A program may be defined as a series of instructions, coded in a form recognized by the digital computer, calling for the operations to be performed by the digital computer in the order necessary to solve a given problem. Even the solution of a simple arithmetic problem requires a program, whether the problem is solved by a digital computer or by a man with pencil and paper. Although the man can recognize the necessary steps in a program if the problem is simple enough, the digital computer must be given step-by-step directions for the solution of any problem.

2.3.3 Examples

As an example of the solution of a problem, consider the evaluation of the expression

$$ax^2 + bx + c$$

when a , b , c , and x are given. The problem could be solved manually by performing the following calculation:

1. Multiply x by x .
2. Multiply x^2 by a .
3. Multiply x by b .
4. Add ax^2 to bx .
5. Add c to sum of ax^2 and bx .

The steps in this solution are not particularly simple. A number of intermediate operations would be required for a digital computer which can usually operate on only two numbers at a time.

The full sequence of instructions necessary for a digital computer in solving the problem as it was solved manually is as follows:

1. Bring x out of memory.
2. Multiply by x .
3. Multiply x^2 by a .
4. Store result in memory.
5. Bring b out of memory.
6. Multiply by x .
7. Bring previous result, ax^2 , out of memory and add to bx .
8. Add c to sum of ax^2 and bx .

An experienced programmer could use a shorter, and hence faster, series of instructions to accomplish the same result. By inspection, the expression

$$ax^2 + bx + c$$

can be rewritten as

$$(ax + b)x + c$$

and evaluated by this series of instructions:

1. Bring a from memory.
2. Multiply by x .
3. Add b to product.
4. Multiply by x .
5. Add c .

2.3.4 Iterative Programming

If the expression $ax^2 + bx + c$ is to be evaluated for many different values of x , the digital computer can be programmed to repeat its calculations for each value of x . The program directing the evaluation can simply repeat the series of instructions necessary for any one evaluation of x for each value of x to be used. For five values of x , 25 instructions are required. However, many digital computers avoid the necessity of writing out and storing all 25 instructions. Instead, the same five instructions are repeatedly used, once for each value of x . The repetitive calculations are controlled by several other instructions which count the number of repetitions remaining to be performed. The repeated series of instructions is called an iterative loop.

The use of an iterative loop in solving a problem requiring repetitive operations is highly desirable. The instructions within a program, as well as the numbers to be processed by that program, must be stored in memory. If the number of instructions within a program is large, the space within memory for operands is small. Therefore, the number of operations that can be performed before exhausting the supply of operands in memory is small, and calculation must be interrupted to place more operands in memory.

It should be noted that an entire program may be iterative. The last instruction may cause the digital computer to start again with the first instruction in the program. An iterative program is used to perform the same sequence of operations on each of several sets of numbers. One use of an iterative program is discussed in Section 3.

2.4 NEEDS OF PROGRAMMER

Personnel preparing programs for digital computers require specialized knowledge in several fields to meet the requirements of their task. A knowledge of higher mathematics is mandatory in order to reduce a mathematical problem to a series of simple operations within the capabilities of the digital computer. A general familiarity with digital computer operation is desirable, both in the task of formulating problems and in the prior step of learning the operation of the specific digital computer to be programmed. Included within this field of general familiarity with digital computer operations is a knowledge of the number systems used by and with digital computers and of the method of execution of the basic arithmetic operations by digital computers.

In addition to this general background knowledge, programmers require information in three specific areas:

- a. The possible operations of the digital computer to be programmed and the instructions that call for these operations.
- b. The requirements placed upon the programming of the digital computer by any special input or output equipment used with the digital computer.
- c. The type of problem to be solved by the digital computer being programmed.

The programmer must know what the digital computer can do and what it cannot do. The programmer must know the features of the digital computer that can reduce the time required to execute a program, thus extracting the greatest advantage from the high operat-

ing speed of the digital computer. To write up a program, the programmer must know the instructions recognized by the digital computer; i.e., the machine language.

If the digital computer being programmed uses special input and output equipment, the programmer must know what requirements this equipment places upon programming. Specifically, the programmer must know how to write a program which can most rapidly obtain information from the input equipment and then prepare processed information for presentation to the output equipment.

A knowledge of the type of problems to be solved is patently necessary before these problems can be formulated in terms of possible digital computer operations.

SECTION 3

AN/FSQ-7 COMBAT DIRECTION CENTRAL

3.1 PURPOSE OF EQUIPMENT

3.1.1 Air Defense Problem

The equipment designated as AN/FSQ-7 Combat Direction Central is the result of applying automation techniques to the problem of defense against air attack. The defense of an area against air attack requires:

- a. Compilation of information from many sources on air movements within the area.
- b. Processing of this information for presentation in a common form.
- c. Sorting and presentation of the processed information to tactical air defense personnel.
- d. Provisions for direction of air defense weapons by tactical personnel on the basis of the information presented.

The operations involved in an air defense problem can be classified as data simplification operations; i.e., operations that reduce a large body of discrete items of information into a smaller, better-organized, more comprehensible body of information.

The operations carried on by AN/FSQ-7 Combat Direction Central during the processing of an air defense problem are of several types. Each type of operation is controlled by a separate subprogram. All the subprograms taken together constitute the air defense program.

The air defense program may be divided into two major blocks, an air surveillance program and a weapons direction program. In general terms, the air surveillance portion of the program compiles and processes information on air movements, while the weapons direction portion of the program utilizes this information in directing the defense of the area.

3.1.2 Iterative Air Defense Program

In order to employ a digital computer, which works with discrete numbers or measurements, in processing continuously changing information on airplanes in flight, a sampling technique must be used. In effect, the air defense program accepts one report on each airplane within the defense area during a specific time interval to produce a still picture of the air movements within the area. During the next time interval, and during each succeeding time interval, another still picture of air movements is assembled. This series of still pictures provides information on the motion of aircraft within the

defense area in much the same way as a motion picture camera achieves the illusion of motion by taking a series of still pictures, called frames, at regular short intervals. To accomplish this sampling, the air defense program must be an iterative program, one which is repeated during each sampling interval. The sampling interval is called a program frame.

The time allowed for a program frame must be long enough to obtain a complete picture of air movements within the entire defense area. This time is related to the scan rate of the radar sets supplying information to AN/FSQ-7 Combat Direction Central. The time allowed for one program frame must also be sufficient to complete one iteration of the air defense program, which includes an immense number of operations.

3.1.3 Sample Program Routine

As an example of the operations included within the air defense program, consider the conversion of radar target reports from different radar sets into a common form for presentation. Each radar set reports its targets in polar co-ordinate form; namely, range from the radar set to the target and the azimuth angle from radar north clockwise to the radar target line. All reports must be converted into rectangular co-ordinate form and shifted onto some reference co-ordinate system common for all radar sets. While the conversion cannot yet be discussed in terms of the operation of this equipment, it can be followed mathematically. Figure 1-1 shows the conversion to be accomplished.

A radar set furnishes data designated as R , range, and θ , azimuth angle. This polar co-ordinate representation must first be converted to X_{tr} and Y_{tr} , rectangular co-ordinates based on the radar set. Mathematically, the conversion is accomplished as follows:

$$X_{tr} = R \sin \theta \quad Y_{tr} = R \cos \theta \quad (1)$$

If the location of the radar set with respect to the common reference point is known, the target location with respect to the reference point may be calculated as follows:

$$X_t = X_{tr} + X_r \quad Y_t = Y_{tr} + Y_r \quad (2)$$

The time allowed to perform this conversion for all radar reports and to perform all other operations within a program frame is approximately 15 seconds.

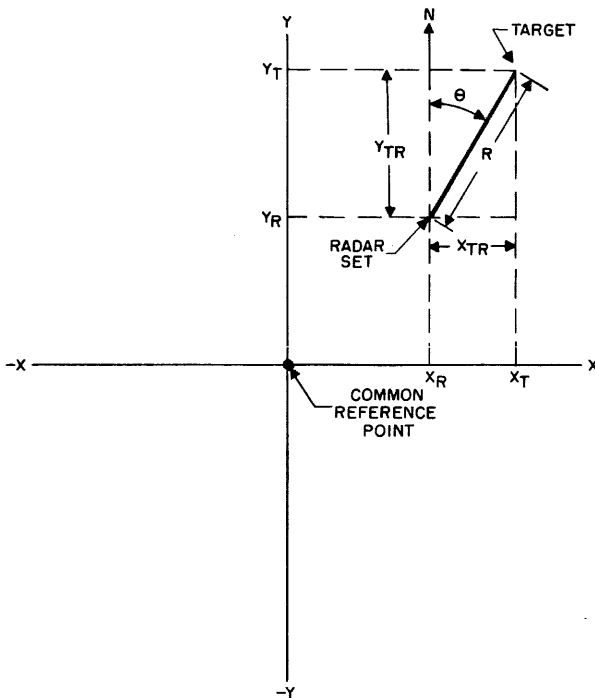


Figure 1-1. Radar Co-ordinate Conversion

3.2 EQUIPMENT FUNCTIONS

3.2.1 Component Systems

The equipment included in AN/FSQ-7 Combat Direction Central is divided into six operational systems (as shown in fig. 1-2):

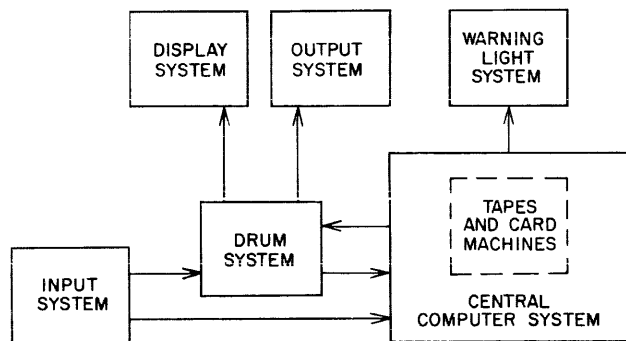


Figure 1-2. AN/FSQ-7 (DC-1) Combat Direction Central Information Flow

- a. Central Computer System
- b. Input System
- c. Drum System
- d. Display System
- e. Output System
- f. Warning Light System.

To introduce the programmer to this equipment, a brief description of each system follows:

3.2.2 Central Computer System

3.2.2.1 Logical Divisions

The Central Computer System performs all data-processing involved in the air defense problem under control of the air defense program. As shown in figure 1-3, the Central Computer System is divided into six logical elements:

- a. Core memory element
- b. Arithmetic element
- c. Program element
- d. Instruction control element
- e. Selection control element
- f. Maintenance control element.

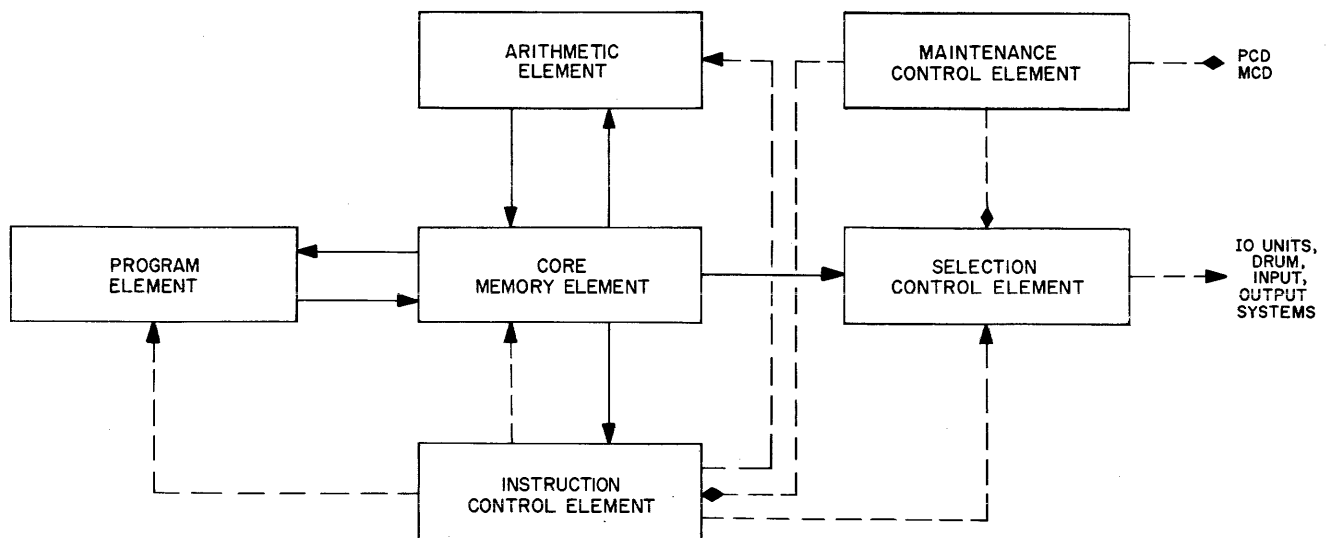
3.2.2.2 Element Functions

The core memory element provides internal storage, for the program followed by the Central Computer System, for raw information and for processed information ready for presentation to the output equipment. The core memory element provides storage for 8,192 binary words (binary numbers of fixed length) with a random access time for obtaining a given word of 6 microseconds; i.e., 6 microseconds must elapse between successive transfers into or out of core memory.

The arithmetic element performs the actual processing operations, arithmetic and nonarithmetic, carried on by the Central Computer System. It receives raw data from core memory and returns the result of its operations to core memory for temporary storage prior to presentation to output equipment. The arithmetic element operates on numbers in parallel rather than serially. For example, addition is done by adding the corresponding order digits of two numbers simultaneously in separate circuits rather than sequentially in the same circuit. This parallel operation, although requiring more equipment than serial operation, allows higher computing speeds.

The type of information handled within the processing of an air defense problem is such that dual operation (simultaneous manipulation of the two halves of a polar or rectangular co-ordinate) materially reduces computation time. The arithmetic element contains two arithmetic units, each capable of separate, simultaneous calculation. Thus, the radar co-ordinate conversion, discussed in 3.1.3, can be handled more rapidly by performing each of the two calculations contained in equation (1) or in equation (2) simultaneously in the two arithmetic units.

The step-by-step operations of the arithmetic element are directed by commands from the instruction control element. Each command causes the execution of one intermediate step in an operation called for by



an instruction. Thus, the instruction control element receives a program instruction, recognizes the operation to be performed, and generates the series of commands necessary to execute, in the correct order, all the intermediate steps of that operation.

Instructions are delivered for decoding to the instruction control element under the control of the program element. The instructions comprising a program are stored in sequential locations in core memory. As the operation called for by an instruction is completed, the program element directs the transfer of the next instruction from core memory to the instruction control element. An operand required for a particular operation is transferred from core memory to the arithmetic element as directed by the program element during the execution of an instruction. Further, the program element contains the circuits necessary for the control of an iterative loop. (Refer to 2.3.4.) In general, the program element controls all internal transfers of information (transfers between elements of the Central Computer System) and all internal aspects of input-output (IO) transfers of information.

The external aspects of IO transfers are controlled by the selection control element. These functions include the selection, and mechanical operation if necessary, of a specific unit for an IO transfer. They also include selection of units for sensing of certain conditions within the Central Computer System. For example, the selection control element contains provisions for examining the result of a calculation held in the arithmetic element. The examination may be used to prevent a meaningless operation with that calculated result. (Refer to 2.3.1.)

Included within the Central Computer System are several units which can supply information to, or receive

information from, the Central Computer System. (See fig. 1-2.) These units include six magnetic tape units, a card reader, a card punch, and a line printer. These units, known generally as included IO units, offer unlimited storage space at the expense of long access time. Therefore, these units are used only for bulk transfers of information when no time limitation exists. For example, programs are usually loaded initially from the card reader. The line printer may be used to obtain a printed tabulation of a program or of processed information. Generally, none of these units is used during the execution of an air defense program because of their low speed of operation.

The maintenance control element is used primarily in testing the Central Computer System. However, this element also includes controls used in loading programs into core memory and in determining the response of the Central Computer System to certain errors and alarm conditions.

3.2.2.3 System Information Flow

There are two general classes of information flow in the Central Computer System, internal information flow and IO information flow. The internal information flow paths are shown in figure 1—4. For each operation performed by the Central Computer System, an instruction must be obtained from core memory. The program element specifies the address in core memory from which the instruction is to be taken. The instruction is transferred in two parts: one part to the instruction control element; the other part to the program element. In the instruction control element, the operation called for by the instruction is identified, and the necessary commands are generated. If the operation requires an operand from core memory, the program element receives a com-

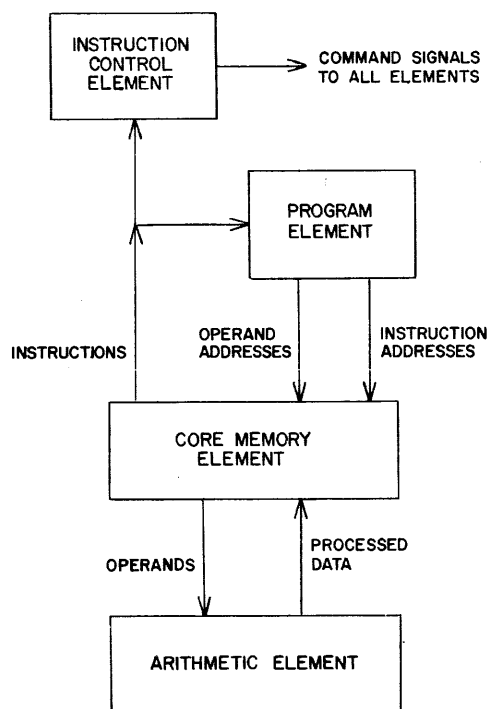


Figure 1-4. Internal Information Flow

mand causing it to specify the address in core memory from which the operand is to be taken. The operand is transferred from core memory to the arithmetic element, where it is operated on in accordance with the commands generated in the instruction control element. Upon completion of the operation, the program element receives a command from the instruction control ele-

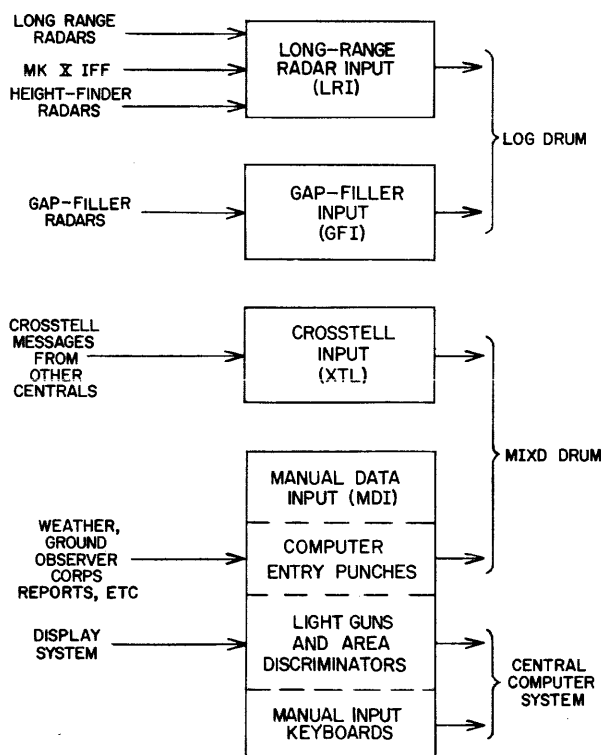


Figure 1-6. Input System Information Flow

ment to obtain the next instruction from core memory. It should be noted that instructions may cause the transfer of information from the arithmetic element back to core memory, or initiate the transfer of information between core memory and an IO unit.

Input-output operations are generally carried on as block transfers; i.e., a group of information units is assembled for transfer and then transferred as rapidly as possible in one programmed operation. Input-output information flow paths are shown in figure 1-5. Each IO operation requires a series of preparatory instructions to perform the following functions:

- To notify the selection control element which IO units is to be used
- To indicate to the program element the first core memory address to be involved in the IO operation
- To specify the direction of transfer and the amount of information to be transferred.

Upon initiation of the IO operation proper, information words are transferred between core memory and the selected IO unit via intermediate transfer registers located in the program and arithmetic elements. The IO units operate slowly enough to allow the processing of information between steps in an IO transfer. Thus, the Central Computer System continues to process information between successive IO word transfers that bring in

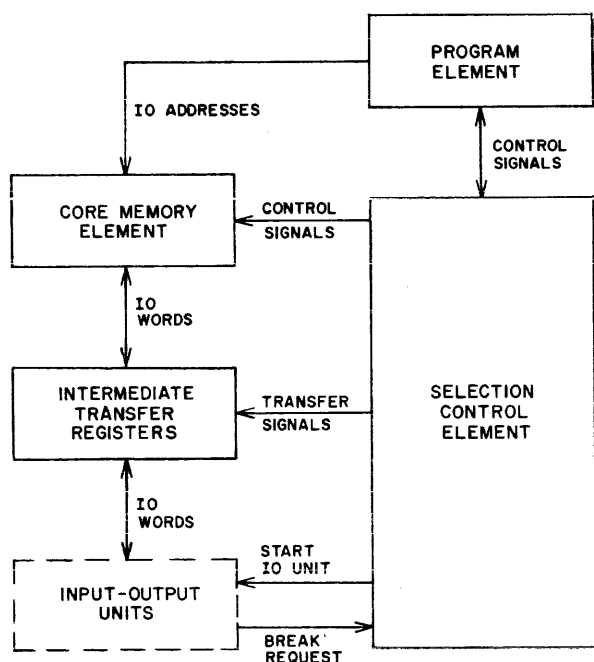


Figure 1-5. IO Information Flow

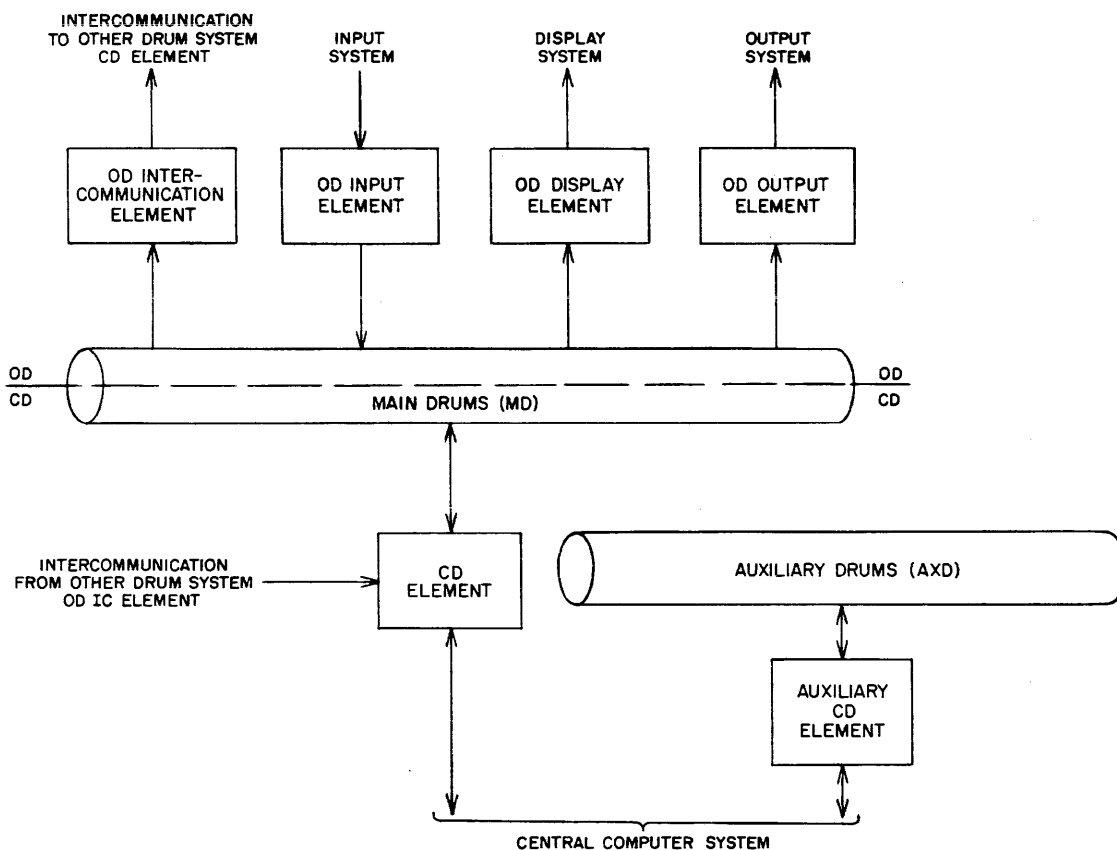


Figure 1-7. Drum System Information Flow

raw information or deliver processed information to other systems.

3.2.3 Input System

All information on air movements is supplied to the Central Computer System by the Input System. Figure 1-6 shows the four elements of the Input System:

- Long-range radar input (LRI)
- Gap-filler input (GFI)
- Crosstell (XTL) input
- Manual data input (MDI).

The bulk of the information supplied to the Central Computer System is received by the LRI and the GFI elements. These two elements receive target reports from radar sets, convert them into forms suitable for use in the Central Computer System, and supply them to the Drum System. (Refer to 3.2.4.) The Drum System assembles these randomly received messages for a rapid block transfer of many reports to the Central Computer System in one IO operation.

Reports from other Combat Direction Centrals are received by the XTL input element. These reports are also assembled in the Drum System for a block IO transfer into the Central Computer System.

The manual data input element handles three different types of information. Verbal messages reporting aircraft, weather conditions, or any other miscellaneous information are prepared on punched cards and read into the Drum System through three computer entry punches. The other two types of information handled by the manual data input element are closely related to the Display System and are discussed with that system.

3.2.4 Drum System

The Drum System performs two functions in AN/FSQ-7 Combat Direction Central. It acts as a time buffer between the rapidly operating Central Computer System and the slower Input, Output, and Display Systems. The Drum System also provides added memory capacity to supplement the capacity of the core memory element of the Central Computer System. Figure 1-7 shows the operational elements of the Drum System. (Three elements not shown are used solely for test purposes.)

The two general types of information flow involving the Drum System comprise CD (Computer System-Drum System) operations and OD (other than Central Computer System-Drum System) operations.

Elements on the OD side operate independently of each other and of elements on the CD side of the drums. Thus, the OD input element may be receiving data from

the Input System while the OD display element transfers data to the Display System, the OD output element supplies data to the Output System, and the OD intercommunication element transfers data to the other Drum System. (Refer to 3.3.) Simultaneously with all these operations, one of the CD elements may be handling an information transfer between the Drum System and the Central Computer System.

The drums shown pictorially in figure 1-7 are divided physically into two groups of six drums each:

- a. Main drum (MD) group
 1. LOG drum
 2. MIXD drum
 3. Track display (TD) drum
 4. Radar data (RD) drum
 5. Auxiliary memory A (AM-A)
 6. Auxiliary memory B (AM-B)
- b. Auxiliary drum (AXD) group
 1. Auxiliary memory C (AM-C)
 2. Auxiliary memory D (AM-D)
 3. Auxiliary memory E (AM-E)
 4. Auxiliary memory F (AM-F)
 5. Auxiliary memory G (AM-G)
 6. Auxiliary memory H (AM-H).

With one exception, each physical drum is divided into six logical fields. (The RD drum contains nine fields.) Each field has a storage capacity one-quarter the capacity of core memory. Total drum storage capacity is

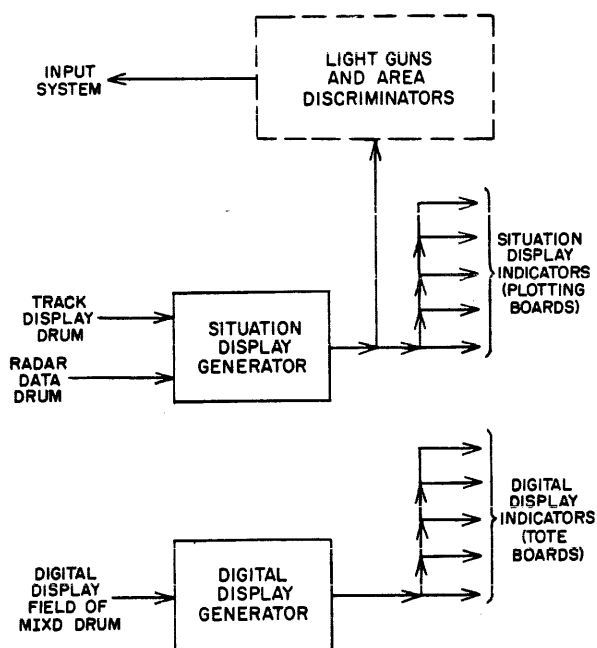


Figure 1-8. Display System Information Flow

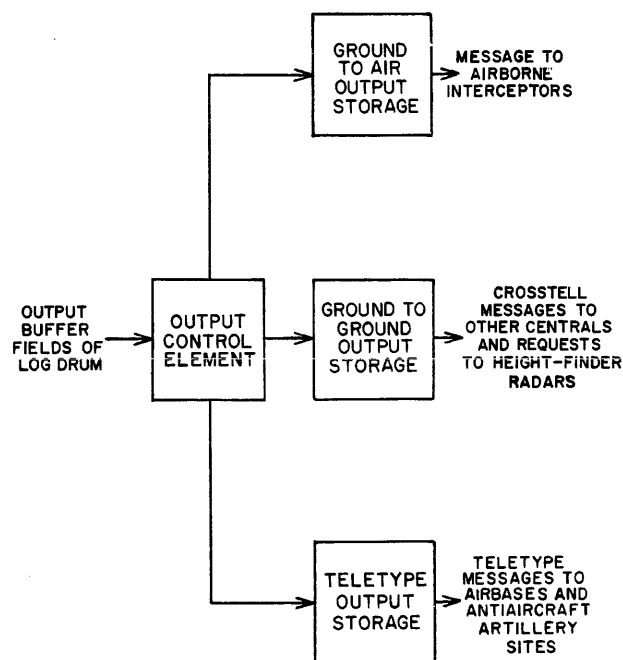


Figure 1-9. Output System Information Flow

approximately twenty times that of core memory. The CD average access time for the Drum System is 10 milliseconds, as compared with 6 microseconds for core memory. However, successive CD information transfers can occur at 10-microsecond intervals on addressable drums, almost as rapidly as successive transfers from core memory (6-microsecond intervals).

Auxiliary memory drums A through H provide added memory capacity for the Central Computer System. The LOG drum (named from the initial letters of the logical field names contained thereon) includes two fields for information from the LRI element, three fields for information for the Output System, and one field for information received from the GFI element.

The MIXD drum includes one field of manual data input element information, one field for supplying information to the OD intercommunication element, one field for XTL element information, and one field for Display System information. The remaining two fields on this drum are spares and are usable for auxiliary memory storage. The LOG, MIXD, TD, and RD drums perform time buffer functions for information transfers between the Central Computer System and the other systems.

3.2.5 Display System

The Display System provides for visual observation of the data processed by the Central Computer System. It receives this information from the Drum System at a

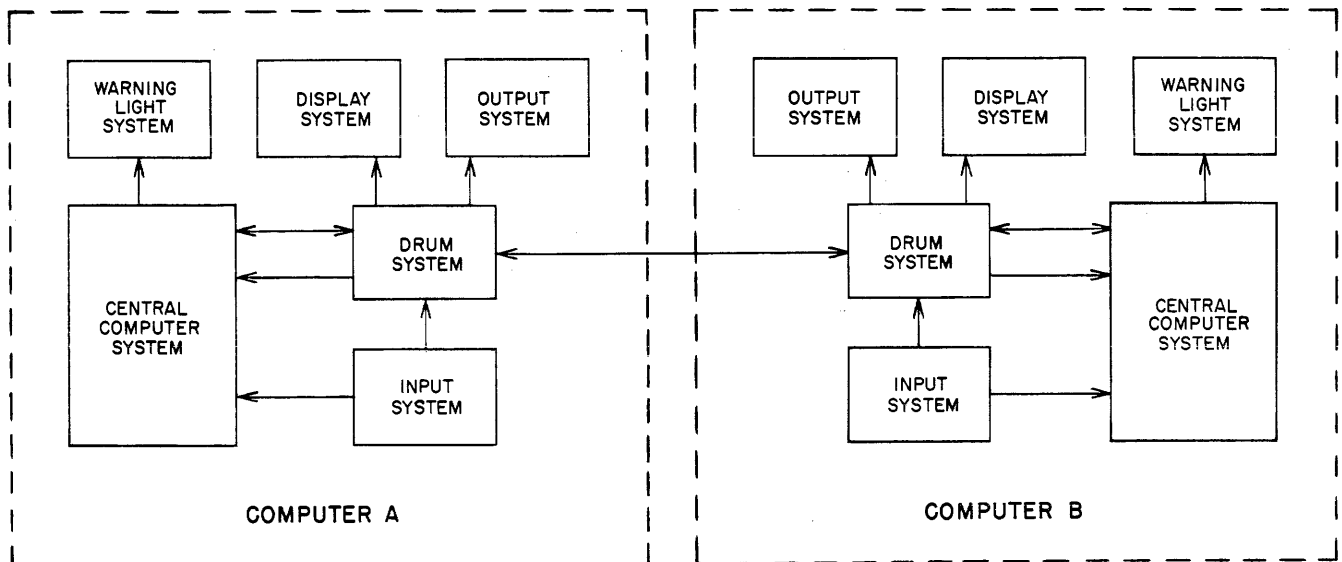


Figure 1-10. Duplexing, Simplified Block Diagram

rate that can be handled by the Display System equipment. The Display System generates displays from the binary information prepared by the Central Computer System. The blocks shown in figure 1-8 handle different types of information. Situation display handles air movement information, which requires the displays to change as the information changes. A situation display is equivalent to the plotting board display of a manual air defense center. Digital display handles statistical information or information summaries that need not be changed as rapidly as situation displays. The digital display is equivalent to a tote board display.

The Display System is closely related to the Input System through the manual data input element. At almost every console is a keyboard, part of the manual data input element. The air defense personnel at a console may direct the action of the Central Computer System, on the basis of information displayed, by setting up a message on the manual input keyboard. (See fig. 1-6.) If a specific target is to be reported back to the Central Computer System, information on that target may be transferred out of the situation display subsystem (by a light gun or by an area discriminator) through the manual data input element to the Central Computer System. (See fig. 1-8.)

3.2.6 Output System

Among other functions, the programs of the Cen-

tral Computer System make up messages directing the actions to be taken against hostile aircraft. These messages are presented to the Output System via the output buffer fields of the Drum System. In the Output System, shown in figure 1-9, each message is processed for transmission to a specific receiving station. Three types of messages are processed by the Output System:

- Ground-to-air, to airborne interceptors
- Ground-to-ground, to other centrals
- Teletype, to manual centrals, airbases, etc.

3.2.7 Warning Light System

The Warning Light System provides the means for the Central Computer System to notify operators at various units in AN/FSQ-7 Combat Direction Central of situations that require their attention or action. The Warning Light System controls neon indicators and audible alarms located at the various consoles of the Display System and at consoles of the GFI element. As shown in figure 1-2, the Warning Light System is controlled directly by the Central Computer System.

3.3 DUPLEXING

The equipment within each system discussed so far is duplicated in whole or in part within AN/FSQ-7 Combat Direction Central to attain the greatest possible operational reliability for that equipment. This duplex equipment is contained in two groups designated com-

puter A and computer B. (See fig. 1-10.) Computer A and computer B each contain an entire Central Computer System, Drum System, Output System, and those elements of the other three systems whose failure would render all systems useless if not replaced or repaired. For example, since the failure of a generator element in the Display System renders the entire Display System inoperative and limits the usefulness of the entire equipment, the generator elements are duplex (duplicated in computer A and computer B). On the other hand, the failure of one display console will not render the entire Display System inoperative. Therefore, display consoles are not duplex. Instead, spare display consoles are available as substitutes in case of failure of a console. Non-

duplex units are known as simplex equipment and are not shown in figure 1-10. They are discussed in succeeding chapters on the systems within AN/FSQ-7 Combat Direction Central.

It should be noted that only one set of duplex equipment is processing raw air defense data at any one time. This set, called the active computer, may be either computer A or computer B. The nonactive computer, called the standby computer, may be undergoing maintenance or, if operative, may be performing supplementary data-processing operations. The intercommunication fields of the two Drum Systems allow for interchange of information between the active and the standby computers.

CHAPTER 2

NUMBER SYSTEMS

SECTION 1

GENERAL

1.1 THE CONCEPT OF NUMBER

There exist two general definitions of number. In one definition, a number denotes a quantity or amount of units. For example, the term dozen denotes a quantity or amount of eggs, rolls, or any other unit items. Similarly, the symbols 12 and XII denote the same quantity. These three symbols (dozen, 12, and XII), although each is from a different system of notation, all designate the same amount of unit items.

The other general definition describes a number as a member of an ordered set of symbols. Although the number in an ordered set of numbers is capable of denoting quantity, the idea of quantity is unimportant. For example, a house or telephone number is a member of an ordered set of symbols. A house number identifies the house to which it is assigned and generally locates that house with respect to other numbered houses. House number 26 gives no idea of the size of the house or the quantity of houses. The number does imply that a house, numbered 26, lies between the houses numbered 24 and 28 on the same street and is distinct from them. The number 26 gives the address of the house; the address locates something and asserts its distinction from other addresses.

A telephone number, like a house number, gives no indication of quantity but rather distinguishes a particular telephone from other telephones. A telephone number, unlike a house number, contains no implication of location. Thus, while both a house number and a telephone number belong under the definition of a number as a member of an ordered set of symbols, only the house number can properly be called an address. The telephone number must be called an identification code, instead. There are, therefore, three applications of numbers that must be considered:

- a. Quantity designators
- b. Addresses
- c. Identification codes

The application of numbers to the designation of quantity will be discussed most completely. The demonstrations using this application are valid in the other two applications as well.

1.2 DECIMAL NUMBER SYSTEM

1.2.1 General Properties

When the decimal number 333 is read by anyone familiar with the decimal number system, it is read as three hundred thirty-three. Although the digits in each position are identical in appearance, their positions indicate their relative magnitude. The leftmost 3 is in the hundreds column, the next 3 in the tens column, and the last 3 in the units column. Thus the number 333 is an abbreviation of $300 + 30 + 3$. The advantage of using position to represent magnitude becomes apparent when the Roman numeral designation of the same quantity, CCCXXXIII, is examined. In this notation, a different symbol is used for hundreds, still another for tens, and yet another for units. Further, the order in which these symbols are written does not indicate their magnitude; rather, the magnitudes of the symbols determine the order in which the symbols are written. The Roman numeral system requires a more unwieldy set of symbols than does a system employing positional notation of magnitude.

1.2.2 Positional Notation of Magnitude

Positional notation of magnitude in the decimal number system can be described this way. Each position to the left of the decimal point represents a different positive power of 10 with the power increasing by 1 from each position to the next leftward position. The first position is known as the units position; $10^0 = 1$. The second position is the tens position; $10^1 = 10$. Similarly, each position to the right of the decimal point represents a negative power of 10 with the negative power increasing in absolute magnitude by 1 from each position to the next rightward position. Thus the number 2,437.923 actually represents the sum of the terms: $2 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 7 \times 10^0 + (.9 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3})$. Decimal numbers are seldom written out in this form since the convention of positional notation of magnitude is generally understood. In scientific calculations, use is made of the convention to abbreviate long numbers. The velocity of light can be written as 299,800,000 or abbreviated as 2.998×10^8 meters per second.

1.2.3 Radix

The decimal number system is so called because it uses the base 10 in counting. In any position in a decimal number, one of 10 symbols is admissible. These symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. When counting, adding a 1 to any number from 0 through 8 gives the next number in the series. Add a 1 to 9, however, causes a return in that position to 0 and the appearance of a 1 in the next higher value position. On every tenth count, the stepping of the next higher column from 0 to 1 and the return of the preceding column from 9 to 0 repeats. The decimal number system thus works with the base or radix 10. The term radix denotes both the number of steps in the counting cycle of each position before a step to the next higher column is necessary and the number of symbols from which one symbol may be chosen to appear in a given position.

The decimal number system uses the same radix in each position. This is not a necessary feature of a number system. For example, the English system of linear measure uses radix 12 in the first (inches) position, radix 3 in the second (feet) position and radix 1760 in the third (yards) position, with each radix expressed decimally. Since a number system using several radices is not as easy to manipulate as is a system using a single radix, number systems used in calculations are based on a single radix.

1.2.4 Modulus

Just as the radix of a number system specifies certain properties of each digit position, the modulus specifies certain properties of a given combination of digit positions. If a list of all the decimal numbers that can be written in m digit positions is prepared, it will be found that there are 10^m numbers in the list, ranging in value from 0 to $10^m - 1$. This number of numbers which can be written in a limited number of digit positions is called the modulus of that number of digit positions.

For example, the mileage indicator on an automobile uses the radix 10 in each digit position. Most mileage indicators provide six digit positions. They are thus modulo 10^6 decimal counters. The largest number which such a counter can represent is 999999 or $10^6 - 1$. However, most mileage indicators place a decimal point to the left of the last significant digit. The largest distance the counter can represent is therefore 99,999.9 miles. The mileage indicator counts in tenths of miles. After counting 10^6 tenths of miles, its represented contents return to 0. Thus, the radix specifies both the number of counts, R , between 0 and a return to 0 and the largest count, $R - 1$, in a single digit position, while the modulus specifies both the number of counts, R^m , between 0 and a return to 0 and the maximum count, $R^m - 1$, in a group of m digit positions. The significance of the

modulus becomes apparent upon considering the number represented on a mileage indicator after the automobile has traveled a distance greater than the modulus. The distance indicated is the difference between the actual distance and the modulus. If the distance travelled is 105,000.0 miles, the indicator will show only 5,000.0 miles. More generally stated, a modulo M counter (where $M = R^m$) cannot recognize the difference between two numbers which differ in magnitude by integral multiples of M . Thus, for the mileage indicator used as an example, the numbers 105,000.0, which is $M + 5,000.0$, and 5,000.0 are indistinguishable.

1.3 GENERAL NUMBER SYSTEM

The two features of the decimal number system discussed above, positional notation of magnitude and radix, are features of most number systems. This discussion is limited to systems using a constant radix but, with slight modification, is applicable to other number systems.

The most general expression of a number, constructed in a system employing the two features mentioned, is as follows:

$$N_p \times R^p + N_{p-1} \times R^{p-1} + \dots + N_0 \times R^0 (.) N_{-1} \times R^{-1} + \dots + N_{-q} \times R^{-q}$$

Where R = radix

N = any integer from 0 through $R - 1$

(.) = radix point

p = most significant position and power of that position

q = least significant position and power of that position

When R is specified, the expression applies to the number system based on the radix. If R is specified as 10 as in the decimal system, N can be replaced by one of the integers 0 through 9 and (.) becomes the decimal point.

If it were desired to construct a number system using the radix 7 for example, N could be replaced by one of the integers 0 through 6, (.) would become the septenary point, and positions would ascend or descend in significance by powers of 7. The decimal number 49 would be written in septenary form as 100; i.e., $1 \times 7^2 + 0 \times 7^1 + 0 \times 7^0$. The two numbers, 49_{10} and 100_7 , represent the same quantity. (Subscript indicates radix.) They are different notations of equal validity, just as twenty in English and vingt in French are different notations of equal validity for the same thing.

SECTION 2

BINARY NUMBER SYSTEM

2.1 INTRODUCTION

The binary number system is important in the understanding of digital computers for one simple reason. The binary number system, with a radix of 2, utilizes combinations of only two allowable digits to represent any number. Therefore, a binary number can be represented in a computer by the settings of a group of two-position switches, by a series of pulses and no-pulses on a transfer line, or by the settings of a group of flip-flops with the two possible conditions of each flip-flop defined, respectively, as 0 and 1.

Some of the mystery of an unfamiliar system of notation may be resolved by counting binarily from 0 through decimal 10:

Decimal	0	1	2	3	4	5	6	7	8	9	10
Binary	0	1	10	11	100	101	110	111	1000	1001	1010

The decimal and binary notations for 0 and 1 are identical. The radix of the binary number system, however, is expressed binarily just as is the radix of the decimal system expressed decimally as 10. Similarly, 4 (2^2) and 8 (2^3) are expressed binarily as 100 and 1000 respectively.

In Section 1, a definition for a general number system is given as equation (1). Limiting this definition to apply to the binary number system yields:

$$B_p \times 2^p + B_{p-1} \times 2^{p-1} + \dots B_0 \times 2^0 + (.) \\ B_{-1} \times 2^{-1} + \dots B_{-q} \times 2^{-q} \quad (2)$$

With the radix specified as decimal 2, B can be either 0 or 1. The radix point (.) becomes the binary point and, as in any number system, separates positive powers of the radix from negative powers.

As an example of binary notation, consider binary 101111.01 which is equivalent to decimal 47.25. The definition of the binary number system, equation (2) above, can be used to demonstrate the equivalence of the two notations. Decimal notation is used in this demonstration.

$$\begin{aligned} 47.25_{10} &= 101111.01_2 \\ &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 \\ &\quad + 1 \times 2^0 + (.) 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 32 + 0 + 8 + 4 + 2 + 1 + 0 + 1/4 \\ &= 47.25 \end{aligned}$$

2.2 CONVERSION FROM DECIMAL TO BINARY NOTATION

2.2.1 Conversion by Definition

A decimal number may be converted to binary notation using the definition given in equation (2). By inspection, determine the highest power of 2 which is smaller than or equal to the decimal number. For example, in converting decimal 26 to binary notation, inspection reveals that 2^4 or 16 is the highest power of 2 which is yet smaller than or equal to 26. Therefore the binary notation must be of the form, 1----. The difference between the decimal number and the decimal value of the power of 2 chosen by inspection is subjected to the same sort of inspection. In this case the remainder is 10 ($26 - 16$). The highest power of 2 smaller than or equal to 10 is 2^3 or 8. The binary notation of decimal 26 now has the form, 11----. The remainder ($10 - 8$) is equal to 2^1 or 2. Thus, the decimal number 26 has the binary notation, 11010.

2.2.2 Systematic Conversion

2.2.2.1 Integral Decimal Numbers

A more systematic method of conversion of integral decimal numbers to binary notation involves the use of successive divisions of the decimal number by the radix of the binary system expressed decimally. Decimal 26 is again used as the example in the procedure shown in table 1-1.

TABLE 1-1. INTEGRAL DECIMAL TO BINARY CONVERSION

DECIMAL NUMBER	$\div 2 =$ QUOTIENT	REMAINDER
26	13	0 (least significant digit)
13	6	1
6	3	0
3	1	1
1	0	1 (most significant digit)

The remainders written from right to left in the order obtained represent the binary notation of the given decimal number. Thus:

$$26_{10} = 11010_2$$

2.2.2.2 Fractional Decimal Numbers

A fractional decimal number may be converted to binary notation by a process similar to that used for conversion of integral decimal numbers. Instead of successive divisions by the binary radix in decimal form, the decimal fraction is successively multiplied by the binary radix in decimal form. Each multiplication yields an integral and a fractional part as the product. The integral part is taken as the binary digit while the fractional part is again multiplied by decimal 2. Using decimal 0.625 as the example, the conversion process is shown in table 1-2.

TABLE 1-2. FRACTIONAL DECIMAL TO BINARY CONVERSION

DECIMAL		PRODUCT	
NUMBER	x 2 =	INTEGRAL	FRACTION
0.625		1	.25
0.25		0	.50
0.50		1	.00

The result of the conversion is obtained by writing the integral parts from left to right after the binary point in the order obtained. Thus:

$$0.625_{10} = 0.101_2$$

2.2.2.3 Mixed Decimal Numbers

A mixed decimal number (containing an integral and a fractional part) is converted to binary notation by treating the two parts independently. Thus, to convert 26.625_{10} to binary notation, the procedures shown in tables 1-1 and 1-2 are followed, yielding:

$$26.625_{10} = 11010.101_2$$

2.2.2.4 Justification

These systematic methods of conversion from decimal to binary notation are based on this principle: Given a mixed number (containing an integral and a fractional part) expressed in both notations, both

notations for the integral part are equal to each other and both notations for the fractional part are equal to each other. Thus, using the example just given:

$$26.625_{10} = 11010.101_2$$

$$26_{10} = 11010_2$$

$$.625_{10} = .101_2$$

When the integral part of the decimal number is divided by 2, a mixed number results. This mixed number is the integral quotient and a remainder of the value $1/2$ or $0/2$. The remainder must equal the fractional part of the binary number produced by dividing it by 2. For example:

$$26 \div 2 = 13 + 0/2$$

$$11010_2 \div 2 = 1101_2 + .0_2$$

$$0/2 = .0_2$$

Division of the integral quotient by 2 will yield the least significant bit of its binary equivalent, and so forth.

When a decimal fraction is multiplied by 2, the result is a mixed number containing an integral part, equal to 1 or 0, and a fractional part. The integral part must equal the integral part resulting from multiplication of the binary equivalent by 2. For example:

$$.625 \times 2 = 1 + .25$$

$$.101_2 \times 2 = 1_2 + .01_2$$

$$1 = 1_2$$

Multiplication of the remaining fraction by 2 will again yield the most significant bit of the binary equivalent of the remaining fraction, and so forth.

2.3 CONVERSION FROM BINARY TO DECIMAL NOTATION

It has been shown that a number in binary notation may be converted to decimal notation using equation (2). Conversion using this method employs decimal arithmetic. It is also possible to convert from binary to decimal notation using binary arithmetic. This conversion follows the pattern used in conversion from decimal to binary notation described in 2.2.

SECTION 3

BINARY ARITHMETIC

3.1 GENERAL

Binary arithmetic is identical in principle to decimal arithmetic. The procedures of binary arithmetic however are unlike those of decimal arithmetic to the extent that binary notation differs from decimal notation. This Section is devoted to descriptions of the four basic arithmetic operations, addition, subtraction, multiplication, and division, as they are performed binarily. Although no special attempt is made in this Section to describe these operations exactly as they are performed in AN/FSQ-7 Combat Direction Central, it is desirable to describe these operations within the limitations imposed upon their execution by the design of that equipment. The description of these operations as performed by AN/FSQ-7 Combat Direction Central is presented in Chapter 3.

3.2 ARBITRARY LIMITATIONS

3.2.1 Design Limitations

The design of the Central Computer System of AN/FSQ-7 Combat Direction Central imposes certain limitations upon the execution of basic arithmetic operations. The physical size of the computing elements of the Central Computer System places a limit upon the number of significant figures within a binary number. The limit is set at 16 bits (binary digits) and each 16-bit group is designated a half-word. (The designation, word, is reserved for a 32-bit group which contains two half-words, one half-word for each half of the dual arithmetic element of the Central Computer System.)

3.2.2 Sign Convention

Although 16 bits are available in each half-word for binary notation of a quantity-designating number, only 15 bits are used for the representation of the magnitude of the number (modulo 2^{15}). One bit, called the sign bit, is reserved for the representation of the sign of the number. By convention, a 0 in the sign bit position indicates the number is positive; a 1 in that position indicates the number is negative. This convention is not entirely arbitrary as will be seen in 3.4.3.1.

3.2.3 Fixed Binary Point

The sign bit is distinguished from the remaining bits in a half-word by fixing the position of the binary point between the sign bit and the remaining bits in the half-word. Since the sign bit is placed in the leftmost position in the half-word, the magnitude of the number

designated by the half-word is restricted to fractional values lying between decimal $+1$ and -1 . The form of the half-word is thus:

Bit position:	S	.	1	2	3	4	...	14	15
Significance:	Sign bit		2^{-1}	2^{-2}	2^{-3}	2^{-4}		2^{-14}	2^{-15}

One advantage of fixed binary point operation appears when contrasted with floating point operation. Floating point operation requires the use of some bits within the half-word to indicate the position of the point, thus reducing the number of bits in the half-word available for representation of significant figures. Thus, given the same size for the half-word, fixed point operation allows greater precision than does floating point operation.

An advantage resulting from the restriction of numerical values within the range of decimal $+1$ to -1 becomes apparent upon consideration of the multiplication operation. Since multiplication of two fractional numbers yields a fractional result smaller than either of the original terms, no special programming precautions are necessary to prevent the result from exceeding the capacity of the computing elements as would be the case in working with integers.

No difficulty is encountered in converting into fractions the numbers on which the Central Computer System is to operate. Each physical measurement can be presented to the Central Computer System as the ratio of that measurement to the maximum possible measurement. For example, when the measurement of the distance between two points is presented to the Central Computer System, that measurement can be interpreted as the ratio of the distance measured to the maximum distance of which the Central Computer System is cognizant.

Another point to be considered in regard to the scaling of all numbers down to fractional values is this; all numbers are represented in the Central Computer System as integral multiples of $1/32,768$, or 2^{-15} . This fractional conversion factor can be eliminated from any calculation performed by the Central Computer System with no significant change in the results.

In effect, the Central Computer System converts all numbers supplied to it, whether constants or measurements of variables, into fractional form by multiplying those numbers by 2^{-15} , then reconverts the results into integral form by multiplying by 2^{15} .

3.3 BINARY ADDITION

The basic rules of binary addition are shown in table 1–3.

TABLE 1–3. RULES OF BINARY ADDITION

ADDEND	+	AUGEND	=	SUM
0		0		0
1		0		1
0		1		1
1		1		*(1)0

*Carry of 1 to higher order column.

As an example of binary addition, consider the following, using 5-bit binary numbers for convenience:

$$\begin{array}{r} 5/16 \quad 0.0101 \\ +9/16 \quad +0.1001 \\ \hline 14/16 \quad 0.1110 \end{array} \quad \begin{array}{r} 3/16 \quad 0.0011 \\ +1/16 \quad +0.0001 \\ \hline 4/16 \quad 0.0100 \end{array}$$

One difficulty, known as an overflow, may occur in addition if the sum of the two numbers added equals or exceeds unity:

$$\begin{array}{r} 0.1001 \quad 9/16 \\ +0.1001 \quad +9/16 \\ \hline 1.0010 \quad 18/16 \end{array}$$

In this case, although the binary number representing the sum is correct, the sign convention will cause its interpretation to be incorrect. If the magnitude bits are read directly and the sign convention is observed, the sum of 9/16 and 9/16 will be interpreted as $-2/16$. Overflow is discussed further in 3.4.5.

3.4 BINARY SUBTRACTION

3.4.1 Direct Subtraction

The rules of direct binary subtraction are shown in table 1–4.

TABLE 1–4. RULES OF DIRECT BINARY SUBTRACTION

MINUEND	–	SUBTRAHEND	=	DIFFERENCE
0		0		0
1		0		1
1		1		0
*(1)0		1		1

*Borrow of 1 from higher order column.

As an example of direct binary subtraction, consider the following:

$$\begin{array}{r} 0.1101 \quad 9/16 \\ -0.0101 \quad -5/16 \\ \hline 0.0100 \quad 4/16 \end{array} \quad \begin{array}{r} 0.1110 \quad 14/16 \\ -0.1001 \quad -9/16 \\ \hline 0.1001 \quad 5/16 \end{array}$$

Although direct subtraction is possible, certain disadvantages in its execution make alternative methods of subtraction attractive. For example, direct subtraction of a number from another number smaller than the first yields an incorrect result:

$$\begin{array}{r} 0.0101 \quad 5/16 \\ -0.1001 \quad -9/16 \\ \hline 1.1100 \quad \neq -4/16 \end{array}$$

The binary number recorded as the result, if the magnitude bits are read directly and the sign convention is observed, is $-12/16$.

The difficulty encountered with negative results and the problem of mechanically providing for borrows is eliminated by converting the subtraction process to an addition process using negative numbers expressed in what is known as the complement form. Subtraction by addition of complements offers the further advantage of using the same computing elements as are used for addition.

3.4.2 Complements

The use of complements in performing subtraction by addition of a negative number is based upon the fact that computing elements of fixed modulus (due to fixed capacity) are unable to recognize the difference between two numbers which differ in magnitude by an integral multiple of the modulus. In the equipment under consideration, the computing elements are restricted in size to 16-bit positions and are, therefore, modulo 2^{16} devices. (Refer to 1.2.4.) It follows that the equation $m - s = d$ (where m represents minuend, s represents subtrahend, and d represents difference) is equivalent in final results to the equation $m + 2^{16} - s = d + 2^{16}$ since the computing elements recognize $d + 2^{16}$ as d . The term $(2^{16} - s)$ is called the complement of s .

It would appear that, although the original subtraction has been converted to an addition process, a subtraction must still be performed to obtain the complement of a number. This subtraction to generate the complement of a number is rendered unnecessary, however, by an adventitious feature of the binary number system. If a binary number is rewritten, substituting a 1 in the rewritten number wherever the original number contains a 0 and substituting a 0 in the rewritten number wherever the original number contains a 1, the

rewritten number represents the difference between the original number and the modulus less 1. For example:

$$\begin{array}{rcl}
 \text{original number:} & 001010 & 111111 \\
 & & (\text{modulus less 1}) \\
 & & -001010 \\
 & & \underline{\hspace{1cm}} \\
 & & (\text{original number}) \\
 \\
 \text{rewritten number:} & 110101 & = \quad 110101 \\
 & & (\text{rewritten number})
 \end{array}$$

This process called complementing a number, of writing 1 for 0 and 0 for 1 in the number, is accomplished readily by computing devices. Further, if the complement of the original number with respect to the modulus rather than with respect to the modulus less 1 is desired, it can be obtained by adding a 1 to the least significant bit position of the rewritten number. Subtraction can therefore be performed binarily by a combination of complementing and addition.

3.4.3 Application of Complements

3.4.3.1 Interpretation

When a positive binary number is complemented, its sign bit changes from 0 to 1. Thus numbers in complement form, as representations of negative numbers, agree with the sign convention stated in 3.2.2. The magnitude bits of a number in complement form cannot be interpreted directly, however, since the results are misleading. In general, a number in complement form is recomplemented when the absolute magnitude of the number is desired.

3.4.3.2 Generation

Two forms of complements are used in the Central Computer System of AN/FSQ-7 Combat Direction Central. One, with respect to the modulus, is called the 2's complement. The other, with respect to the modulus less 1, is called the 1's complement. Given the binary number representing decimal 9/16, the two forms of complements are generated mathematically as follows:

2's complement		1's complement	
10.0000		1.1111	
- 0.1001	9/16	-0.1001	9/16
<u>1.0111</u>		<u>1.0110</u>	

Physically, the 1's complement of a number is generated by reversing the condition of each bit position. The 2's complement is generated physically by first generating the 1's complement, then adding a 1 into the least significant bit position. Thus:

1.0110	-9/16 (1's complement)
+ 1	
<u>1.0111</u>	-9/16 (2's complement)

Generating the 1's complement of a number requires one step. Generating the 2's complement of a number requires two steps.

To interpret the magnitude of a number in complement form (sign bit = 1), the number must be recomplemented following the same procedure as was used to generate the original complement. This recomplementing yields the absolute magnitude of the number in complement form which must be prefixed by a minus sign to show that the number so processed was negative (sign bit = 1). Again, recomplementing a number in 1's complement form requires one step while a number in 2's complement form requires two steps.

3.4.3.3 Subtraction Using 2's Complement

When the 2's complement is used in subtraction, the result is obtained in one step. For example, consider the subtraction of 9/16 from 14/16:

0.1110	14/16
+1.0111	- 9/16
<u>1.1100</u>	<u>5/16</u>
(1) 0.0101	

The actual result obtained mathematically is 10.0101, which is larger than the desired result by the modulus, the amount by which decimal 9/16 was increased when it was complemented. Since the computing element which holds the result cannot contain the sixth significant figure (shown in parentheses), that figure drops out, thus reducing the result to the correct value. In effect, neglecting the carry-out of 1 is equivalent to subtraction of 10,000 from the actual result. The carry-out of 1 does indicate that the result is positive but it is unnecessary mathematically, since the sign bit of the result indicates the sign of the result.

Use of complements in subtraction eliminates the problem encountered in direct subtraction when the subtrahend is larger than the minuend. For example, consider the subtraction of 9/16 from 5/16:

0.0101	5/16
+1.0111	-9/16
<u>1.1100</u>	<u>-4/16 (2's complement)</u>

Note that, when the result of the subtraction is negative, the sign bit of the result is a 1 and no carry-out is generated.

To show that 1.1100 is the 2's complement of 4/16, its absolute magnitude can be determined by recomplementing as follows:

1.1100	-4/16 (2's complement)
-0.0001	
<u>1.1011</u>	-4/16 (1's complement)
0.0100	4/16 (recomplemented)

3.4.3.4 Subtraction Using 1's Complement

The subtraction of 9/16 from 14/16 can also be performed, using the 1's complement:

$$\begin{array}{r} 0.1110 \\ + 1.0110 \\ \hline (1) 0.0100 \\ + 0.0001 \\ \hline 0.0101 \end{array} \quad \begin{array}{r} 14/16 \\ - 9/16 \\ \hline \\ \\ 5/16 \end{array}$$

The actual result of the first addition is 10.0100, which is larger than the desired result by the modulus less 1 or by 1.1111, the amount by which decimal 9/16 was increased when it was complemented. The fixed size of the computing element will eliminate the carry-out as was the case in using the 2's complement. However, neglecting the carry-out when using the 1's complement reduces the actual result by 10.0000, which is too large a reduction by the amount of 0.0001. The problem is resolved by adding the carry-out into the least significant digit position. This process yields the following result:

$$\begin{array}{r} 0.1110 \\ + 1.0110 \\ \hline 10.0100 \\ - 10.0000 \\ + 0.0001 \\ \hline 0.0101 \end{array} \quad \begin{array}{l} 14/16 \\ - 9/16 \\ \hline \\ \text{actual result} \\ \text{remove carry-out} \\ \text{end-carry} \\ \text{correct result (5/16)} \end{array}$$

The process of removing the carry-out and adding it into the less significant bit position is called the end-around-carry or, more simply, the end-carry.

It can be seen that the use of the 1's complement in subtraction requires two steps, the addition and the end-carry, in comparison to the one step required when the 2's complement is used. The two-step process is required with the 1's complement only when the result is positive (sign bit = 0, with carry out of 1). For example, consider the subtraction of 9/16 from 5/16 using 1's complements:

$$\begin{array}{r} 0.0101 \\ + 1.0110 \\ \hline 1.1011 \end{array} \quad \begin{array}{r} 5/16 \\ - 9/16 \\ \hline \\ - 4/16 \text{ (1's complement)} \end{array}$$

No carry-out is produced and, therefore, no end-carry is necessary. To show that 1.1011 is the 1's complement of 4/16, it can be recomplemented by inspection to yield 0.0100, which equals decimal 4/16 in magnitude.

3.4.3.5 Uses of 2's and 1's Complements

The choice of complement form to be used in calculation is dependent upon the relative importance of speed in generation to speed in calculation. It has been shown that the 1's complement is generated or recomplemented

in one step but may require two steps when used in the subtraction process. On the other hand, the 2's complement requires two steps for generation or recomplementation while its use in subtraction requires only one step. The convenience of conversion into and out of the 1's complement form has dictated the use of that form in almost all applications within the Central Computer System. All negative numbers are handled and stored in the Central Computer System in 1's complement form. The 2's complement form is used in binary division (discussed in 3.6) and in certain control functions (discussed in Chapter 3).

3.4.4 Zero

The use of the 1's complement for the representation of negative numbers introduces the anomaly of a negative zero. The binary number 0.0000 represents decimal 0; its 1's complement, 1.1111, also represents decimal 0 but in negative form. The two numbers, positive 0 (0.0000) and negative 0 (1.1111), are arbitrarily defined as identical. Both numbers are recognized by the Central Computer System as 0. The result of adding a binary number (other than zero) to its complement is negative zero.

3.4.5 Overflow

If the result of a binary addition or subtraction equals or exceeds unity, an overflow is said to have occurred; it cannot be interpreted meaningfully by the Central Computer System. The overflow condition was mentioned briefly in 3.3. The example given there is the addition of 9/16 and 9/16:

$$\begin{array}{r} 0.1001 \\ + 0.1001 \\ \hline 1.0010 \end{array} \quad \begin{array}{r} 9/16 \\ 9/16 \\ \hline \\ \neq 18/16 \\ \approx -13/16 \text{ (1's complement)} \end{array}$$

A similar result is obtained if 9/16 is subtracted from -9/16 or if -9/16 is added to -9/16:

$$\begin{array}{r} 1.0110 \\ + 1.0110 \\ \hline (1) 0.1100 \\ (1) \end{array} \quad \begin{array}{r} -9/16 \\ -9/16 \\ \hline \\ \\ \neq -18/16 \\ \approx 13/16 \end{array}$$

In both cases, the results exceed unity in the positive or negative direction and can not be interpreted meaningfully by the Central Computer System. The indication of an overflow may be seen from these two examples: the two numbers being added (whether in the addition or the subtraction process) are of the same sign whereas the sign of the result differs from the sign of the two numbers. This indication is used by the Central Computer System in detecting the occurrence of an overflow.

When addition of numbers in complement form is executed, it is possible that an overflow condition will be indicated at the completion of the first step and that the end-carry will remove the indication of overflow. This condition, known as a false overthrow, can be demonstrated in the subtraction of 8/16 from -7/16:

$$\begin{array}{r}
 1.1000 \quad -7/16 \\
 +1.0111 \quad -8/16 \\
 \hline
 (1) 0.1111 \quad (\text{false overflow}) \\
 \quad (1) \\
 \hline
 1.0000 \quad -15/16
 \end{array}$$

The possibility of treating a false overflow as an error must be avoided.

3.5 BINARY MULTIPLICATION

3.5.1 Direct Multiplication

The rules of binary multiplication are given in table 1-5.

TABLE 1-5. RULES OF BINARY MULTIPLICATION

MULTIPLIER	x	MULTIPLICAND	=	PRODUCT
0		0		0
0		1		0
1		0		0
1		1		1

Binary multiplication can be accomplished in the same manner as that used in decimal multiplication; i.e., the product of each multiplier digit times the multiplicand is written with the least significant bit of each partial product aligned under the corresponding bit of the multiplier, then added to yield the product. For example, the multiplication of 9/16 by 5/16 is accomplished as follows:

$$\begin{array}{r}
 0.1001 \quad \text{or} \quad 9/16 \quad (\text{multiplicand}) \\
 \times 0.0101 \quad \times 5/16 \quad (\text{multiplier}) \\
 \hline
 01001 \\
 00000 \\
 01001 \quad (\text{partial products}) \\
 00000 \\
 00000 \\
 \hline
 0.00101101 \quad 45/256 \quad (\text{product})
 \end{array}$$

Since the allowable digits in the binary number system are 1 and 0, each partial product will be either the multiplicand or a number containing all zeros. The change in order of each partial product can be accomplished by shifting the multiplicand left before recording it as a partial product or by shifting all earlier partial products

right before recording a new partial product. Further, it is not necessary to record each partial product. Instead, as each partial product is generated, it can be added to the sum of the previously obtained partial products. Multiplication can thus be reduced to a series of additions and shifts. Further, since the sign of the product can be determined by the rule of signs, the multiplication can be performed with the absolute magnitudes of the operands, thus eliminating the problem of recognizing the magnitude indicated by a number in complement form.

3.5.2 Multiplication by Addition and Shifting

A binary multiplication operation can be executed as a series of additions and shifts in accordance with the following rules:

1. Determine sign of product by rule of signs, then obtain absolute magnitude representations of multiplier and multiplicand by recomplementing if negative.
2. Form partial product:
 - a. Examine least significant bit of multiplier not already examined; if 1, add multiplicand to partial product; if 0, take no action.
 - b. Shift new partial product right one position; fill vacated positions with 0's.
3. Repeat step 2 for each significant bit position of multiplier (not for sign bit of multiplier).
4. After last repetition of step 2, correct sign of product according to sign determined in step 1.

Using these rules, the multiplication of 9/16 x 5/16 is accomplished as shown in table 1-6.

TABLE 1-6. BINARY MULTIPLICATION

STEP	MULTIPLIER	PARTIAL PRODUCT	MULTIPLICAND
A	0.0101	0.0000	0.1001
B 1	0.0101	0.1001	
B 2		0.1001	
B 1	0.0101	0.01001	
B 2		0.001001	
B 1	0.0101	0.1001	
		0.101101	
B 2		0.0101101	
B 1	0.0101		
B 2		0.00101101 (product)	

The method of multiplication as executed in the Central Computer System, is shown in table 1-6 and discussed in Chapter 3.

3.5.3 Shifting

Shifting the bits of a binary number with respect to the binary point is equivalent to multiplication of the binary number by decimal 2 or some power of decimal 2. It is thus analogous to multiplication of a decimal number by 10 or a power of 10 by the expedient of moving the decimal point. In the binary system number, it is convenient to think of the binary point as fixed, with the bits of the binary number shifting. For example, if the binary number 0.0101 is shifted left to appear as 0.1010, the original binary number has been multiplied by decimal 2. As another example, consider the binary number 0.0011 shifted left two places to appear as 0.1100. The shift of two places is equivalent to multiplication by decimal 4.

3.5.4 Round-Off

The product of two fractional binary numbers is a binary number, smaller in magnitude than either of the original numbers, but expressed with more significant figures. This precision of expression of the product is generally excessive. The excess precision and excess length in terms of bit positions of the product is corrected by a process known as round-off.

When a positive binary number is rounded off, a 1 is added to the least significant bit being retained if the most significant bit being dropped is a 1. Using the product of 9/16 and 5/16 as an example, if this number is to be rounded off to four significant bits, the following process results:

0.00101101

0.0010 1101

0.0011

For a negative binary number, a 1 is added to the least significant bit retained if the most significant bit being dropped is a 0. The round-off process is generally preceded by a shift left. The combined shift left and round-off allows the scaling up to representable size, with properly corrected precision, of a product which is smaller than the smallest binary number representable within the Central Computer System. If the product to be rounded off were 0.00000101, the result would be zero unless a shift left preceded the round-off.

3.6 BINARY DIVISION

3.6.1 Principles

Binary division is greatly simplified, in comparison to decimal division, by the fact that only 0 and 1 may appear as digits in the quotient. A 1 is written in the quotient if the divisor can be subtracted from the dividend leaving a positive remainder, while a 0 is written in the quotient if the remainder is negative. Binary division is, therefore, reduced to a series of

trial subtractions. One limitation is placed on binary division; the divisor must exceed the dividend in magnitude to yield a quotient of value less than unity. Violation of this limitation leads to results which will be interpreted incorrectly by the Central Computer System. In effect, the result of a division which does not observe the limitation is similar to an overflow condition.

3.6.2 Direct Division

The binary division process makes use of the 2's complement described in 3.4.3. The 2's complement is used in this application to reduce the time required to execute each trial subtraction. Use of the 1's complement would require added time for end-carries. Further, since the sign of the quotient can be determined by rule of signs from the signs of the divisor and dividend, no sign bit for the quotient is calculated. Instead, division is performed using the absolute magnitudes of the divisor and dividend, then the proper sign is affixed to the quotient and remainder, if any.

Direct binary division can be described best through the use of an example. Consider the division of 45/256 by 5/16:

(divisor)		.10010	
(2's complement)	0.0101	/ 0.001011010	(dividend)
	1.1011	11011	
		<u>1 000001010</u>	(current remainder)
		11011	
		<u>0 11100010</u>	
		00101	(restoration)
	(1)	<u>00001010</u>	(current remainder)
		11011	
		<u>0 1110110</u>	
		00101	(restoration)
	(1)	<u>0001010</u>	(current remainder)
		11011	
		<u>1 000000</u>	(final remainder)

Since both divisor and dividend are positive, the sign of the quotient is positive and the absolute magnitudes of the operands are as shown. The first step in the division process is the trial subtraction of the divisor from the dividend shifted left one position (the dividend is smaller than the divisor in magnitude by convention and a trial subtraction without shifting would be unsuccessful). The trial subtraction is successful (indicated by the 1 carry-out and by the 0 as the most significant bit of the remainder). A 1 is therefore written into the quotient. For the next trial subtraction, either the divisor must be shifted right or the current remainder shifted left one position. Consider the current remainder shifted left. The next trial subtraction is

unsuccessful (0 carry-out and 1 as most significant bit of remainder). Therefore a 0 is written in the quotient.

To continue the trial subtraction process, the negative remainder is restored (made positive) by adding the divisor. The carry-out produced by this addition is ignored. The restored current remainder is shifted left one position and another subtraction attempted. A negative remainder results. Another 0 is written in the quotient and the negative remainder again restored. The restored current remainder is shifted left one position and another subtraction attempted. The carry-out of 1 indicates that the subtraction is successful. Therefore, a 1 is written in the quotient. The final remainder is 0.

The direct or restoring method of division just described is effective but is not too rapid. In the example given, six additions are performed, four for trial subtractions and two for restoring current remainders. If the restoration steps are eliminated, only four addition steps are required for the division of a number by another number with four significant bits.

3.6.3 Non-Restoring Method

The effect of the restoration step can be analyzed as follows: given a current remainder x , this current remainder is doubled and the divisor d subtracted from it to perform a trial subtraction, giving $2x - d$ as the new current remainder. If this remainder is negative, a restoration step is carried out before the next trial subtraction. The divisor is added to the negative remainder, the result is doubled, and the divisor is subtracted from it:

$$2x - d + d = 2x$$

$$2(2x) - d = 4x - d$$

The restoring step can be eliminated as follows. If the remainder after a trial subtraction, $2x - d$, is negative, this remainder can be doubled and the divisor d added to it to accomplish another trial subtraction without an intervening restoration step:

$$2(2x - d) + d = 4x - d$$

The non-restoring method thus accomplishes the same result as does the restoring method but saves one step for each negative remainder.

Binary division, using the non-restoring method, can be executed as a series of subtractions (addition of complements) and shifts in accordance with the following rules:

1. Determine sign of quotient by rule of signs, then make divisor and dividend positive to allow calculation with absolute magnitudes.
2. Perform a trial subtraction:
 - a. Make sign of divisor unlike sign of current remainder (dividend for first trial subtraction)

by complementing divisor (2's complement) and shift remainder left one place.

- b. Add divisor (or 2's complement of divisor) to shifted current remainder; if new remainder is positive (carry-out of 1), write a 1 in quotient; if new remainder is negative (carry-out of 0), write a 0 in quotient.

3. Repeat step 2 once for each divisor bit, including sign bit.

4. If, after last repetition of step 2, divisor and remainder are both negative, complement divisor and add to remainder to restore it to positive form. Ignore carry-out. Correct sign of quotient and remainder in accordance with stored sign.

The rules of binary division using the non-restoring method applied to the division of 45/256 by 5/16 is shown in table 1-7.

TABLE 1-7. BINARY DIVISION

STEP	DIVISOR	CURRENT REMAINDER	QUOTIENT
A	0.0101	0.001011010 (dividend)	
B 1	1.1011	0.01011010	
2		1.1011	
		<u>1 0.00001010</u>	.1
B 1	1.1011	0.0001010	
2		1.1011	
		<u>1.1100010</u>	.10
B 1	0.0101	1.100010	
2		0.0101	
		<u>1.110110</u>	.100
B 1	0.0101	1.10110	
2		0.0101	
		<u>1 0.00000</u>	.1001
B 1	1.1011	0.0000	
2		1.1011	
		<u>1.1011</u>	.10010
D	0.0101	1.1011	0.10010
		0.0101	
		<u>1 0.0000</u>	

This method of division, as executed in the Central Computer System, is discussed in Chapter 3.

3.7 NUMBER CONVERSION USING BINARY ARITHMETIC

Conversions from binary to decimal and decimal to binary notations are possible using binary arithmetic in exactly the same manner as described in Section 2. One added feature is required to allow the handling of decimal numbers by a binary machine; binary coding of decimal numbers. A distinction must be made between a binary-coded decimal number and a decimal number expressed binarily. The decimal number 33 is expressed binarily as 100001; the same decimal number can be coded binarily as 0011 0011. The binary-coded decimal number is written by representing each decimal digit in binary form. The binary codes for the decimal digits 0 through 9 and for several other decimal numbers are given in table 1–8.

TABLE 1–8. BINARY-CODED DECIMAL NUMBERS

DECIMAL	BINARY CODE	DECIMAL	BINARY CODE
0	0000	20	0010 0000
1	0001	21	0010 0001
2	0010	22	0010 0010
3	0011	23	0010 0011
4	0100	24	0010 0100
5	0101	25	0010 0101
6	0110	30	0011 0000
7	0111	45	0100 0101
8	1000	60	0110 0000
9	1001	75	0111 0101
10	0001 0000	99	1001 1001

SECTION 4

OCTONARY NUMBER SYSTEM

The octonary number system is useful in connection with digital computers as a form of shorthand for binary notation. Its radix, expressed decimally, is 8, and the allowable digits in octonary notation are 0, 1, 2, 3, 4, 5, 6, and 7. It can be seen from table 1-9 that each octonary digit can be represented binarily by three bits and, further, that every combination of three binary bits has a corresponding octonary digit.

TABLE 1-9. BINARY-CODED OCTONARY NUMBERS

OCTONARY	BINARY	OCTONARY	BINARY
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Just as is the case for a decimal number, an octonary number can be expressed in binary-coded form. (Refer to 3.7.) Unlike a binary-coded decimal number, however, a binary-coded octonary number is identical to the binary notation of the quantity represented by the octonary number. For example, decimal 47 is expressed octonarily as 57_8 ($5 \times 8^1 + 7 \times 8^0$). The octonary number 57_8 can be binarily coded as 101 111. The binary number representing decimal 47 is 101111. Thus, an octonary number can be translated into its true binary equivalent by translating each octonary digit into the corresponding three binary bits. The converse is also true; a binary number can be translated into its true octonary equivalent by replacing each successive group of three binary bits (starting from the binary point) with its corresponding octonary digit.

The justification for this direct translation between binary and octonary notation can be demonstrated by the following reasoning. The modulus of a group of three binary bits is 2^3 or 8. By definition, the radix of the octonary system is 8. The maximum count representable by three binary bits is $2^3 - 1$ or 7, which corresponds exactly to the value of the maximum allowable octonary digit, 7. These statements indicate that there is a point-to-point correspondence between each combination of three binary bits and some octonary digit. More generally, each value of the 3-bit binary number

$b_2b_1b_0$ can be equated to a value of the octonary digit e . Expressed decimally:

$$b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0 = e \times 8^0 \quad (3)$$

where $b = 0$ or 1

$e = 0, 1, 2 \dots 6, 7$

The equation just given states the equality of the first three bits to the left of the binary point with the first octonary digit to the left of its radix point. The equation can be generalized to cover the equality for any position:

$$2^{3p} (b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0) = 2^{3p} (e \times 8^0) \quad (4)$$

where p = position of digit or bit group with respect to radix point. The generalized equation given as (4) describes the equivalence for corresponding positions of octonary digits and 3-bit binary groups. Since equals added to equals are equal, it follows that, if each octonary digit is equal in value (including magnitude represented by position) to each 3-bit binary group in a larger number, the octonary number is equal to the binary number.

The utility of the octonary number system to programmers falls into two categories:

- a. Octonary notation is briefer than binary notation.
- b. Conversion between octonary and binary notation can be performed by inspection.

A 16-bit binary number may be written in 6 octonary digits. For example:

$$0.010100111011110 = 0.24736_8$$

It is obvious that the octonary notation is briefer than the binary notation. Further, the octonary notation gives a rough idea of the magnitude of the number in binary notation. The only other means of obtaining a rough approximation of magnitude is the rather tedious process of conversion to decimal notation.

Conversion between octonary and binary notation is performed by inspecting the notation to be converted and substituting on the basis of one octonary digit per three binary bits. In the example just given, 0.010100111011110 is grouped:

$$0.010 \ 100 \ 111 \ 011 \ 110$$

The sign bit is written as a 0 and each remaining group

of three bits is replaced by the equivalent octonary digit as shown in table 1-9. The resulting octonary number is 0.24736_8 .

A binary number in complement form may also be represented octonarily. The octonary digit preceding the radix point represents the sign of the number represented just as is the case in binary notation; a 0 indicates a positive number while a 1 indicates that the number is negative and in complement form. For example, the complement of the binary number just given, 1.101 011 000 100 001, is represented octonarily as 1.53041_8 . It should be noted that the significant bits of the octonary notation of the complemented number appear as a 7's complement of the direct octonary digits.

The choice of the octonary system as a shorthand for binary numbers is determined by three factors. First, only a number system whose radix is a power of 2 offers the advantage of direct translation. A binary number can be translated directly into a number system

based on 4, for example. However, translation of a binary number into the notation based on 9, for example, involves the same difficulties as those encountered in translation into decimal notation without the advantage of familiarity offered by decimal notation. Second, the higher the power of 2 used as radix for the shorthand notation, the greater the economy offered. Thus, the number system based on 4 offers a saving of only one digit for each two bits rather than the one digit for three bits of octonary notation. Finally, any number system of radix greater than 10 introduces the problem of recognizable single symbols for values greater than decimal 9. For example, the number 11 in the sexa-decimal system (radix = 16_{10}) can be interpreted as 11_{10} or as 17_{10} unless some added symbolism is used to eliminate the ambiguity. Therefore, the octonary number system offers the greatest economy without ambiguity as a shorthand notation of binary numbers.

CHAPTER 3

CENTRAL COMPUTER SYSTEM

SECTION 1

INTRODUCTION

1.1 SYSTEM CHARACTERISTICS

Stated generally, the characteristics of the Central Computer System include:

- a. Computation with fractional binary numbers of fixed binary point
- b. Parallel operation rather than serial
- c. Dual arithmetic operation
- d. Automatic sequence stored program of single-address instructions
- e. Real-time operation
- f. General applicability

A knowledge of the significance of these characteristics is useful in introducing the operation of the Central Computer System. Accordingly, these characteristics are discussed in the remainder of this Section.

1.2 DESCRIPTION OF CHARACTERISTICS

1.2.1 Information Form

All data in the Central Computer System are handled as 16-bit binary numbers with a fixed binary point between the first (sign) bit and the remaining 15 magnitude bits. With negative numbers stored internally in 1's complement form, any information number n falls within the limits $+1 > n > -1$.

The physical size of this information unit allows sufficiently precise computation within the precision limitations imposed by the measurements generating air defense data. The Central Computer System, with an inherent precision of one part in 32,768, can handle this data without degrading its precision.

The use of a fixed binary point, although it introduces scaling problems if maximum precision is desired, avoids the need for order-indicating bits as used in scientific notation and for space within words for these bits. The placement of the binary point, separating the sign bit from the magnitude bits of the number, allows simplification of some of the logical operations of the Central Computer System.

1.2.2 Handling Methods

1.2.2.1 Parallel Operation

The Central Computer System uses parallel rather than serial operation to gain the advantage of higher operating speeds in spite of greater equipment requirements. The transfer of a 16-bit number in parallel requires the use of 16 lines in comparison to the one line needed for a serial transfer. However, the serial transfer takes 16 times as long as the parallel transfer, since all bits are transferred simultaneously in the parallel operation while each bit is transferred sequentially in the serial operation. Similarly, parallel addition of two 16-bit numbers requires 16 adder circuits to add each pair of corresponding order bits simultaneously; this is in contrast to serial addition, which needs only one adder circuit. However, the time-saving aspect of parallel operation makes up for the additional equipment requirement in the Central Computer System.

1.2.2.2 Dual Operation

The operation speeds attained with parallel operation are increased in the Central Computer System by dual operation in two arithmetic elements. In general, the information handled by the Central Computer System consists of point designations in polar or rectangular co-ordinate form. The dual arithmetic elements allow simultaneous manipulation of the two halves of the co-ordinate pair, thus halving the calculation time for each pair. The dual arithmetic elements each handle one 16-bit number, generally in the same manner at the same time.

1.2.2.3 Computer Word

To meet the requirements of dual arithmetic operation, the basic information unit of the Central Computer System is modified in that two 16-bit numbers are handled as a single unit, known as a word. The two numbers are contained in half-words, designated the left and right half-words corresponding with the designations of each half of the dual arithmetic element. (See fig. 1-11, B12.) All information is handled within the Central Computer System as a word. In general, the left half-word is processed by the left arithmetic element

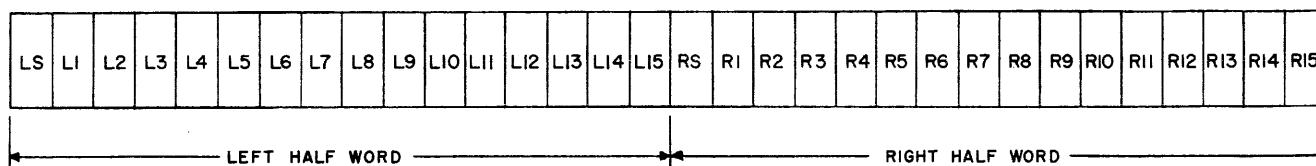


Figure 1–11. Computer Word

while the right half-word is processed by the right arithmetic element.

1.2.3 Internal Memory

The information on which the Central Computer System operates, and the instructions that direct these operations, are stored as words in the core memory element. Processed information is returned to core memory before transfer to the other systems of AN/FSQ-7 Combat Direction Central. Core memory can store only 8192 words. However, total data storage capacity is supplemented by the Drum System. (Refer to Chapter 4.)

The computational speed of the Central Computer System is largely determined by the operational speed of core memory. Successive words (whether they contain instructions or data) can be obtained from core memory at 6-microsecond intervals. Therefore, an instruction which does not require an operand from core memory is allowed 6 microseconds for its execution, while an instruction requiring an operand is allowed 12 microseconds for its execution. The controlling aspect of core memory operation requires discussion prior to a detailed discussion of overall Central Computer System operation. Accordingly, core memory operation is described in Section 2, and system operation in Section 3.

1.2.4 Instructions

1.2.4.1 Significance

As stored in core memory, an instruction word and an information word are indistinguishable; both appear as groups of binary bits. However, instruction words direct the operation of the Central Computer System upon information words. An instruction word differs drastically in significance from an information word, which can be read as two binary numbers having only quantitative significance. The significance of the binary bits within an instruction word can be understood best after examining the function of an instruction.

1.2.4.2 Instruction Functions

A given arithmetic operation, such as addition of two numbers, can be performed by an automatic digital computer only if the following control information is supplied by the program:

- Identification of the operation
- Addresses in memory at which the two operands may be found

- Address in memory for storage of the result
- Address from which the next program operation information may be obtained.

If one instruction is to supply all this information, it must contain four addresses in addition to the code signal identifying the operation. Such a four-address instruction is feasible only if internal memory is small. For the Central Computer System, a four-address instruction would be extremely unwieldy. Each binary address of a word in core memory must be at least 13 bits in length ($\text{modulo } 2^{13} = 8192$), requiring 52 bits for the four addresses alone.

A four-address instruction can be simplified in accordance with the following reasoning. The address specifying the next program operation storage location can be eliminated by storing program instructions sequentially in core memory in the order in which they are to be performed. If this is done, a counter can be used to obtain instruction words in the proper order. The address for storage of the operation result can be eliminated by dividing the arithmetic operation into a computation and a storage operation, with separate instructions for each. Using similar reasoning, the arithmetic operation may be divided into two operations, each one controlling the transfer of one operand from core memory. The end result is the use of several single-address instructions in place of the unwieldy four-address instruction.

1.2.4.3 Single-Address Instruction

In the single-address instructions used by the Central Computer System, the left half of each instruction word is reserved for identification of the operation to be performed and for other control functions. This half-word is called the operation half-word. The right half of an instruction word contains an address when necessary and is called the address half-word. Although 16 bits are available in the operation half-word, only bits L4 through L10 are used to identify the operation to be performed. These bits comprise the operation code. The functions of the remaining bits in the operation half-word are discussed in Section 3.

The address half-word generally specifies the address of an operand which is to be manipulated by the instruction. If no operand is required, the address half-word may be meaningless or may have quantitative

significance. The functions of the address half-word are also discussed in Section 3.

1.2.5 Programs

Although the instructions used with the Central Computer System are few in number, they may be assembled into a much larger number of programs. These may be divided into three major classes: operational, utility, and maintenance programs.

Operational programs are those programs directly concerned with the processing of air defense data. Utility programs are those which perform operations needed in many other programs. Instead of containing the instructions of the utility program within themselves, these other programs refer to the utility programs to perform the operation required. For example, if a number of other programs require translations from decimal to binary notation, a utility program for this translation might be prepared for use by the other programs. Maintenance programs used with AN/FSQ-7 Combat Direction Central fall into two types, reliability and diagnostic programs. The former check its operation to discover malfunctions; the latter are used to localize the cause of these malfunctions.

1.2.6 Real-Time Operation

The Central Computer System approximates real-time operation in two ways. The passage of real-time as recorded by a clock is taken into account in calculating the velocities of moving objects. In addition, these calculations are performed rapidly enough to be of use

in controlling a real-time situation. The latter consideration has dictated the adoption of various design features which increase speed of computation, such as parallel operation and dual arithmetic operation.

1.2.7 General Applicability

Although it is seldom stressed, the Central Computer System is capable of general application. The features included within it for rapid processing of air defense data are applicable to the handling of any type of data; even the fixed word length used in the Central Computer System offers no serious difficulties. It is possible to program double-precision calculations to utilize the full word length of the Central Computer System. An extension of the same principles allows multiple-precision calculation at the expense of speed. These techniques are not often used in air defense data processing, but their availability makes the Central Computer System a highly versatile digital computer.

1.3 DUPLEXING

Since failure of the Central Computer System would render inoperative all the equipment contained in AN/FSQ-7 Combat Direction Central, two Central Computer Systems are included in the equipment; one in the duplex group is called Computer A and the other in the group is termed Computer B. (Refer to 3.3, Ch 1, Sec 3.) The duplication of the entire Central Computer System in each duplex group eliminates any need for special switching circuits other than those required for switching between the entire Computer A and the entire Computer B.

SECTION 2

INTERNAL MEMORY OPERATION

2.1 CORE MEMORY

2.1.1 Introduction

Internal memory of the Central Computer System consists principally of two magnetic core memory units. Each unit has a storage capacity for 4096 words, giving a total capacity of 8192 words. Random access time for a single word in core memory is 6 microseconds. That is, 6 microseconds must elapse between successive word transfers.

The storage component of each core memory unit is a three-dimensional array of 135,168 ferrite cores, arranged to provide 4096 storage locations for 33-bit words. Specific registers (word storage locations) within the storage array are selected for reading and writing, using a coincident-current method by circuits within each unit. (See fig. 1-12.) Core memory is compact, requires no periodic regeneration, is nonvolatile (stored information is retained through a normal power off-on sequence), and is suited to parallel operation.

2.1.2 Principles of Operation

Ferrites, the materials of which the cores in core memory are made, are magnetizable ceramic-like materials whose magnetic state can be switched at high speeds. A ring-shaped ferrite core can be magnetized in one of two directions and can, therefore, store a binary bit if one direction is defined as binary 1 and the other direction defined as binary 0.

Figure 1-13 shows the hysteresis loop of a ferrite core with variation in magnetic state plotted along the vertical axis against the variation in magnetizing current plotted along the horizontal axis. A core is said to contain a 0 when its magnetic state is at point A or a 1 when its magnetic state is at point E. To change the state of a core from 0 to 1, a current of amplitude $-I$ is applied to a magnetizing winding on the core, causing the magnetic state to change from point A along the path ABCDE to point E. If a current of $-I$ is applied to a core containing a 1, the core state shifts along the line EDC until the current ends, allowing the core state to return to point E. Thus, after an application of a current of $-I$, known as a write current, to a core, that core is left in the 1 state.

A core is switched from the 1 to the 0 state by applying a current of amplitude $+I$ to the magnetizing winding. When the current is applied, the magnetic state of the core shifts along the line EFGHA to point

A. If a current of $+I$ is applied to a core in the 0 state, then the magnetic state shifts from A along AHG until the current is removed, allowing the magnetic state to return to point A. Thus, after an application of a current of $+I$, known as a read current, to a core, the core is left in the 0 state.

The rectangularity of the core hysteresis loop indicates that the core is relatively insensitive to half-read or half-write currents. A half-read or half-write current, once applied and removed from a core, produces no appreciable net change in the magnetic state of the core. A means exists, therefore, for selecting a given core within an array of cores for reading or writing.

2.1.3 Coincident Current Selection

2.1.3.1 Core Array

The core array of each core memory unit contains 33 planes, with each plane containing 4096 cores in a 64 by 64 matrix. (See fig. 1-14.) Each core in a given plane is supported by two wires at right angles to each other, called the X and the Y windings, respectively. There are 64 X and 64 Y windings to support all the cores in the plane. The location of each core can be specified by identifying the X and the Y winding intersecting at that core, just as a point in a plane may be identified by specifying the X and Y co-ordinate of the point. The 33 planes in a core array are stacked vertically, with each plane lying horizontal. A word is stored with one bit in each plane of the core array. The cores containing a given word (a core register) lie along a vertical Z line running perpendicular to the 33 X, Y planes. (See fig. 1-15.) A given register can be identified by specifying one X and Y winding, since the corresponding windings in each of the 33 planes are wired together.

2.1.3.2 Word Storage

If half-write currents are applied to both an X and a Y winding, those cores through which only one of these windings pass will feel only a half-write current and will be unaffected. However, the coincidence of the two half-write currents at the intersections of the pulsed X and Y windings will apply a full-write current to the cores at those intersections. Thus, the cores in the register specified by the X and Y winding will all be switched to the 1 state; all other cores in each plane will remain in their original states.

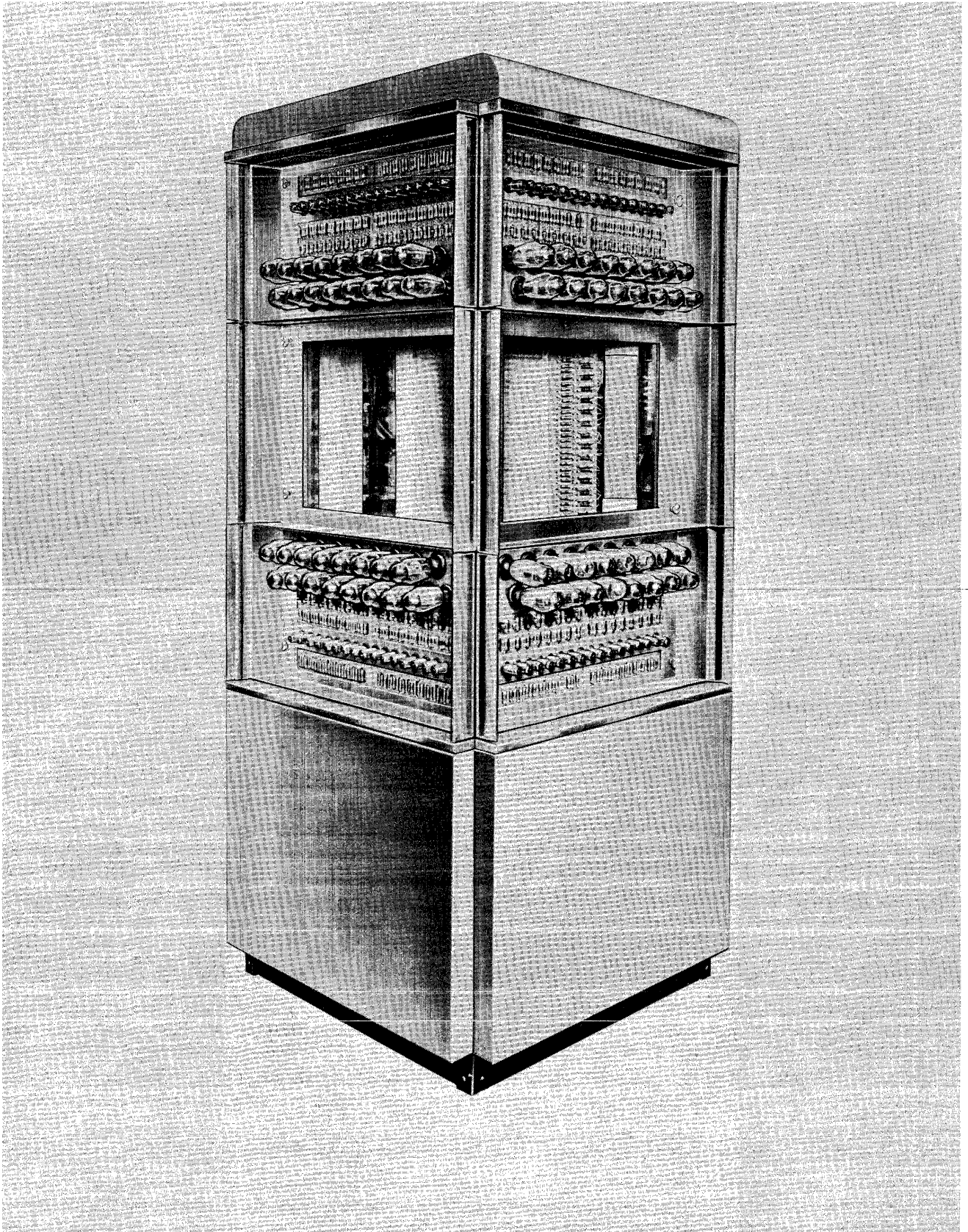


Figure 1-12. Core Memory Unit

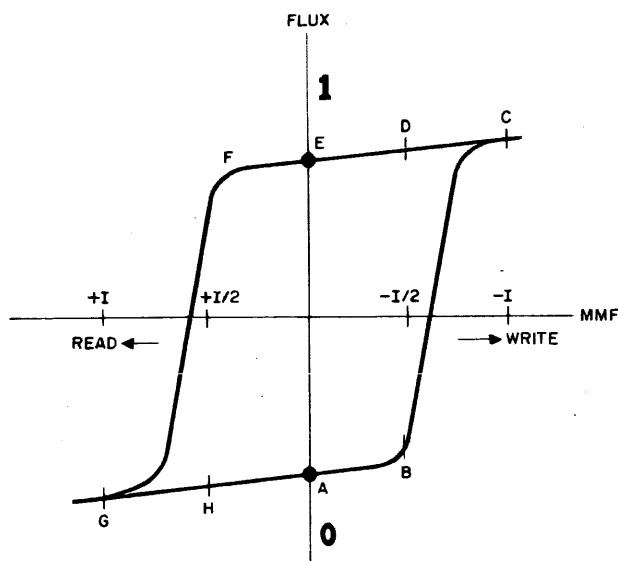


Figure 1-13. Hysteresis Loop of Ferrite Core

When a word is stored in a register, it is necessary to duplicate a given pattern of 1's and 0's in the cores of that register corresponding to the pattern of 1's and 0's in the word. Applying half-write currents to an X and Y winding will switch a register to all 1's. Some means is needed to prevent the writing of a 1 in a core which should contain a 0. This is accomplished by providing an extra winding in each plane, called the inhibit winding, which passes through every core in the plane. In order to duplicate a word in a core register, the register is first cleared (the cores are read to the 0 state). A half-write current is applied to the X and Y winding specifying the desired core register. Simultaneously, those bits of the word to be stored which contain a 0 cause the application of a half-read current on the inhibit winding of the planes in which those 0 bits are to be stored. A core in the register which is to remain at 0 thus feels two half-write currents and a half-read current. The net effect, a half-write current, is insufficient to switch the core to the 1 state. All cores in the register that do not feel a half-read current on the inhibit windings of their planes are switched to the 1 state. Thus, in one parallel operation, the pattern of 1's and 0's representing a word is duplicated in the cores of the register.

2.1.3.3 Word Reading

The reading of a word from a core register is based upon this principle: switching of a core in the 1 state to the 0 state generates a large flux change, while no appreciable flux change is generated when an attempt is made to switch a core in the 0 state to 0. The flux change produced in switching a core from the 1 to the 0 state can generate a relatively large voltage pulse in a wire threaded through the core being switched. Accordingly,

there is a winding in each plane, called the sense winding, which is used solely for reading. The sense winding, like the inhibit winding, passes through each core in the plane. Unlike the inhibit winding, no current is applied to sense winding. Since each bit of a word stored in a core register is stored in a separate plane, the word can be read out in parallel on the 33 sense windings of the core array.

In reading a word out of a core register, the pattern of cores in the 1 and 0 states is transformed into a pattern of pulses (1's) and lack of pulses (0's) on the sense windings for transmission to a flip-flop register where the pattern of 1's and 0's is represented by the states of the flip-flops. The reading is accomplished by applying half-read currents to the X and Y windings designating the desired core register. All cores in the selected register thereby feel a full-read current, thus switching them to the 0 state. A core which originally contained a 0 produces very little flux change and consequently no appreciable pulse on the sense winding of that plane. A core which originally contained a 1 produces a large flux change during the switching and induces an output pulse on the sense winding of its plane. Reading of a word from a core register destroys the contents of that core register and leaves the cores cleared. The pulses produced by the reading are supplied to the memory buffer register, a flip-flop register with a 33-bit capacity, setting the contents of that register to duplicate the original contents of the core register just read.

Reading a core register destroys the contents of that register. To retain the word in core memory, it must be restored in the core register. The word is held in the memory buffer register for this purpose. Those bits in the memory buffer register which are 0's cause the application of the half-read currents on the necessary inhibit windings. The use of the memory buffer register in controlling restoration of the read-out word to the core register in no way impedes or delays any transfer of that word from the memory buffer register to another register. The contents of a flip-flop register can be transferred to another register without disturbing the contents of the first register. Thus, the memory buffer register allows the transfer of a word read out of core memory to another element of the Central Computer System while that word is still being rewritten in core memory.

2.1.4 Register Addressing

A register in core memory is selected by specifying the X and Y windings that intersect at the cores of that register. The number of combinations of X and Y windings is the same as the number of cores in a plane or the number of registers in the array (4096). Each combination of X and Y windings that selects a register is

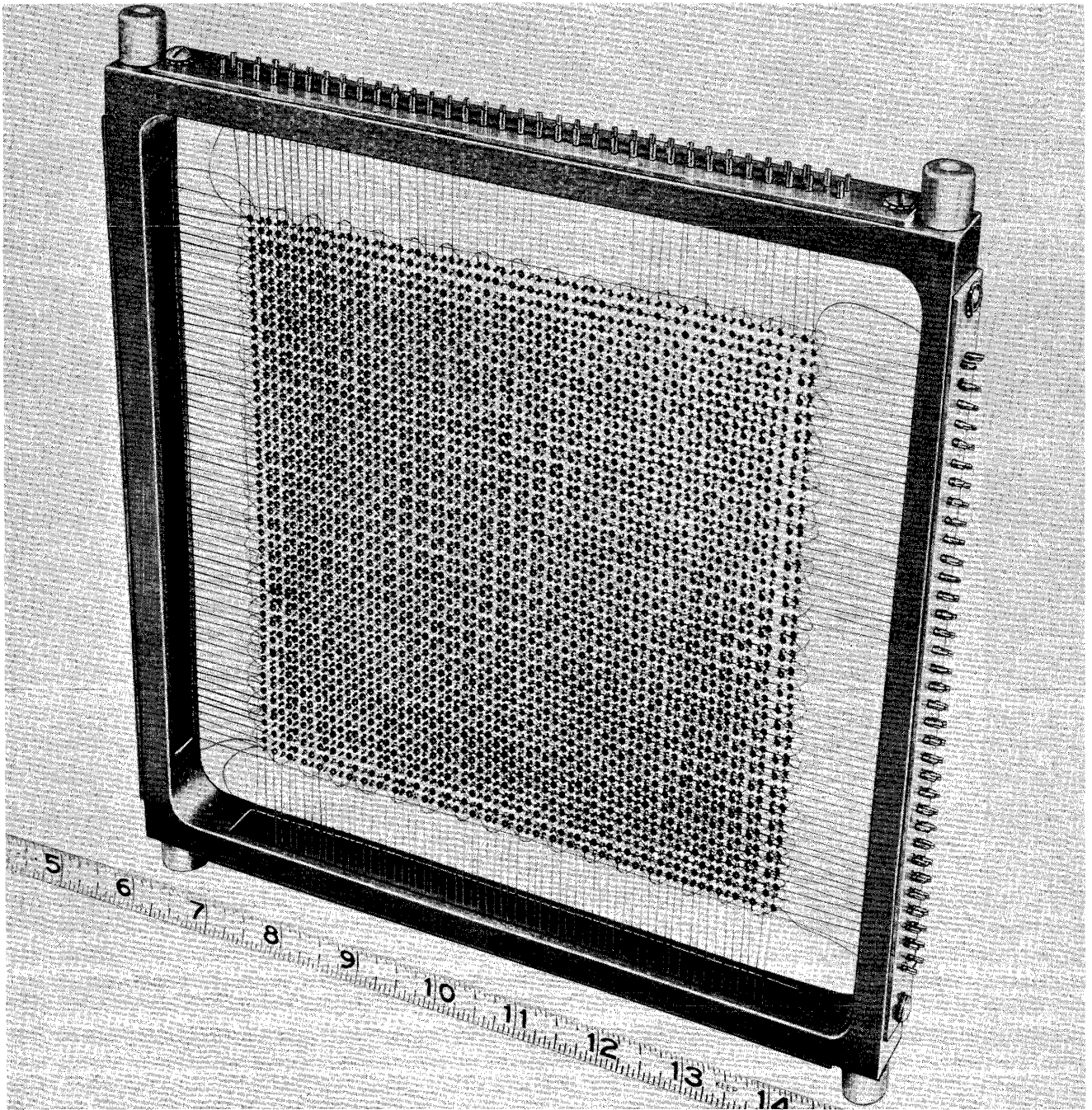


Figure 1-14. Core Memory Plane (55-611)

assigned an address which uniquely identifies that pair of windings and, hence, that core register. These addresses range in numerical value from decimal 0 through 4095. Twelve binary bits are required to represent this range of values in binary form.

The selection of a core register in a core memory unit is accomplished by a 12-bit flip-flop register in the unit. This register, called the memory address register, drives two decoding matrices which select one X and one Y winding in accordance with the contents of the

memory address register. Only one core register may be selected by the memory address register at any one time. Thus, although the sense windings and the inhibit windings of each plane are common to all cores in that plane, they sense or control only the core in that plane selected by the memory address register.

2.1.5 Function of Memory Buffer Register

The memory buffer register, a 33-bit flip-flop register located in the arithmetic element, performs the fol-

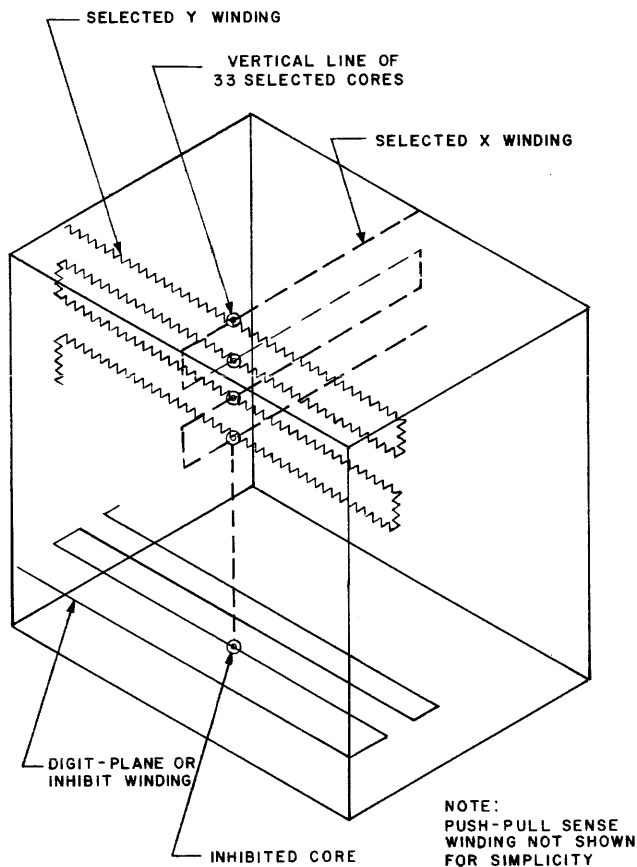


Figure 1-15. Core Array, Simplified

lowing functions in conjunction with the core memory element:

- Provides temporary storage and control for a word being written into core memory
- Receives each word read from memory and allows its transfer to other elements while restoring is done
- Assigns parity bit to words being stored and checks parity of words read out

The function of the memory buffer register in connection with reading and writing is similar to the function of any buffer register. A buffer register provides temporary word storage during transfer operations to free the source or the recipient in the transfer operation as rapidly as possible. For example, in the writing of a word into core memory, the transfer of the word to the memory buffer register from the originating register frees the originating register for other use.

The memory buffer register includes circuits which provide checks on the accuracy of transfers into and out of core memory. The accuracy checks are based on a redundancy principle; a 33rd bit is added to each word written into core memory. This 33rd bit, called a parity

bit, performs no function except in the accuracy check. The content of the parity bit makes the total number of 1 bits in the 33-bit word odd. As each word is read out of core memory, the number of 1's in the word is counted. If this number is even, an error has occurred. If the number of 1's is odd, there may have been no error or there may have been two errors. The likelihood of two errors occurring in one transfer is small enough to be ignored.

An error detected through the use of the parity bit is called a parity error and results in an alarm indication which may be used to stop the operations of the Central Computer System or to branch program control to take account of the parity error.

2.1.6 Memory Cycles

2.1.6.1 Timing

The operation of core memory is organized into cycles of 6-microsecond duration. Two types of cycles are possible, read or write. During a read cycle, a word is read from core memory and placed in the memory buffer register where it is available for transfer to other registers in the Central Computer System. Within the same read cycle, the word is rewritten in core memory. During a write cycle, a core register is cleared and a word previously placed in the memory buffer register is written into the cleared core register. Either cycle requires 6 microseconds for execution. Figure 1-16 shows the timing of the events which occur in both types of memory cycles. Once a memory cycle is started and the type of cycle is specified, the core memory unit performs a prescribed series of steps through the end of the cycle.

2.1.6.2 Read Cycle

Only one word may be read from core memory in a single read cycle. The read cycle is started by the command from the instruction control element that transfers a core register address into the memory address register. The read-out portion of the read cycle is performed as described in 2.1.3.3 and the word in the selected core register is duplicated in the memory buffer register in 3 microseconds. That word can thus be transferred to another element 3 microseconds after the start of the memory cycle.

The remaining 3 microseconds of the read cycle are consumed in rewriting the read-out word in the core register and in clearing the control circuits of the core memory unit. The flux changes produced during the restoration step have no effect on the contents of the memory buffer register, since the sense windings are disconnected from that register during the restoration step.

2.1.6.3 Write Cycle

Only one word may be placed in core memory during a single write cycle. The write cycle is started

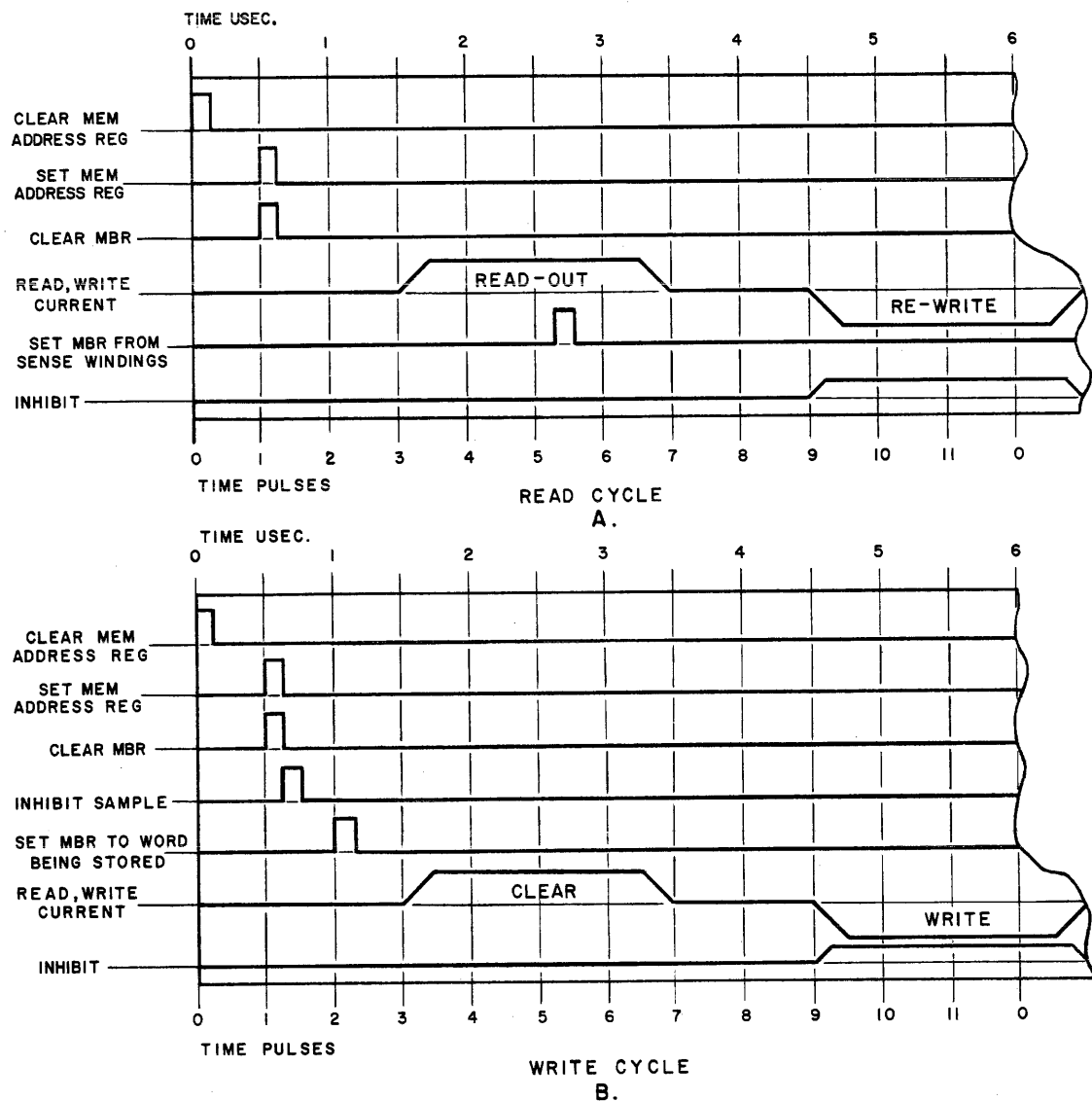


Figure 1-16. Memory Cycles

by the command from the instruction control element that transfers the core register address into the memory address register. The read-out portion of the write cycle is performed just as in a read cycle with one exception; the sense windings are disconnected from the memory buffer register for the entire duration of the memory cycle by a command, *inhibit sample*, from the instruction control element. Therefore, the original contents of the selected core register, although impressed on the sense windings as in a read cycle, are not duplicated in the memory buffer register but are lost instead. The read-out portion of a write cycle only clears the selected core register.

Since the sense windings are disconnected from the memory buffer register throughout the write cycle, the word to be written into core memory can be placed in the memory buffer register at any time within the first

3 microseconds of the memory cycle. During the second 3-microsecond interval, the word in the memory buffer register is written into the selected core register in the same manner as that used to rewrite a word just read from core memory.

2.2 TEST MEMORY

Test memory provides storage for a relatively small number of words for various special purposes. The primary use made of test memory is in testing core memory and in storing short test programs or parameters for core memory programs. The capacity of test memory is 16 words and its access time is identical to that core memory. Test memory is discussed in Chapter 10.

2.3 REAL-TIME CLOCK

The real-time clock provides accurate incremental real-time information for all systems within AN/FSQ-7

Combat Direction Central. The real-time clock consists of a fixed-frequency pulse generator and a 16-bit flip-flop counting register called the clock register. Clock pulses are generated at a 32-pulse-per-second rate. The real-time clock thus allows the determination of time within an interval of 2048 seconds (34.13 minutes) to a precision of 1/32 second.

The real-time clock supplies its basic timing frequency of 32 pulses per second to the Output System and also supplies selected timing pulses at 1/4- and 8-second intervals to the Input and Drum Systems. These supplied timing pulses and the ability of the Central Computer System to read the clock register allows the use of real-time increments in calculation. This use is discussed further in Chapters 5 and 7.

2.4 MEMORY ADDRESSES

Both the unit to be used and the address within that unit must be given in order to read or write a

specific location in internal core memory. The address half of an instruction word can contain this information. Of the 16 bits available within the half-word for addressing, the sign bit is not generally used; this leaves 15 bits for effective addressing. Twelve bits are reserved for selection of a specific register in one of the core memory units, leaving 3 bits for the selection of a memory unit. Table 1-10 shows the form of the address half-word for selection of each memory unit register.

The register address is supplied either to the memory address register in one of the core memory units or to the test memory address register, in accordance with the selection of memory unit by the unit code. No register address is applicable for the clock register, since there is only one clock register for reading. In addition, the memory unit addresses of core memory No. 1 and No. 2 may be interchanged for test purposes by means of a switch on the duplex maintenance console.

TABLE 1-10. MEMORY ADDRESSES

MEMORY UNIT	OCTONARY UNIT CODE	SELECTION CODE			REGISTER ADDRESS
		R1	R2	R3	
Core memory No. 1	0, 4	—	0	0	R4 through R15
Core memory No. 2	1, 5	—	0	1	R4 through R15
Test memory	2, 3	0	1	—	R12 through R15
Clock register	6, 7	1	1	—	Meaningless

SECTION 3

SYSTEM OPERATION

3.1 INTRODUCTION

3.1.1 General

In terms of programming, the operation of the Central Computer System begins with instructions and ends with the results accomplished by the execution of those instructions. However, instruction execution cannot be discussed meaningfully without some prior discussion of the equipment performing the required actions and the modes of operation of that equipment. One part of the equipment, internal memory, has been discussed in Section 2. This section of text therefore describes the remaining operating units and the timing of their operations before describing the execution of specific instructions.

3.1.2 Types of Operations

Operations of the Central Computer System fall into two distinct categories: internal and IO operations. Internal operations process information obtained from core memory and return the results to core memory. IO operations transfer information between core memory and units external to the Central Computer System. The distinction between these two categories of operations arises from the difference in their functions and from differences in the equipment performing the operations. Further, internal and IO operations are performed within the same time interval (although not simultaneously) by separate portions of the operating equipment within the Central Computer System.

3.1.3 Timing

3.1.3.1 Machine Cycles

Since almost all Central Computer System operations are dependent upon the operation of core memory, these operations are organized into machine cycles corresponding in occurrence and duration with memory cycles. (Refer to 2.1.6, Sec 2.) Internal operations are executed within two types of machine cycles: program time (PT) and operate time (OT). A program time cycle is defined as the machine cycle executed during the memory cycle in which an instruction is obtained from core memory. An operate time cycle is defined as the machine cycle executed during the memory cycle in which an operand is obtained from or returned to core memory. There are two types of operate time cycles, OT A and OT B. During an OT A cycle, an operand is obtained from core memory; during an OT B cycle, a

word is stored in core memory. Thus, there are three types of machine cycles during which internal operations are performed; PT, OT A, and OT B. For those operations of variable execution time which do not, however, require the execution of a memory cycle, an acyclic mode of Central Computer System operation is available. This acyclic mode, called an arithmetic pause, is required in the execution of instructions involving repetition of certain basic steps without reference to core memory. The use of the arithmetic pause is discussed in 3.4 and 3.6.

Just as internal operations are organized into machine cycles, so are IO operations. The execution of a memory cycle is required for each word transferred during an IO operation. During the execution of a memory cycle for an IO word transfer, internal operations are suspended while the Central Computer System executes a machine action called a break cycle. A word supplied by an IO unit is written into core memory during a break-in (BI) cycle. Conversely, a word is read out of core memory and delivered to an IO unit during a break-out (BO) cycle.

Break cycles are executed upon request from an IO unit. This break request is usually received during the execution of an internal machine cycle. As soon as this internal machine cycle is completed, a break cycle is executed instead of an internal machine cycle. (One exception exists: a break cycle can be executed during an arithmetic pause.) Upon completion of the break cycle, the Central Computer System returns to the execution of internal machine cycles until another break cycle is requested.

3.1.3.2 Cycle Timing

The events within a machine cycle are initiated and timed by two series of pulses with a basic recurrence rate of 2 megacycles. Within the 6-microsecond duration of a machine cycle, twelve 2-megacycle pulses can occur. Each pulse is numbered from 0 through 11₁₀. Thus, a specific time within a machine cycle can be identified in terms of the type of cycle and the pulse number; e.g., PT 6, OT A-4, BI 8, etc.

The basic 2-megacycle pulses are sorted onto two sets of 12 lines to generate time pulses (TP) and instruction pulses (IP) for control of Central Computer System operation. The latter are used to generate commands during the execution of internal machine cycles.

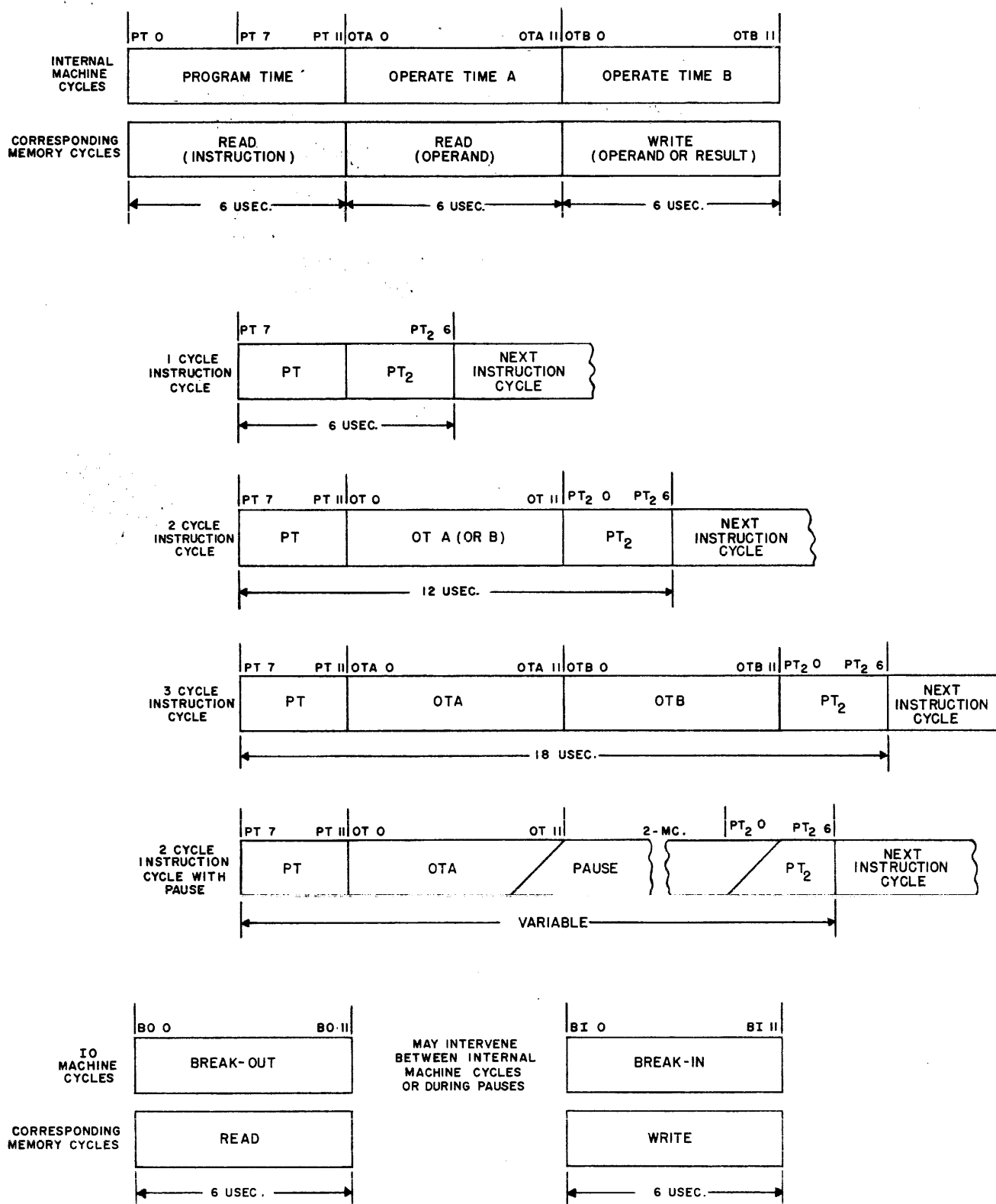
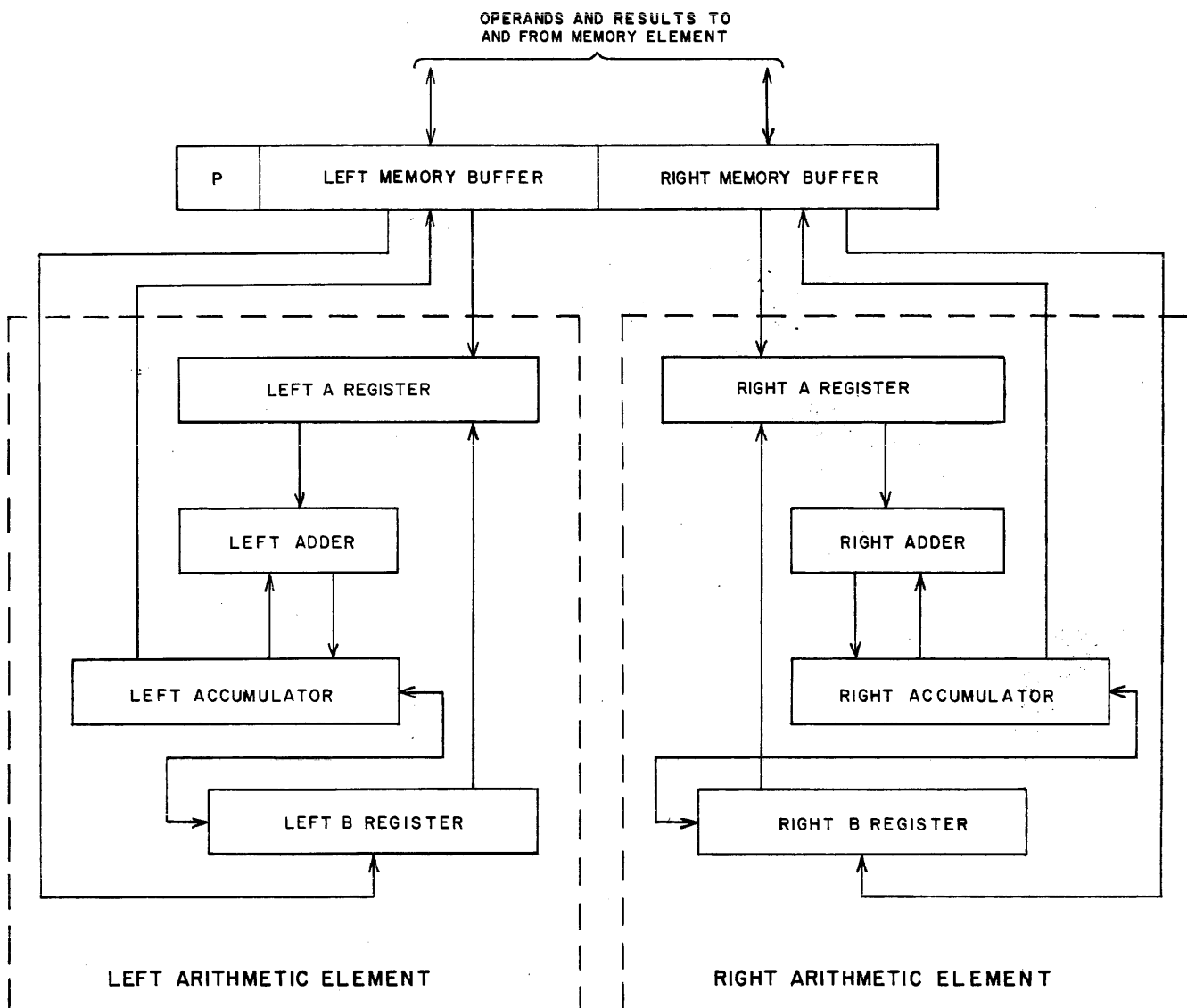


Figure 1-17. Machine and Instruction Cycles



NOTE:
COMMANDS NOT SHOWN

Figure 1-18. Arithmetic Element Information Flow

Time pulses (TP) are used to generate common commands for the execution of internal machine cycles and to generate all commands necessary for the execution of break cycles. The suspension of internal operations during a break cycle is accomplished by shutting off the instruction pulses (IP), thus preventing the generation of commands for the execution of internal machine cycles.

The execution of an arithmetic pause requires neither time pulses (TP) nor instruction pulses (IP). Instead, 2-megacycle pulses are used for the generation of commands, allowing the generation of a specific command at $1/2$ -microsecond intervals rather than at 6-micro-

second intervals, as is the case when a specific IP is used to generate the command. During an arithmetic pause, time pulses may be turned on to execute a break cycle. Instruction pulses must remain off during an arithmetic pause, since they might interfere with the use of 2-megacycle pulses in command generation.

3.1.3.3 Instruction Cycles

An instruction cycle is defined as the time interval allotted for the execution of an instruction. This time interval begins with the decoding of the instruction to be executed. Therefore, an instruction cycle begins at the instant the instruction is available for decoding, at

PT 7. The period within the program time cycle from PT 0 through PT 6, the 3 microseconds in which the instruction is read from core memory, cannot be considered to be within the interval allotted for the execution of the instruction being obtained from core memory during that program time cycle. Rather, it is included in the preceding instruction cycle. In like manner, a given instruction cycle ends at PT₂ 6 of the program time cycle in which the following instruction is read from core memory.

Figure 1-17 shows the sequence of machine cycles possible in different instruction cycles. As can be seen from the figure, a given instruction cycle runs from PT₁ 7 of one program time cycle through PT₂ 6 of the next program time cycle. The number and type of machine cycles within a given instruction cycle are determined by the operation called for by the instruction. An instruction which calls for no operand from core memory is executed without any OT cycle between the two halves of the PT cycles. An instruction which does call for an operand requires an OT cycle between the two PT cycles.

3.1.4 Operating Units

3.1.4.1 Functional Grouping

Although logical division of the Central Computer System into elements is made, it is more convenient in programming to ignore the element divisions and consider the Central Computer System as a group of flip-

flop registers. These registers fall into three functional types:

- Computing registers
- Control registers
- Buffer registers

Computing registers are those registers directly involved in the arithmetic and logical processing of information. Control registers perform control functions either in internal or IO operations of the Central Computer System. Buffer registers are intermediate storage registers that allow more rapid transfer of information between other units. The memory address register is an example of a control register. (Refer to 2.1.4.) The memory buffer register is primarily a buffer register, although it performs other functions as well. (Refer to 2.1.5.) The other buffer registers, the IO and IO buffer registers, do not require any extended discussion.

3.1.4.2 Computing Registers

The three computing registers in each half of the arithmetic element include the A register, the accumulator register, and the B register. A simplified diagram of the information flow paths between these registers and the memory buffer register is given in figure 1-18. Also shown on the diagram is a group of circuits called an adder.

The arithmetic process of addition (by which the other three arithmetic processes are performed) is ex-

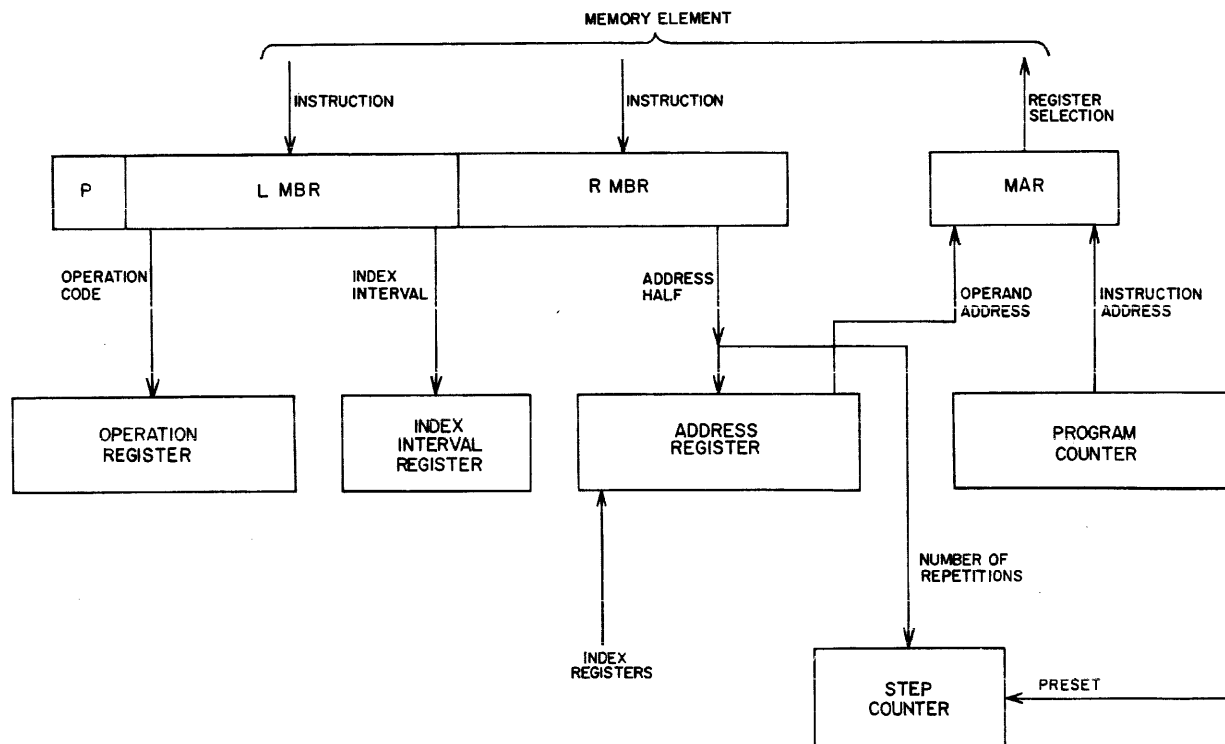


Figure 1-19. Internal Control Information Flow

cuted by the adder circuits together with the A register and the accumulator. Initially, the augend is in the accumulator while the addend is in the A register. The state of each flip-flop in the two registers is indicated to the adder circuits by pairs. When the adder circuits are pulsed, the sum is generated and placed in the accumulator register, replacing the augend. The addend remains in the A register. (This description applies to both halves of the arithmetic element. Therefore, two pairs of numbers are added, one pair in each half.) The same general procedure is followed in subtraction; however, the subtrahend in the A register is complemented before addition to the minuend in the accumulator. The difference appears in the accumulator, replacing the minuend.

The B register is used in multiplication and division. Essentially, the B register extends the capacity of the accumulator in these operations. The 30-bit plus sign product resulting from the multiplication of two numbers appears in the combined accumulator-B registers. Similarly, in division, the combined accumulator-B registers hold the dividend and the remainder. The extra capacity is needed to continue the division process until a 16-bit quotient is obtained. (Both multiplication and division are performed in the two halves of the arithmetic element, processing two pairs of numbers simultaneously.)

3.1.4.3 Internal Control Registers

The control registers connected with internal operations, shown in figure 1-19, include:

- a. Program counter
- b. Operation register
- c. Index interval register
- d. Address register
- e. Memory address register (discussed in Section 2)
- f. Index registers
- g. Step counter

The program counter controls the sequence of execution of instructions. At the start of each program time cycle, the contents of the program counter are transferred to the selected memory address register to obtain the instruction to be performed from core memory. The contents of the program counter are then increased by 1 to specify the address of the instruction to be obtained in the next program time cycle.

The operation part of an instruction, bits L1 through L10, is transferred during a program time cycle from the memory buffer register to the operation register, which drives three decoding matrices to identify the instruction and any modifications necessary for iterative routines.

The index interval register is similar to the operation register in that both registers receive part of an

instruction word to drive decoding matrices. The index interval register receives bits L10 through L15 of each instruction word. Although bit L10 is duplicated in the operation and the index interval registers, no difficulty results, since for an instruction using the index interval, bit L10 is assigned no meaning in the operation register.

The function of the index interval varies with the instruction containing it. The index interval is named from its use in controlling iterative loops. It may also select an IO unit or some other type of unit, a mode of Drum System operation, or the response to overflow in those instructions which may cause an overflow. These functions are discussed separately with the instructions using them.

The address register stores the address half of an instruction word from PT 7 through OT 1 for those instructions using an operate time cycle in their execution. At OT 1, the core register address part of the address register contents are transferred to the memory address register selected by the unit selection part of the address register contents. (Refer to 2.4.) This storage period makes possible the modification of the address part of an instruction being executed while in the address register without changing the address part of the instruction as stored in core memory. Modifications are made when an instruction is indexed in performing iterative loops. The address register also functions in branch operations (which change the sequence in which instructions are performed by changing the contents of the program counter).

The index registers are used to modify the address half of an instruction while it is stored in the address register. The contents of a selected index register may be added to the address part of an instruction stored in the address register. This process is called indexing. Four index registers are available for use in controlling iterative loops. In addition, the right accumulator register may be used to index an instruction. Indexing is described in 3.2.3 and in Section 5.

The step counter is a 6-bit binary counting register provided to regulate the length of arithmetic pauses. The step counter counts the number of 2-megacycle pulses provided to execute a particular operation involving repetitions. When such an operation is called for by an instruction, the step counter is set to the number of repetitions specified by bits R10 through R15 of the address part of the instruction. The operation is executed with the step counter contents reduced by 1 for each repetition. The 2-megacycle operation is usually ended when the step counter steps to 0. If the number of repetitions is inherent in the nature of the operation, the step counter is preset to the required number by a specific command without reference to the address half of the instruction.

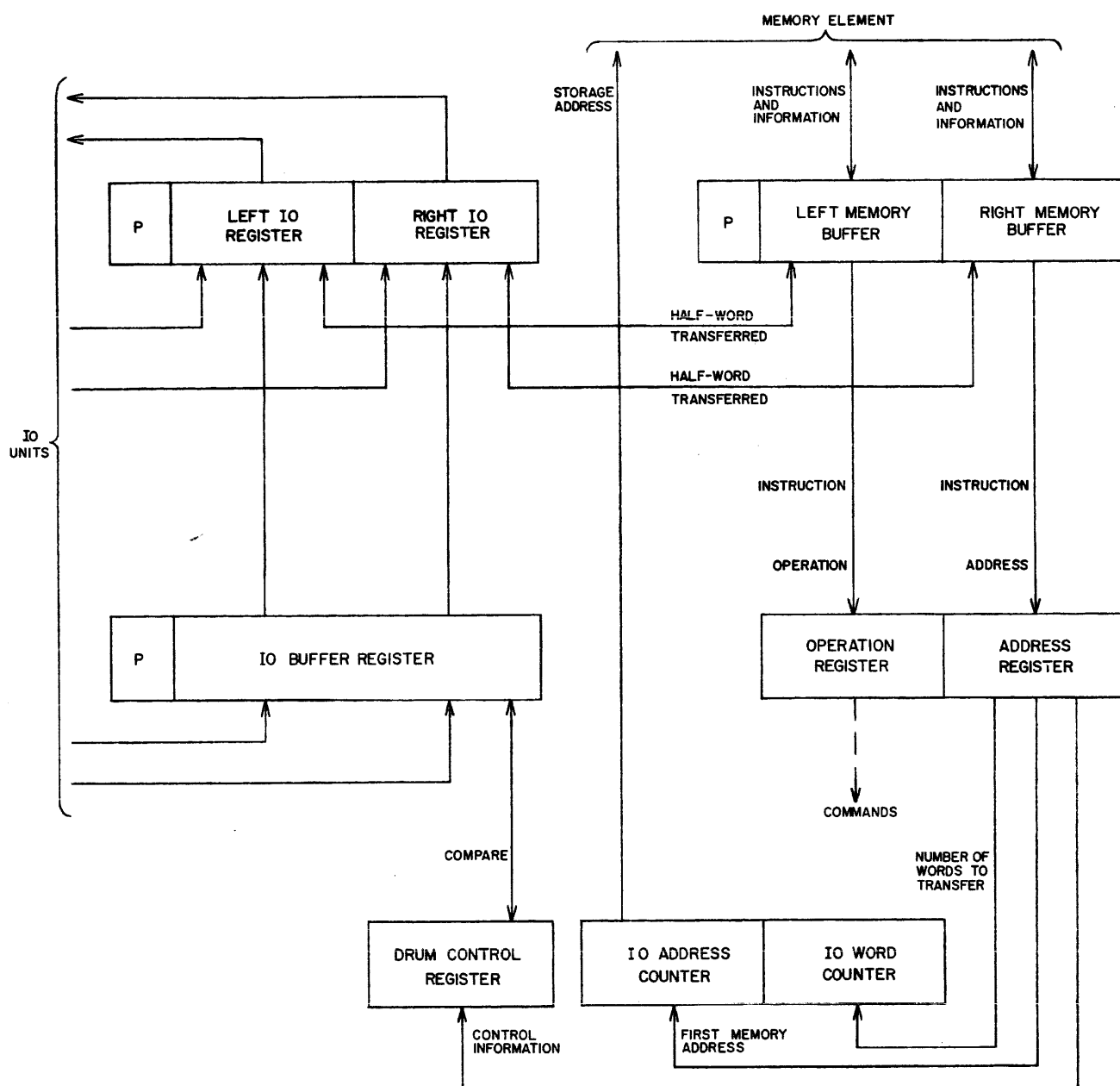


Figure 1-20. IO Control Information Flow

3.1.4.4 IO Control Registers

The control registers connected with IO operations, shown in figure 1-20, include the IO address counter, the IO word counter, and the drum control register. The IO address counter controls the selection of registers in core memory to be affected by an IO operation. Initially, the IO address counter is set to the address of the first core memory register to which or from which a word is to be transferred. At the start of the first break cycle, this address is transferred to the memory address register and the contents of the IO address counter are increased by 1. In effect, the IO address counter functions

for IO operations as the program counter does for internal operations.

The IO word counter controls the number of words transferred during a given IO operation. The IO word counter is initially loaded with the 1's complement of the number of words to be transferred, and a 1 is added to it. As each break cycle is executed, another 1 is added to the IO word counter. Thus, at each point in the IO operation, the IO word counter contains the 1's complement of one less than the number of words remaining to be transferred. As the last word is transferred, a carry-out is generated by the IO word counter. This carry-out pulse ends the IO operation.

The drum control register controls certain aspects of IO transfers involving the Drum System. Its functions are discussed in connection with that system in Chapter 4.

3.2 INSTRUCTIONS

3.2.1 Classification

Each one of the 48 instructions recognized by the Central Computer System is indicated to that system as one of 16 possible variations within one of 8 possible classes. In general, all instructions within a given class are executed similarly and are variations within that class. Although an instruction word can indicate any one of the 128 class and variation combinations (8 times 16), only 48 of these are considered legal instructions; the remaining combinations are termed illegal instructions.

The operation code of an instruction word (L4 through L10) is divided into two parts, as shown in figure 1-21:

- Class indicator, L4 through L6 (one octonary digit)
- Variation indicator, L7 through L10 (two octonary digits)

TABLE 1-11. INSTRUCTION CLASSES

CLASS	CLASS INDICATOR		NUMBER OF VARIATIONS	
	BINARY	OCTONARY	LEGAL	ILLEGAL
Add	001	1	9	7
Multiply	010	2	4	12
Store	011	3	7	9
Shift	100	4	8	8
Branch	101	5	6	8
IO	110	6	5	9
Reset	111	7	3	13
Miscellaneous	000	0	6	9

Each class, its class indicator, and the number of legal illegal variations are listed in table 1-11.

The remaining parts of the left half of an instruction word are the branch control bit, LS; the index indicator, L1 through L3 (one octonary digit); and the index interval, L10 through L15 (two octonary digits). The index indicator controls the indexing of an instruction. If an instruction is indexable, the index indicator specifies the index register whose contents are to be added to the address part of the instruction. (Refer to 3.2.3.)

The index interval can perform control functions in some instructions; it is meaningless in others. It is used in controlling indexing to change the contents of a selected index register by a prescribed amount after each repetition of an iterative loop. The other control functions of the index interval are discussed in connection with specific instructions.

Each legal instruction has a name descriptive of the operation it initiates. In addition, a 3-letter mnemonic code is assigned to each instruction. In general, once the mnemonic code for an instruction is given, the code is used in all subsequent references to the instruction. When reference is made to the actual contents of part of an instruction word, octonary notation is used for the sake of brevity. In references to the address half of an instruction, the letter *x* is used to designate a core memory address while the letter *n* indicates an address half with quantitative significance.

3.2.2 Decoding

All instructions are decoded in precisely the same manner. Figure 1-22 shows the control units which function to obtain an instruction from core memory and to decode that instruction. Each instruction is obtained from the address in core memory specified by the contents of the program counter. The contents of the program counter are transferred to the selected memory address register at PT 1 of the preceding instruction cycle, initiating the memory cycle in which the instruction is read from core memory. The preceding instruc-

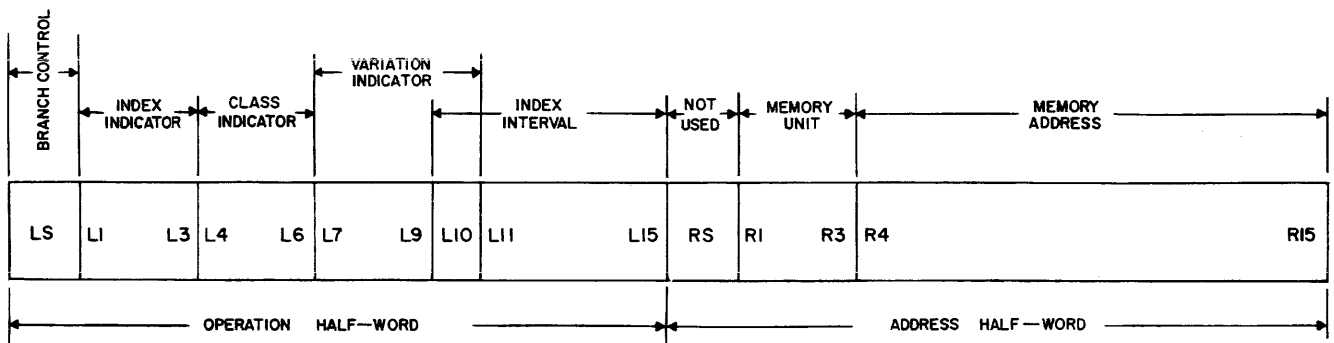


Figure 1-21. Instruction Flow

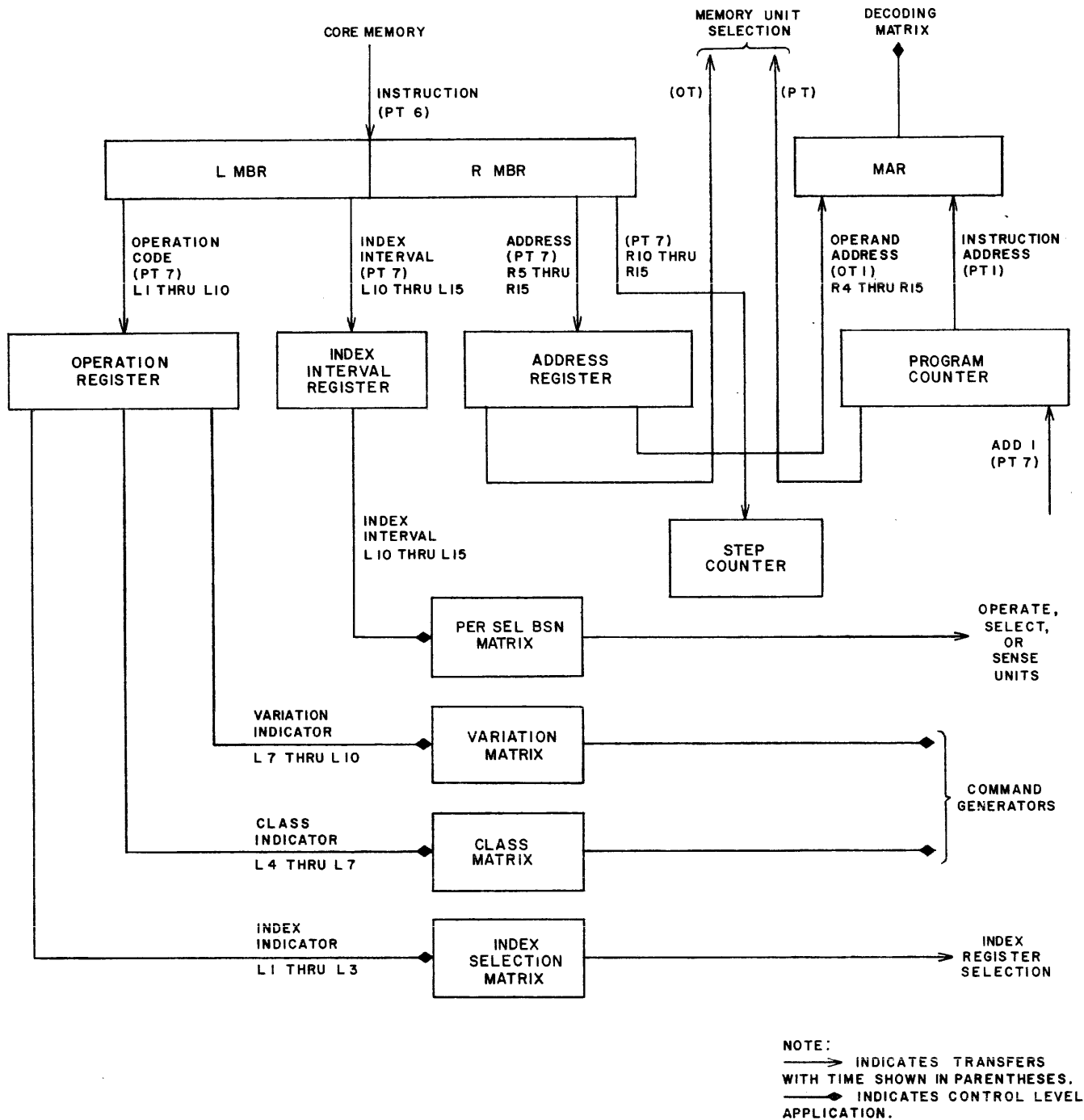


Figure 1-22. Instruction Decoding

tion cycle is completed by PT 6 and the instruction to be decoded is in the memory buffer register.

The instruction cycle during which the instruction is executed begins at PT 7 with four events. The instruction is transferred in four parts from the memory buffer register to four other registers; the operation part (L1 through L10) is transferred to the operation register; the index interval (L10 through L15) is transferred to the index interval register; the address half-word is

transferred to the address register; and R10 through R15 are transferred to the step counter. At the same time, the contents of the program counter are increased by 1. Thus, for the duration of the instruction cycle, the program counter contains the address of the next instruction to be executed.

The operation register drives three decoding matrices for the duration of the instruction cycle; these are the index selection matrix, the class matrix, and the

variation matrix. Similarly, the index interval register drives a decoding matrix whose output is or is not used, in accordance with the instruction containing the index interval. In certain cases, the output of the index interval decoding matrix is meaningless. The function of the index selection matrix is discussed in 3.2.3.

The class matrix and variation matrix determine the generation of the commands necessary to execute the indicated instruction. These commands are actually instruction pulses gated onto appropriate lines by conditioning levels from the class and variation matrices. The time sequence of the commands is established by using appropriate instruction pulses. In general, the class matrix also determines the length of the instruction cycle, in accordance with the type of instruction to be executed.

The time within the instruction cycle from PT 7 through PT 11 may be used for instruction execution in a one-memory-cycle instruction. For a two-memory-cycle instruction, the time is available for indexing. During this period, the address half of the instruction is retained in the address register. At OT 1, the memory address (R4 through R15) is transferred from the address register to the memory address register, thus initiating a second memory cycle within the instruction cycle. The operand so obtained is available for operation at OT 7. The instruction cycle continues through to PT 6, at which time the operation register is cleared in preparation for the next instruction cycle.

3.2.3 Indexing

During the interval from PT 7 to OT 1, in which the address part of the instruction is held in the address register, that address may be modified by a process known as indexing. If an instruction is indexed, the instruction obtains the operand from core memory location $x+i$ (where i is the number contained in the selected index register) rather than the operand at location x . Before the address is transferred to the memory address register, the contents of the selected index register can be added to the contents of the address register. If i can be changed before each indexed repetition of the instruction, then the instruction can process many different operands.

Figure 1-23 shows the units involved in the indexing of an instruction. The index indicator (L1 through L3) of the instruction word selects the index register whose contents are to be added to the address half of the instruction. If the index indicator is 0₈, no indexing occurs. If the index indicator is 1, 2, 4, or 5₈, then the contents of the corresponding index register are added to the address register at PT 9. The indexed address is transferred to the memory address register at OT 1. An index indicator of 3₈ selects the right accumulator for

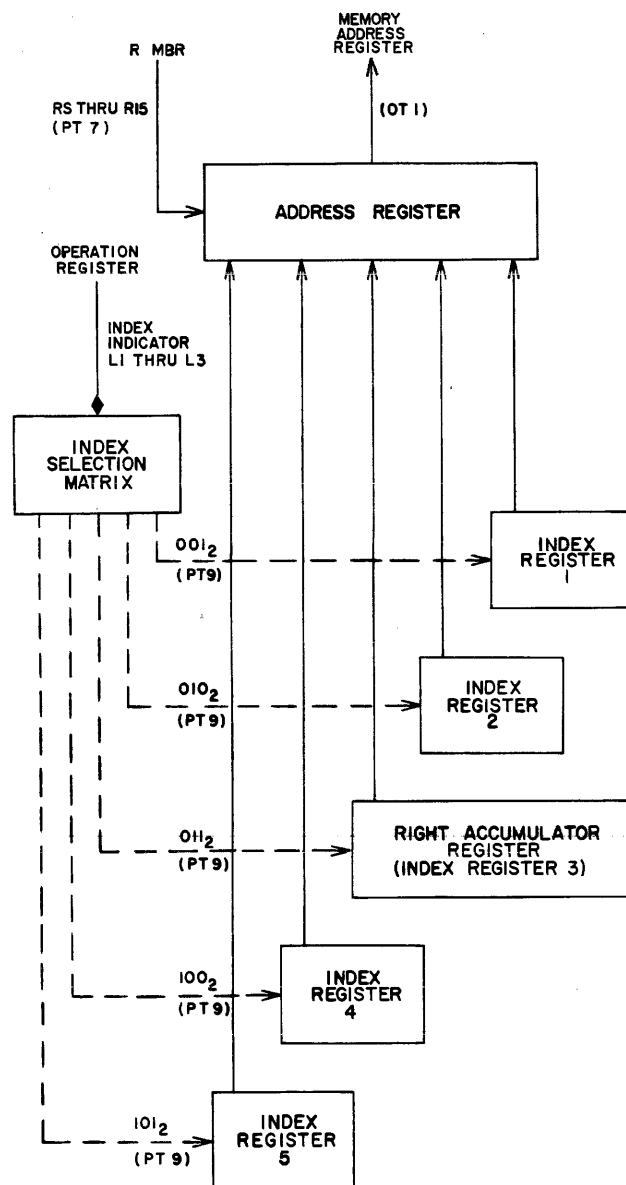


Figure 1-23. Indexed Address Modification

addition to the address register. Indexing is discussed in greater detail in Section 5.

3.3 ADD CLASS INSTRUCTIONS

3.3.1 Principles of Execution

The add class includes nine separate instructions, all performed in the same basic manner. The basic circuit in addition is the adder circuit. (See fig. 1-24.)

The basic adder circuit produces a sum term and a carry term from a pair of binary bits and a carry from a preceding adder. An adder circuit is connected between corresponding flip-flops of the A register and the accumulator. The bit contents of a particular pair of flip-flops determine which of the four AND circuits is conditioned. No output is produced until a pulse is applied

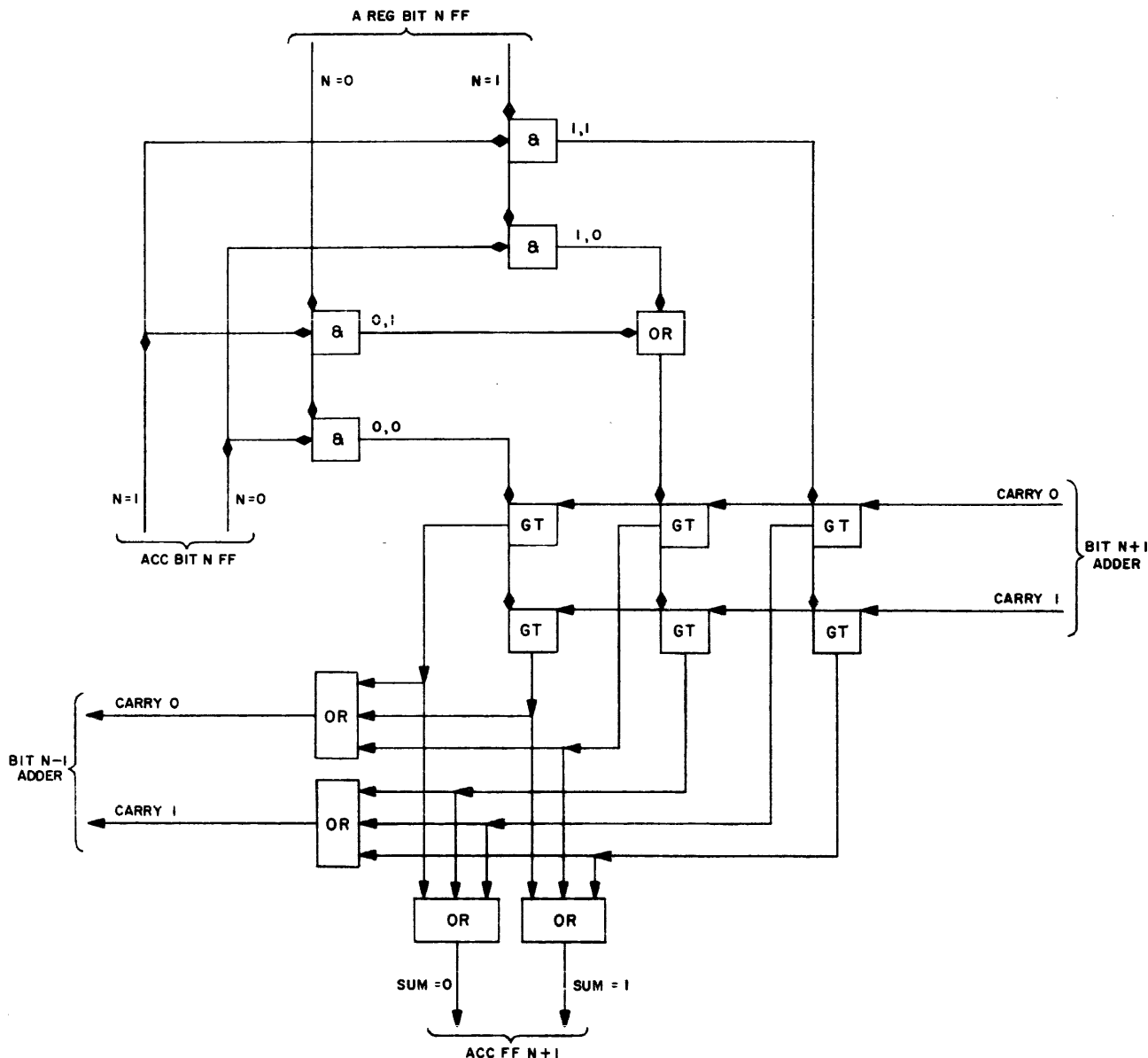


Figure 1-24. Adder Circuit

on one of the incoming carry lines. The incoming carry pulse is gated onto one of the output sum lines and onto one of the output carry lines. The carry-out indicates the addition process in the next adder, while the sum line is used to set a flip-flop in the accumulator to the state appropriate to represent that term of the sum.

The sum outputs of each adder are connected to produce a shift right of one position in storing the sum in the accumulator. (See fig. 1-25.) The B register sign flip-flop is used for temporary storage of the least significant sum term. The inherent shift right is incorporated into the adder circuits to reduce the time required for multiplication but has no material effect upon the execution time of addition. The inherent shift right is

corrected for by a correctional shift left which is executed automatically during an addition operation. If a carry-out of 1 from the sign bit adder is produced, the carry is stored in a flip-flop until the correctional shift left is completed. The end-carry is then completed by pulsing the carry 1 line into the bit 15 adder, followed by another correctional shift left.

In general, the add class of instructions is executed by placing an operand in the A register and then pulsing the carry 0 line of the bit 15 adder to add this operand to the word already in the accumulator. The sum generated is stored in the accumulator (shifted right one position). The correction shift left is executed, followed by the end-carry (if any) and another correctional shift

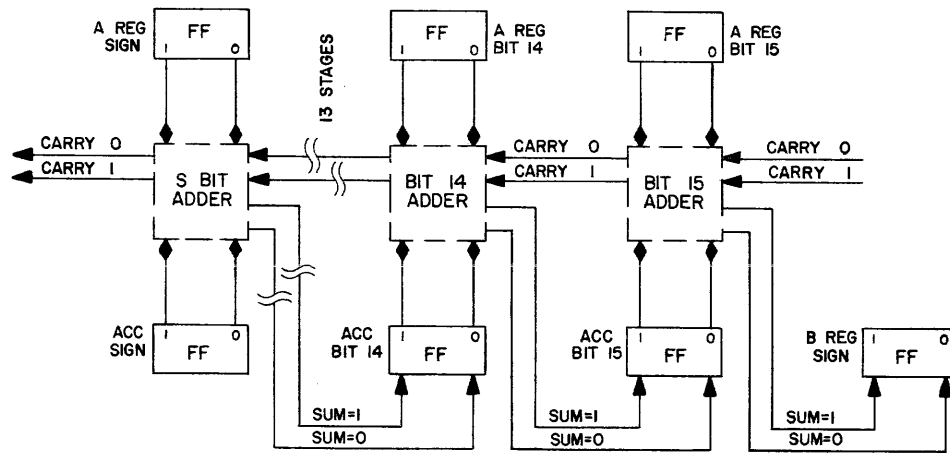


Figure 1-25. Inherent Shift Right

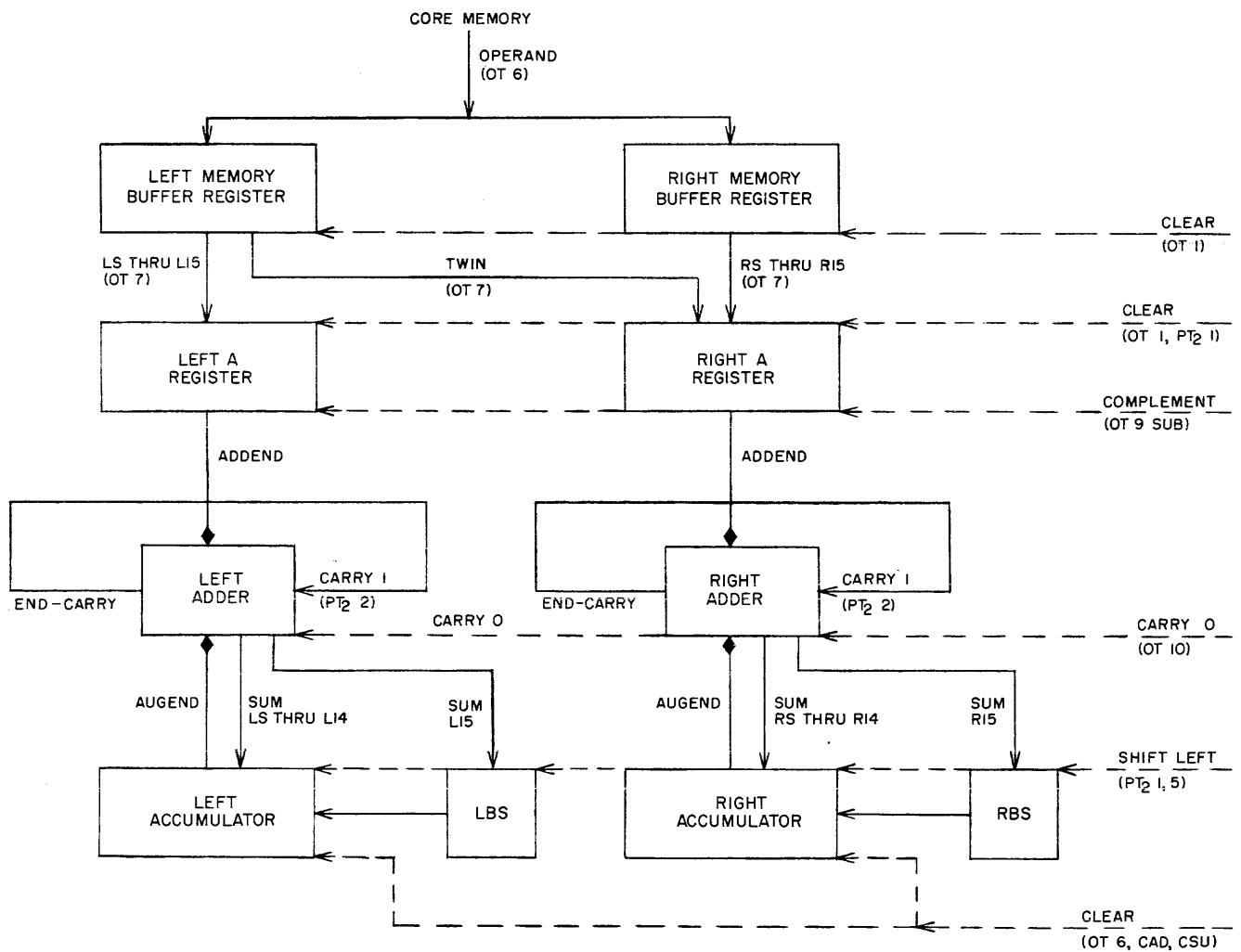


Figure 1-26. Add Class Execution

left (if required). A single addition requires 1 microsecond (maximum) and a correctional shift left requires $\frac{1}{2}$ microsecond. Therefore, the time required for an addition with end-carry is approximately 3 microseconds, while an interval of 6 microseconds is available between the time the operand is available and the end of the instruction cycle. Therefore, the extra time required to execute the correctional shifts left is negligible.

3.3.2 Typical Instruction

The *Add (ADD)* instruction is perhaps the most typical instruction in the add class; figure 1-26 shows the registers involved in its execution. The instruction, like all instructions of the add class, requires a two-memory-cycle instruction cycle of 12 microseconds and is indexable. (Refer to 3.2.3.)

An *ADD* instruction is decoded in a manner similar to any other instruction. Since an operand from core memory is required, execution of the instruction proper is delayed until the operand is available. The operand is available for transfer out of the memory buffer register at OT 7 to the A register. At OT 10, the carry 0 line into the adder is pulsed. The initial sum, shifted right, appears in the accumulator-B sign register by PT₂ 0. At PT₂ 1, the correctional shift left is executed and the A register is cleared. At PT₂ 2, if an end-carry is necessary, the carry 1 line is pulsed, followed by a correctional shift left at PT₂ 5. At the completion of the instruction cycle, the sum remains in the accumulator register, while the memory buffer and A registers are cleared. (The A register is cleared before the end-carry is started to prevent reading its contents into the accumulator.)

3.3.3 Variations

The variations within the add class are of several types. The *Clear and Add (CAD)* instruction, by clearing the accumulator before adding the operand, places the operand unchanged in the accumulator. The *Clear and Subtract (CSU)* instruction is identical with the *CAD* instruction except that the complement of the operand is placed in the accumulator. The *Subtract (SUB)* instruction differs from *ADD* only in the complementing of the A register before adding its contents to the accumulator, thus subtracting the operand from the accumulator.

The twin variations, *Twin and Add (TAD)* and *Twin and Subtract (TSU)*, differ from *ADD* and *SUB* (respectively) in their use of the twin transfer path shown in figure 1-26. The twin transfer path causes the use of the left half-word in both the left and right arithmetic elements; the right half-word of the operand is not used.

The *Add B Registers to Accumulators (ADB)* instruction requires no operand from core memory. Instead, the contents of the B registers are added to the

accumulators via the A registers as if they had been read from core memory. The address half of the *ADB* instruction is meaningless, thus making indexing of the instruction also meaningless. The *Clear and Add Magnitudes (CAM)* instruction clears the accumulator and then places the positive absolute magnitude of the operand in the accumulator. The *CAM* instruction is thus equivalent to *CAD* for a positive operand or to *CSU* for a negative operand. The *Difference Magnitudes (DIM)* instruction generates the difference in absolute magnitude between two numbers in this manner. (See fig. 1-27.) The accumulator contents are first duplicated in the B register. The accumulator contents, if negative, are then complemented. The operand from core memory is placed in the A register, made positive to insure a known starting condition, and then complemented and added to the accumulator. The result of the instruction is that the original contents of the accumulator are in the B register while the accumulator contains the difference in absolute magnitude between that original

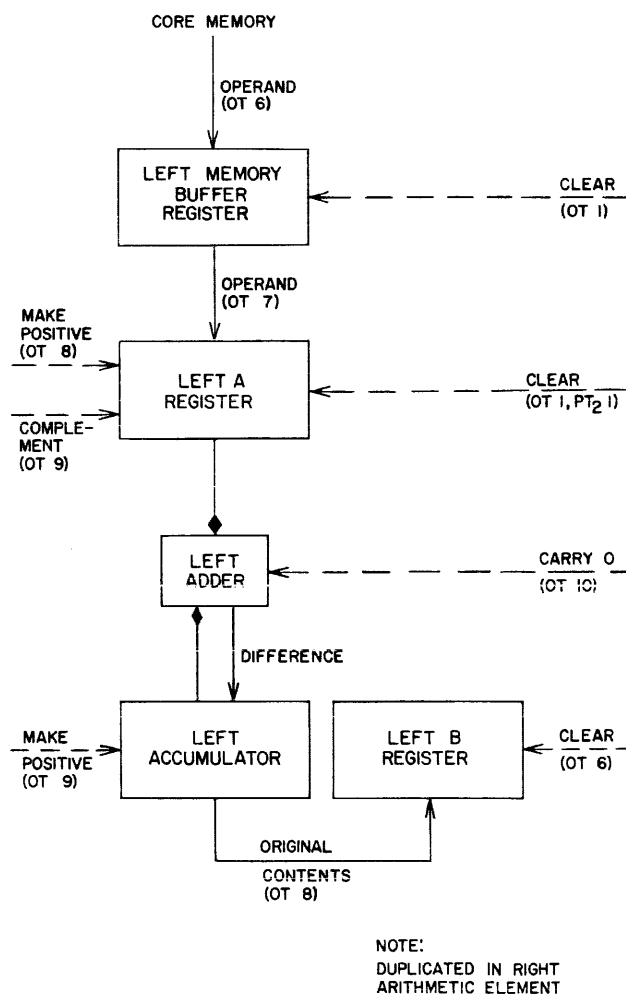


Figure 1-27. Difference in Magnitudes (DIM) Execution

value and the operand from core memory. If the accumulator contents are positive, then the operand was smaller than the original contents of the accumulator.

3.3.4 Overflow Control

Five instructions of the add class may cause an overflow. (Refer to 3.4.5, Chapter 2.) These include *ADD*, *TAD*, *ABD*, *SUB*, and *TSU*. The occurrence of an overflow in either half of the arithmetic element causes the setting of a flip-flop called an overflow sense unit. Either the right or the left overflow sense unit may be examined by a *Sense (BSN)* instruction, modifying the program of the Central Computer System if an overflow has occurred. In addition, each instruction that may cause an overflow can be written to generate an overflow alarm which can be used to take immediate action upon detection of an overflow. (The exact action taken in response to an overflow alarm, when generated, is determined by switch settings on the duplex maintenance console.) Bits L14 and L15 of the index interval within the instruction word specify what overflow conditions will generate an overflow alarm. The contents of these two bit positions (given binarily as well as with the octonary representation of the entire index interval) have the following significance:

- a. 00 — no alarm generation
- b. 01 — generate alarm on right overflow only
- c. 10 — generate alarm on left overflow only
- d. 11 — generate alarm on either overflow

When the index interval is used to control response to an overflow, its use is indicated generically by the use of the letter (*o*); e.g., *ADD (o) x*. The appearance of this

letter after the mnemonic code for an instruction thus indicates that the instruction is capable of causing overflow.

Table 1-12 provides a summary of information on the add class of instructions.

3.4 MULTIPLY CLASS INSTRUCTIONS

3.4.1 General

No truly typical instruction can be chosen from among the four instructions in the multiply class. The class contains two basic instructions, *Multiply (MUL)* and *Divide (DVD)*, and their twin variations. Since each instruction requires the execution of repeated additions or subtractions, an arithmetic pause is provided to extend the execution time allowed for the operation. The inherent shift right provided by the wiring of the adders significantly reduces the required execution time of the *MUL* instruction. On the other hand, the inherent shift right extends the time requirements for the *DVD* instruction. All multiply class instructions are indexable.

3.4.2 Multiply

Execution of a *Multiply (MUL)* instructions produces, in each half of the arithmetic element, the product of two signed 15-bit binary numbers. Such a product contains 30 significant bits plus a sign bit. To contain this product, the accumulator and B registers are combined into a 32-bit register. Only the registers of the left arithmetic element involved in the execution of a *MUL* instruction are described since the actions performed in the right half are identical. A *MUL* instruction is decoded in the same manner as used for all other instructions. (Refer to 3.2.2.) The address half

TABLE 1-12. ADD CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES			EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC		OCTONARY		
<i>Clear and Add</i>	<i>CAD</i>	<i>x</i>	100	12	Yes
<i>Add</i>	<i>ADD (o)</i>	<i>x</i>	104	12	Yes
<i>Twin and Add</i>	<i>TAD (o)</i>	<i>x</i>	110	12	Yes
<i>Add B Registers to Accumulators</i>	<i>ABD (o)</i>	—	114	12	Meaningless
<i>Clear and Subtract</i>	<i>CSU</i>	<i>x</i>	130	12	Yes
<i>Subtract</i>	<i>SUB (o)</i>	<i>x</i>	134	12	Yes
<i>Twin and Subtract</i>	<i>TSU (o)</i>	<i>x</i>	140	12	Yes
<i>Clear and Add Magnitudes</i>	<i>CAM</i>	<i>x</i>	160	12	Yes
<i>Difference Magnitudes</i>	<i>DIM</i>	<i>x</i>	164	12	Yes

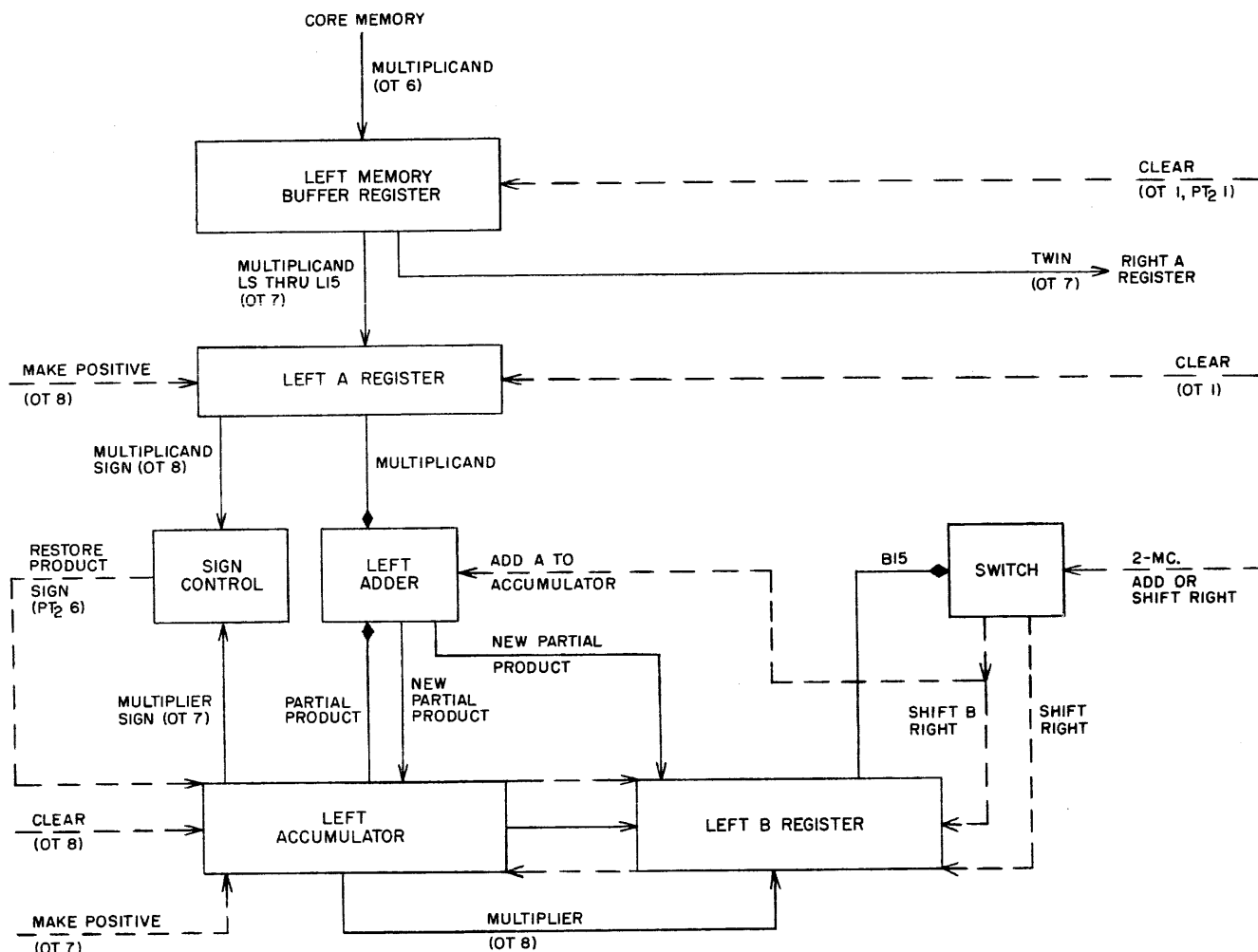


Figure 1-28. Multiply (MUL, TMU) Execution

of the instruction which specifies the multiplicand to be obtained from core memory may be modified by indexing. The multiplier is left in the accumulator by a previous instruction. Multiplication is executed as described mathematically in 3.5.2, Chapter 2. The procedure of multiplication by addition and shifting, set forth there, is as follows:

1. Determine the sign of the product by the rule of signs, then obtain absolute magnitudes of the multiplier and multiplicand by complementing if negative.
2. Form partial product:
 - a. Examine the least significant bit of the multiplier not already examined; if 1, add the multiplicand to the partial product; if 0, take no action.

b. Shift the new partial product right one position.

3. Repeat step 2 for each significant bit position of the multiplier (not for sign bit).
4. After the last repetition of step 2, correct the sign of the product according to the sign of product determined in step 1.

Calculation following these rules begins with the obtaining of the multiplicand from core memory at OT 7. (See fig. 1-28.) At the same time, the multiplier is made positive (complemented if negative) and its original sign is stored in the sign control flip-flop. At OT 8, the multiplier is shifted into the B register, the accumulator is cleared, and the multiplicand in the A register is made positive. The step counter (not shown in figure 1-28) is also set at OT 8 to 15, to control the number of 2-megacycle pulses used to execute the actual multiplication.

The multiplication process is executed using 2-megacycle pulses to initiate the additions and shifting required to form the product. Each 2-megacycle pulse is applied to a switching circuit conditioned by the contents of B15. If this bit (least significant multiplier bit not yet examined) is 1, the carry 0 line to the adder is pulsed, adding the multiplicand to the accumulator, and the B register is shifted right one position. (The inherent shift right from the adder eliminates the need to shift the accumulator to the right.) The shift right of the B register eliminates the already examined bit of the multiplier and moves the next significant bit into the B15 position. If the bit examined by the 2-megacycle pulse is a 0, the combined accumulator-B register is shifted right one position without adding the multiplicand. Thus, each 2-megacycle pulse causes the execution of step 2 of the previously noted multiplication procedure. After 15 pulses have been applied, the multiplication is complete except for the restoration of sign.

The instruction cycle for multiplication is shortened by allowing the operate time (OT) cycle to be completed during 2-megacycle operation. Similarly, the following program time (PT) cycle is begun when the step counter reaches 5. Thus, by PT₂ 5, the 2-megacycle operation is complete, allowing the restoration of sign at PT₂ 6. The total instruction cycle runs 17 ± 0.5 microseconds instead of the 19.5 microseconds required for a two-memory-cycle instruction cycle (12 usec) plus the 15 repeated 2-megacycle operations (7.5 usec). (The 0.5-microsecond variable time allows for delay in synchronization of 2-megacycle pulses.)

The bit in position 15 of the B register is not a significant bit of the product. It must be 0 before restoration of sign, since the 15th shift of the multiplier (made positive at the start of multiplication) places the sign bit of the multiplier in the B15 position. After sign restoration, this bit is identical in content to the sign of the product.

3.4.3 Divide

Execution of a *Divide* (DVD) instruction produces, in each half of the arithmetic element, the quotient of a signed 15-bit divisor and a signed dividend of up to 31 significant bits. The resulting quotient is an unsigned binary number of 16 significant bits. The dividend is held initially in the combined accumulator-B register, the divisor in the A register. The quotient appears unsigned in the B register while any remainder and the sign of quotient and remainder is left in the accumulator. Although only the operation of the left arithmetic element is described, it should be remembered that an identical operation is performed in the right arithmetic element.

A DVD instruction is decoded in the same manner as that used for all other instructions. (Refer to 3.2.2.)

The address half of the instruction may be modified by indexing, thus obtaining a divisor other than the one specified by the original address half of the instruction. The dividend is placed in the accumulator (or accumulator-B register) by a preceding instruction. The mathematical description of division as performed in the Central Computer System was given in 3.6.3, Chapter 2. The procedure of nonrestoring division, set forth there, is as follows:

1. Determine the sign of the quotient by rule of signs, then make divisor and dividend positive to allow calculation with absolute magnitudes.
2. Perform a trial subtraction:
 - a. Make the sign of divisor unlike the sign of current remainder (dividend for first trial subtraction) by complementing the divisor and shift the remainder left one position.
 - b. Add the divisor if positive, or 2's complement of the divisor if negative, to the shifted current remainder; write a 1 in the quotient if the new remainder is positive (carry-out of 1) or a 0 in the quotient if the new remainder is negative (carry-out of 0).
3. Repeat step 2 once for each bit of the divisor including the sign bit.
4. If, after last repetition of step 2, the divisor and remainder are both negative, complement the divisor and add to the remainder, restoring the remainder to positive form. Ignore carry-out. Correct the sign of the quotient and remainder in accordance with the stored sign as determined in step 1.

Calculation following this procedure begins with the obtaining of the divisor from core memory. (See fig. 1-29.) At OT 7, the divisor is placed in the A register, the dividend is made positive (complemented if negative), and its original sign is stored in the sign control flip-flop. At the same time, the step counter (not shown in figure 1-29) is set to 17 to control the number of trial subtractions being performed. At OT 8, the divisor is made positive, and its original sign is stored in the sign control flip-flop. The sign control flip-flop is thus set to the correct quotient sign by the signs of the dividend and the divisor. Command generation with 2-megacycle pulses is called for at OT 8 and started at OT 9.

Since a trial subtraction cannot be performed with one command, 2-megacycle pulses cannot be used directly for the generation of commands to execute the DVD instruction. Instead, the 2-megacycle pulses are divided into a repetitive series of five pulses numbered from 0 through 4, called divide time pulses (DVTP). Each DVTP cycle causes the execution of one trial subtraction.

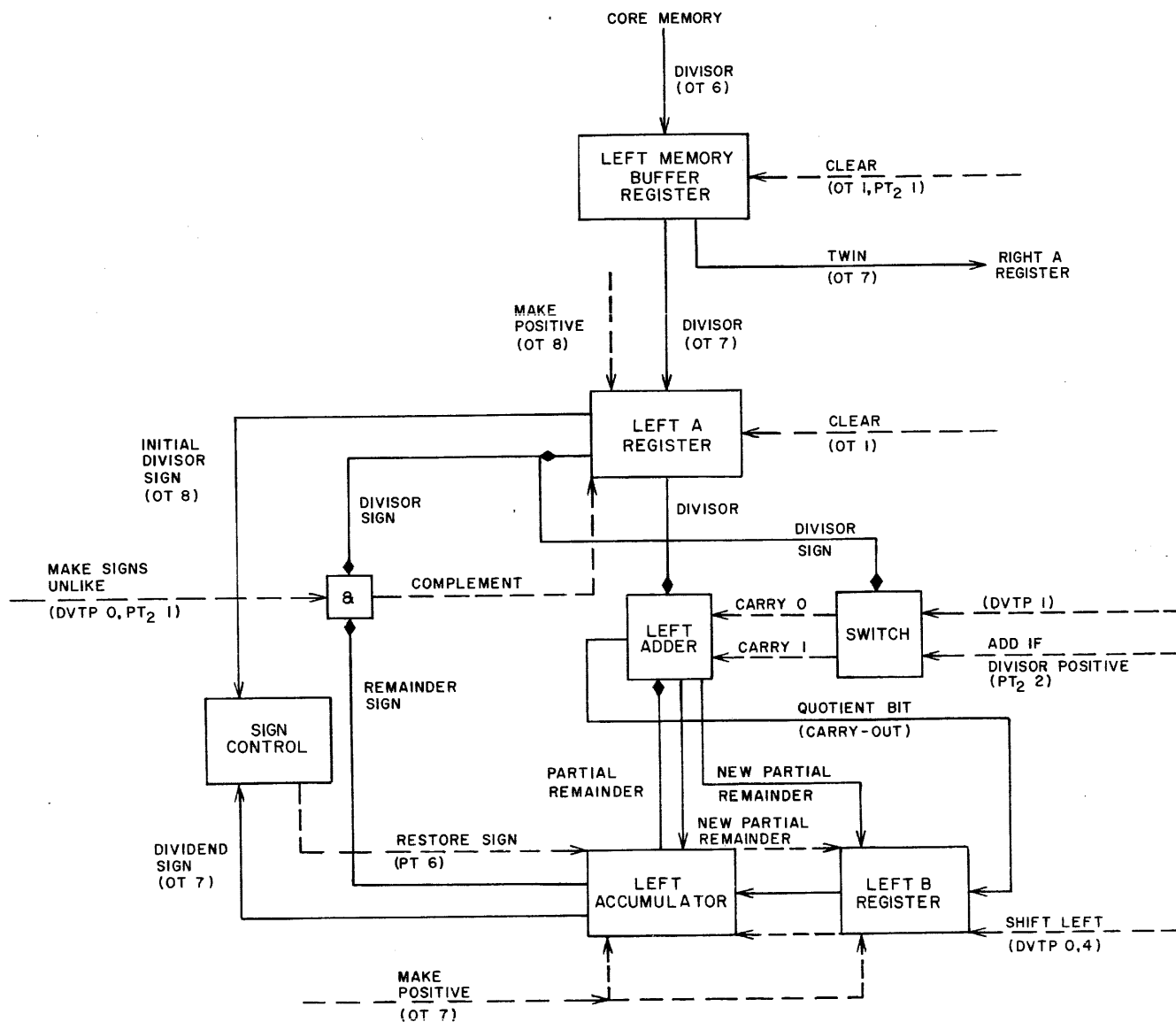


Figure 1-29. Divide (DVD, TDV) Execution

tion. DVTP 0 executes step 2-a of the rules of binary division (makes the divisor unlike in sign to the partial remainder and shifts the partial remainder left). DVTP 1 executes step 2-b (adds either the divisor by pulsing the carry 0 line or the 2's complement of the divisor by pulsing the carry 1 line when the divisor is in 1's complement form in the A register). DVTP 2 and 3 are not used in the arithmetic element and provide for a delay in that element sufficient to complete the addition. However, DVTP 3 does reduce the contents of the step counter by 1, indicating the completion of a trial subtraction. DVTP 4 produces a shift left to correct for the inherent shift right from the adder.

The sign of each partial remainder generated is indicated by the carry-out from the sign bit adder. This

carry-out is used, in each DVTP cycle, to set bit 15 of the B register to 1 or 0. In effect, each carry-out is stored as the quotient bit generated by that trial subtraction. Since the accumulator-B registers are shifted left (at DVTP 0) before each quotient bit is generated, the quotient is built up, bit by bit, in the B register until all 16 bits contain significant bits of the quotient. To generate 16 quotient bits, 16 trial subtractions are required. Although the step counter is set to 17 at the start of 2-megacycle operation, only 16 DVTP cycles are performed. As the step counter is stepped from 2 to 1 at DVTP 3 of the 16th trial subtraction, the 2-megacycle pulses are shut off, allowing only the generation of DVTP 4 to complete the 16th trial subtraction. At the same time, the next program time (PT) cycle is

started. If the remainder in the accumulator is negative, the restoration called for in step 4 of the rules of division is executed, using instruction pulses for command generation.

The restoration of the remainder is accomplished in a manner identical with the procedure of a trial subtraction. The carry-out is ignored, since the connection from the sign bit adder to B15 is disconnected at PT₂ 1, before the addition at PT₂ 2. Finally, at PT₂ 6, the sign of the remainder and quotient is corrected, if necessary, in accordance with the contents of the sign control flip-flop. The total *DVD* instruction cycle has a duration of 51.5 ± 0.5 microseconds, 12 microseconds within the required two memory cycles plus an arithmetic pause of 39.5 ± 0.5 microseconds between OT 11 and PT₂ 0. The *DVD* instruction thus produces, in each half of the arithmetic element, a 15-bit remainder (held in the accumulator, positions 1 through 15), and unsigned 16-bit quotient (held in the B register), and the sign bit of both the quotient and remainder (held in the accumulator sign bit position). In order to store the quotient in a standard half-word, it must be processed by a *Shift Left and Round (SLR)* instruction. (Refer to 3.10.6.)

3.4.4 Variations

Of the four instructions within the multiply class, two are twin variations on the instructions already described. Thus, *Twin and Multiply (TMU)* is simply a twin variation of *MUL*. A *TMU* instruction causes the left half-word brought from core memory to be used as multiplicand in both the left and right arithmetic elements. Similarly, the *Twin and Divide (TDV)* instruction causes the left half-word to be used as divisor in both the left and right arithmetic elements. Table 1-13 provides a summary of information on the multiply class of instructions.

3.5 STORE CLASS INSTRUCTIONS

3.5.1 Principles of Execution

The seven instructions within the store class are basically transfer instructions. Most of the instructions within the class transfer information from the arith-

metic element to the memory element for storage therein. Since many of the instructions require the storage of only half a word without destroying the half-word already in core memory, two OT cycles are required for their execution. In general, therefore, store class instructions require three-memory-cycle instruction cycles, or 18 microseconds. All instructions in the store class are indexable.

3.5.2 Typical Instruction

An instruction typical of the store class is the *Left Store (LST)* instruction. The *LST* instruction places the contents of the left accumulator in the left half-word positions of a selected core memory register without changing the contents of the right half-word positions in the register. The accumulators are unchanged by this instruction.

The *LST* instruction is decoded in the same manner as are other instructions. (Refer to 3.2.2.) The address half of the *LST* instruction may be modified by indexing during the interval from PT 7 through OT A-1. At OT A-1, the address of the location to be affected is transferred to the memory address register from the address register. Since a word cannot be read out, changed, and rewritten into core memory in one memory cycle, two OT cycles must be executed. (Refer to 2.1.6.3.) The OT A cycle for the *LST* instruction obtains the right half-word from core memory for temporary storage in the right A register. (See fig. 1-30.) The OT B cycle transfers the desired pair of half-words to the memory buffer registers to be written into the selected register in core memory. The total execution time for an *LST* instruction is 18 microseconds.

3.5.3 Similar Variations

The *Right Store (RST)* instruction differs from the *LST* instruction in that the contents of the right accumulator register replace the right half-word in the selected memory register. The *Store Address (STA)* instruction is similar to the *RST* instruction with the exception that the *STA* instruction stores the contents of the right A register rather than those of the right accumulator. The *Full Store (FST)* instruction places the contents of the

TABLE 1-13. MULTIPLY CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES		EXECUTION		INDEXABLE
	MNEMONIC	OCTONARY	(MICROSECONDS)		
<i>Multiply</i>	<i>MUL x</i>	250	17	± 0.5	Yes
<i>Twin and Multiply</i>	<i>TMU x</i>	254	17	± 0.5	Yes
<i>Divide</i>	<i>DVD x</i>	260	51.5	± 0.5	Yes
<i>Twin and Divide</i>	<i>TDV x</i>	264	51.5	± 0.5	Yes

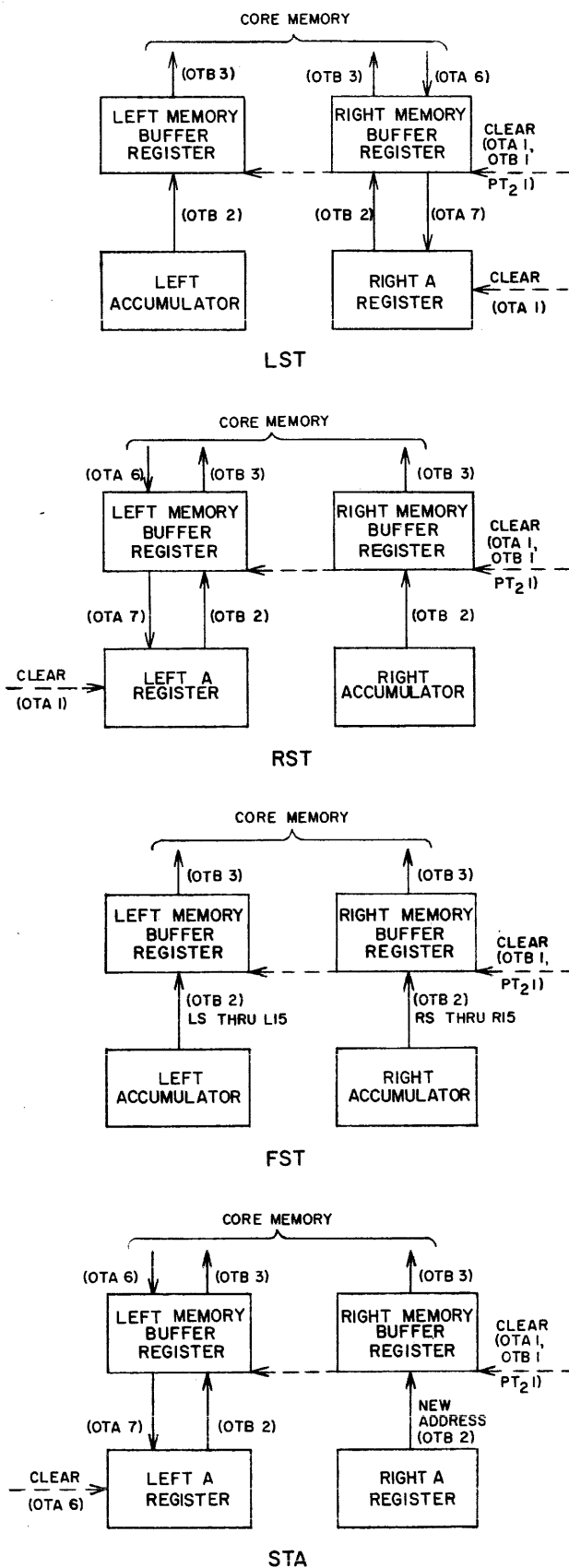


Figure 1-30. Store Class Execution

left and right accumulators, as a full word, in a selected register in core memory. (See fig. 1-30.) Since no reading from core memory is required, only one OT cycle is required for the execution of the *FST* instruction, reducing its execution time to 12 microseconds, in comparison to the 18 microseconds required by each other instruction of the store class. The remaining three instructions of the store class differ sufficiently from the typical instruction to require separate descriptions.

3.5.4 Different Variations

3.5.4.1 Add One

The *Add One (AOR)* instruction removes a full word from core memory, adds a 1 to the least significant bit position of the right half-word, and replaces the modified word in its original location in core memory. (See fig. 1-31.) The original word is read from core memory and placed in the A registers by OT A-7. The right accumulator has been cleared at OT A-6. Thus, when the carry 1 line to the right adder is pulsed, the right half-word is increased by 1 and placed in the right accumulator. (The inherent shift right and correction are not shown.) The OT B cycle returns the modified word to its core memory location. The *AOR* instruction can cause a right overflow. (Refer to 3.3.4.)

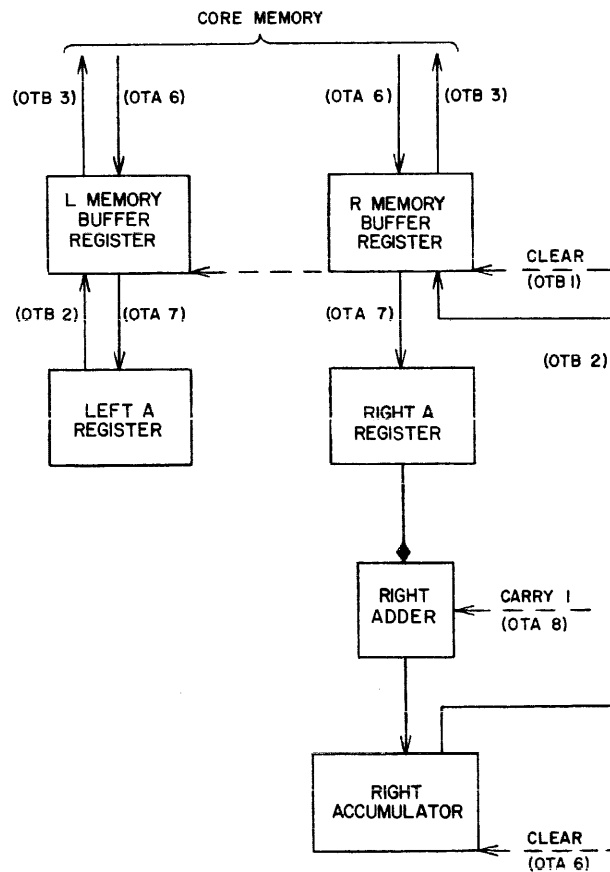


Figure 1-31. Add One (AOR) Execution

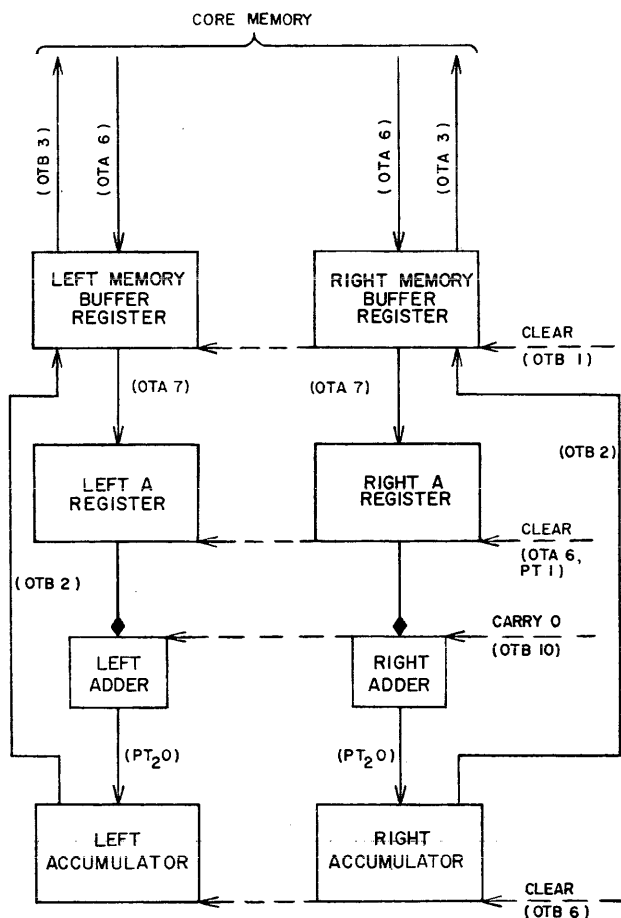


Figure 1-32. Exchange (ECH) Execution

3.5.4.2 Exchange

The *Exchange (ECH)* instruction interchanges the word held in the left and right accumulators with a word held in a selected core memory register. (See fig. 1-32.) During the OT A cycle, the word in core memory is transferred into the A registers. At OT B-2, the contents of the accumulators are transferred to the memory buffer registers to be placed in the core memory register just read. At OT B-6, the accumulators are cleared in preparation for the transfer of the word read from core memory into the accumulators as performed by a *CAD* instruction.

3.5.4.3 Deposit

The *Deposit (DEP)* instruction is unique within the store class in that it allows replacement of part of a word in core memory with the corresponding part in the accumulators on a bit-by-bit basis rather than by half-word or full word. The *DEP* instruction can cause replacement of only one bit of the word in core memory or of as much as the entire word. Before the *DEP* instruction can be given, the bits to be placed in the stored word must be in the accumulators and a control

word must be placed in the B registers. The contents of the control word determine which bits of the stored word are to be changed; each bit position in the B register containing a 1 causes the replacement of the corresponding bit in the stored word by that bit in the accumulator.

The *DEP* instruction is decoded in the same manner as are other instructions. (Refer to 3.2.2.) The address half of the instruction, which selects the word in core memory to be modified, may be changed by indexing prior to the modification operation. Although the operation is described only for the left half of the arithmetic element, corresponding actions are performed in the right half.

Execution of the *DEP* instruction begins at OT A-2 when the contents of the accumulator are complemented. (See fig. 1-33 and table 1-14.) At OT A-4, the control half-word is transferred from the B register to the A register. At OT A-5, the accumulator contents are logically multiplied by the contents of the A register. The logical multiply step clears those bits of the accumulator whose positions correspond to bit positions in the control half-word containing 0. In effect, the bits in the

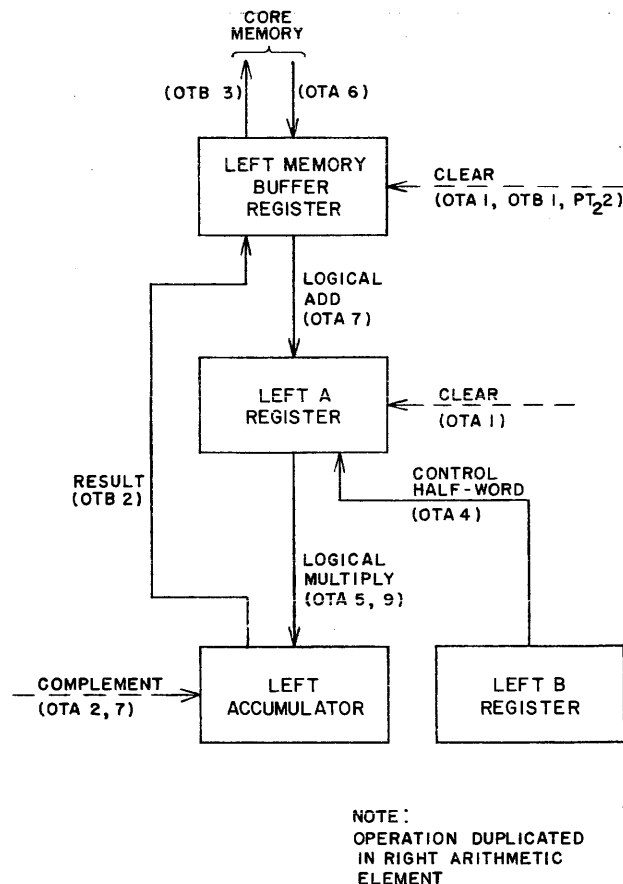


Figure 1-33. Deposit (DEP) Execution

TABLE 1–14. DEPOSIT INFORMATION CHANGES

TIME	ACCUMULATOR	A REGISTER	MEMORY BUFFER REGISTER
OT A 1	0.000 010 001 110 010	Clear	Clear
OT A 2	1.111 101 110 001 101	Unchanged	Unchanged
OT A 4	Unchanged	0.000 111 000 111 000	Unchanged
OT A 5	0.000 101 000 001 000	Unchanged	Unchanged
OT A 6	Unchanged	Unchanged	0.101 101 110 000 110
OT A 7	1.111 010 111 110 111	0.101 111 110 111 110	Unchanged
OT A 9	0.101 010 110 110 110	Unchanged	Unchanged
OT B 1	Unchanged	Unchanged	Clear
OT B 2	Unchanged	Unchanged	0.101 010 110 110 110

accumulator which are not to be placed in core memory are erased. At OT A-7, the accumulator is again complemented and the selected word from core memory is logically added to the control half-word in the A register. The logical add step sets to 1 those bits of the selected word which are to be replaced and leaves unchanged the remaining bits of the word brought from core memory. At OT A-9, the accumulator is again logically multiplied by the contents of the A register, leaving the new half-word in the accumulator. At OT B-2, this half-word is transferred to the memory buffer register to be written into core memory.

A summary of information concerning store class instructions is given in table 1–15.

3.6 SHIFT CLASS INSTRUCTIONS

3.6.1 Principles of Execution

The instructions within the shift class are transfer instructions which affect only the accumulator and B

registers. (Refer to 3.5.3, Chapter 2.) The bits within various combinations of these registers can be shifted left or right. The number of shifts performed is specified by the address half of the shift class instruction. To differentiate this use of the right half-word of an instruction as a numerical constant rather than as a core memory address, it is designated generically as *n*.

The basic operation performed in response to a shift class instruction is the simultaneous shift of each bit in a selected register one place to the right or to the left. The basic operation requires the transfer of a bit from each flip-flop to the next flip-flop within the same register. This basic operation is repeated as many times as necessary to accomplish the desired displacement of the bits being shifted.

Two basic variations exist within the shift class. Shift instructions proper affect all bits of the selected registers except the sign bit of the accumulator. Cycle instructions, on the other hand, affect all bits, including

TABLE 1–15. STORE CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES			EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC		OCTONARY		
<i>Full Store</i>	<i>FST</i>	<i>x</i>	324	12	Yes
<i>Left Store</i>	<i>LST</i>	<i>x</i>	330	18	Yes
<i>Right Store</i>	<i>RST</i>	<i>x</i>	334	18	Yes
<i>Store Address</i>	<i>STA</i>	<i>x</i>	340	18	Yes
<i>Add One</i>	<i>AOR (o)</i>	<i>x</i>	344	18	Yes
<i>Exchange</i>	<i>ECH</i>	<i>x</i>	350	18	Yes
<i>Deposit</i>	<i>DEP</i>	<i>x</i>	360	18	Yes

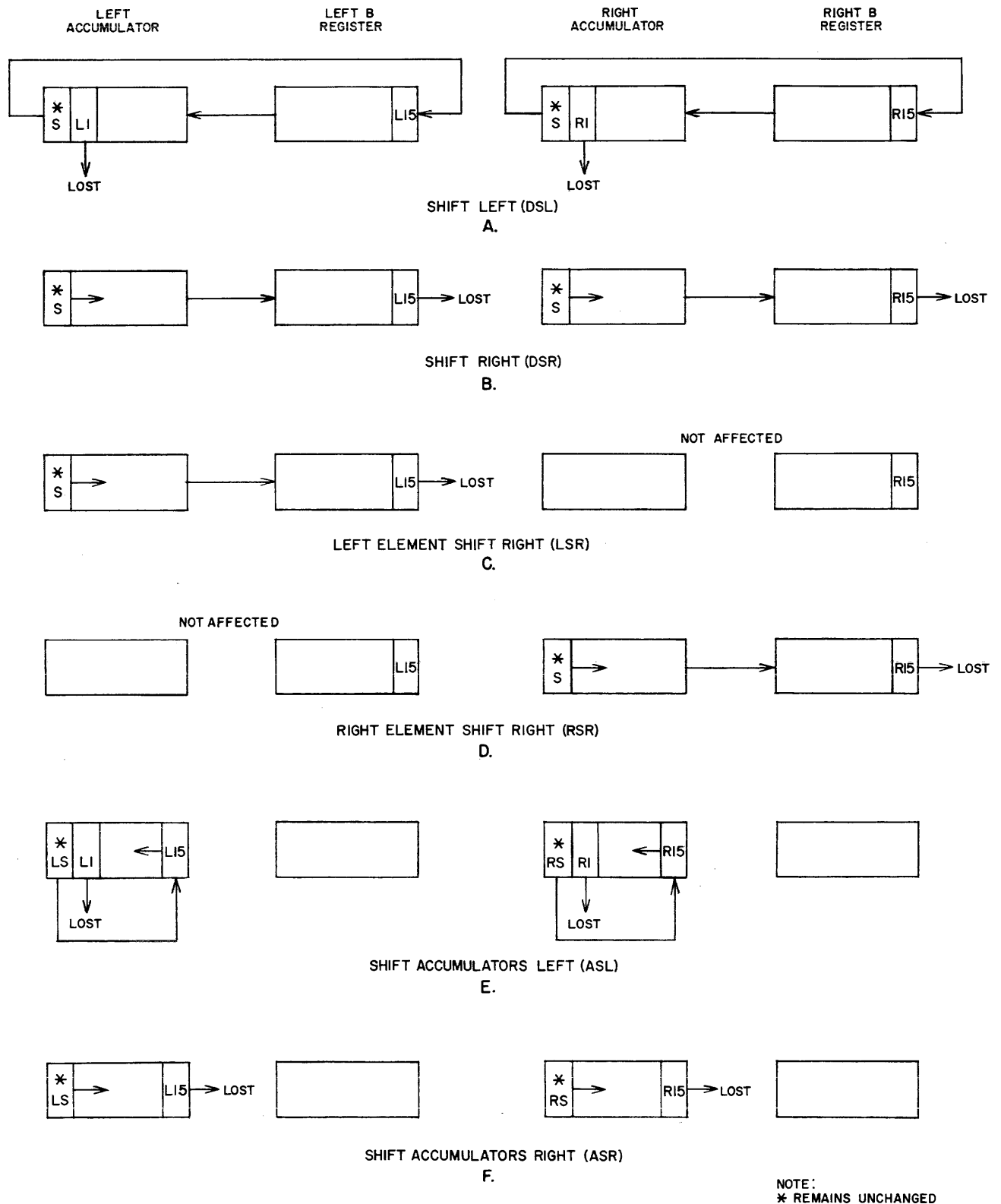


Figure 1-34. Shift Class Execution

the accumulator sign bit. Neither type of shift class instruction makes reference to core memory, and therefore none of the instructions in the shift class are indexable.

The execution time for a shift class instruction varies with the number of shifts called for by the instruction. Each basic shift operation (one place left or right) is initiated by 2-megacycle pulses rather than

instruction pulses (IP). Thus, successive shifts can be executed at $1/2$ -microsecond intervals. Since 2-megacycle operation can start at PT 11, up to 6 shifts may be executed within the minimum instruction cycle of 6 microseconds. Thus, for n shifts where $n = < 6$, execution time for a shift class instruction is 6 microseconds. Where $n > 6$, execution time is $6 + 0.5(n-4) \pm 0.5$ microseconds. A limit on the maximum number of shifts which may be performed is established by the step counter, a modulo 100_8 counter which counts out the required number of 2-megacycle pulses to execute the required number of shifts. Since the step counter is a modulo 64 counter, n is interpreted modulo 64. Therefore, a maximum of 63 or 77_8 shifts may be performed, and the maximum execution time is 35.5 ± 0.5 microseconds. (The 0.5-microsecond variation is caused by the variable time required for synchronization of 2-megacycle pulses.) If n is assigned a value greater than 77_8 in a shift class instruction, it is interpreted modulo 100_8 . Thus, for $n = 107_8$, only 7 shifts are executed. The execution time when $n = 107_8$ is the same as for $n = 7$ and therefore less than for $n = 77_8$.

3.6.2 Typical Instruction

The *Shift Left (DSL)* instruction is typical of the shift class. The *DSL* instruction combines the accumulator and B registers in each half of the arithmetic element into two 32-bit shifting registers. (See fig. 1–34.) With each shift, the content of the accumulator sign bit is duplicated in the bit 15 flip-flop of the B register, while the content of the accumulator bit 1 flip-flop is lost. The content of the accumulator sign bit flip-flop remains unchanged.

The *DSL* instruction is decoded in a manner similar to that used for all other instructions. As is done for

all instructions, bits R10 through R15 of the instruction word are placed in the step counter as well as in the address register. At PT 9, 2-megacycle operation is initiated. At PT 10, a pause is called for if n is greater than 6. When the step counter contents are reduced to 5, the arithmetic pause is ended, since the completion of the shifts will not interfere with the operations between PT₂ 0 and PT₂ 6 for termination of the instruction cycle. As can be seen from figure 1–34, if $n = > 31$ or 37_8 , then all significant bits of the accumulator and all bits of the B register are made identical to the sign bit of the accumulator.

3.6.3 Shift Variations

The variations of the *DSL* instruction are relatively close variations. The *Shift Right (DSR)* instruction shifts the combined accumulator-B register in each half of the arithmetic element to the right. (See fig. 1–34.) Again, for $n = > 37_8$, the accumulator sign is duplicated in all remaining bit positions of the accumulator-B register.

The *Left Element Shift Right (LSR)* instruction and the *Right Element Shift Right (RSR)* instructions are similar to the *DSR* instruction, with the exception that the *LSR* instruction affects only the left half of the arithmetic element while the *RSR* instruction affects only the right half. The *Shift Accumulators Left (ASL)* instruction and the *Shift Accumulators Right (ASR)* shift the contents of each bit position (except the sign bits) within the accumulator. For these two instructions, if $n = > 17_8$, all bits of the accumulator are made identical to the sign bit content.

3.6.4 Cycle Variations

Two instructions within the shift class are known as cycle instructions. They differ from the shift instructions

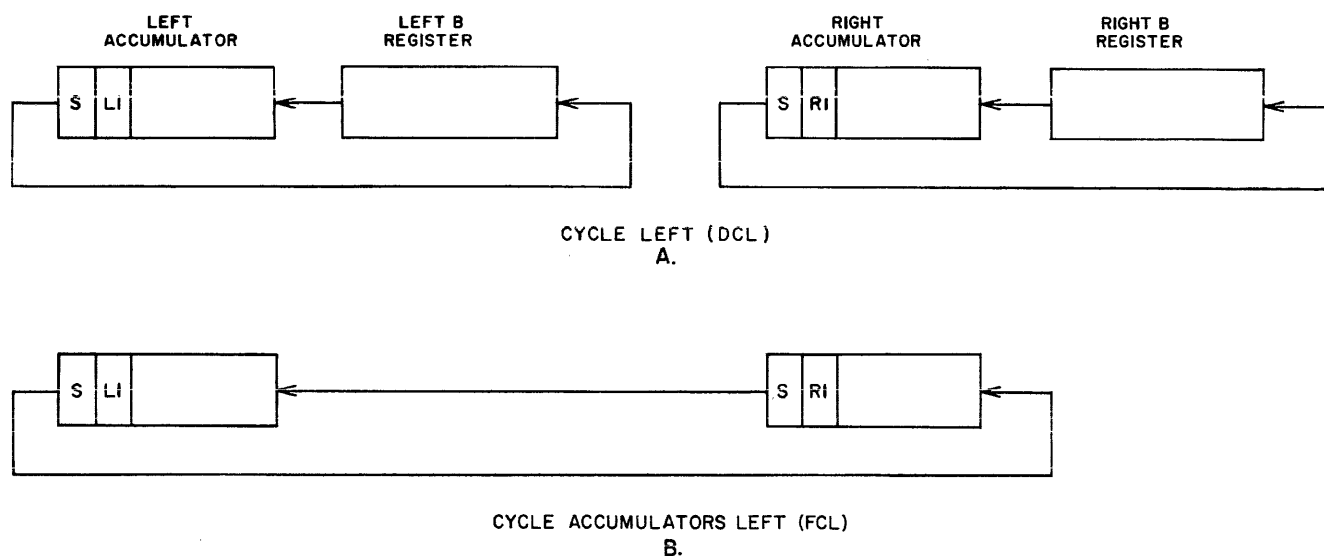


Figure 1–35. Cycle Variations Execution

just described in that they treat the sign bit of the accumulator register just as all other bits are treated.

The *Cycle Left (DCL)* instruction forms the accumulator and B register of each half of the arithmetic element into 32-bit cycling registers. (See fig. 1-35.) Unlike the *DSL* instruction, the *DCL* instruction cannot lose any bits but can move the accumulator sign bit. Where $n = 16$ or 20_8 , a *DCL n* instruction interchanges the contents of the accumulator and B registers. Where $n = 40_8$, the final result of the *DCL* instruction is equivalent to no shifting at all, since 32 shifts bring the bits back to their original positions. The effect of a *DCL* instruction where $n > 32$ is that of a *DCL* instruction calling for a number of shifts equal to the difference between the number of shifts specified by n minus 32. Thus, a *DCL 44₈* instruction takes longer to execute but is equivalent in effect to a *DCL 4₈* instruction.

The *Cycle Accumulators Left (FCL)* instruction differs from the *DCL* instruction in that it combines the two accumulator registers into a 32-bit cycling register without affecting the B registers. The *FCL* instruction can thus shift the positions of the bits within a full word. Where $n = 16$ or 20_8 , the contents of the left and right half-words are interchanged. If $n = 40_8$, the *FCL* instruction has no net effect on the contents of the accumulators. A summary of information concerning shift class instructions is given in table 1-16.

3.7 BRANCH CLASS INSTRUCTIONS

3.7.1 Principles of Execution

The instructions within the branch class allow modifications of the sequence in which a program is executed

in response to selected conditions resulting from Central Computer System operation. A branch class instruction can be used to prevent a meaningless operation; i.e., an operation using an intermediate result of a calculation in further calculation which would be meaningless if the actual intermediate result were used. For example, dividing a number into 0 can be avoided through the use of a branch class instruction.

A branch of program control is executed by a branch class instruction if the condition specified by the instruction is satisfied. The branch is performed as shown in figure 1-36. At PT 11, the address contained in the program counter is transferred to the right A register and the program counter is cleared. At PT₂ 0,

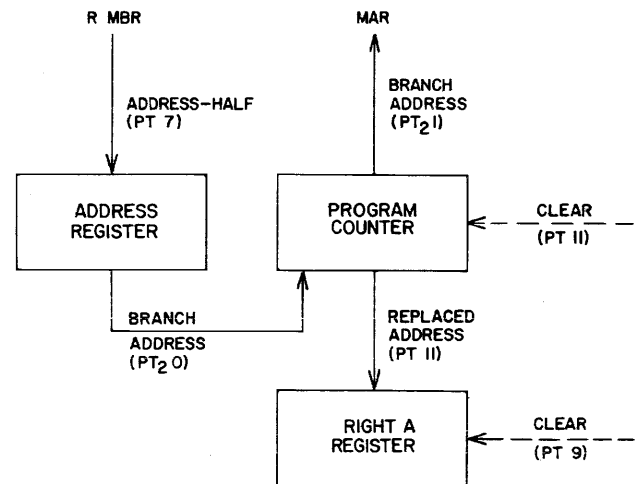


Figure 1-36. Branch Class Execution

TABLE 1-16. SHIFT CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES			EXECUTION	
	MNEMONIC		OCTONARY	(MICROSECONDS)	INDEXABLE
<i>Shift Left</i>	<i>DSL</i>	<i>n</i>	400	*	No
<i>Shift Right</i>	<i>DSR</i>	<i>n</i>	404	*	No
<i>Shift Accumulators Left</i>	<i>ASL</i>	<i>n</i>	420	*	No
<i>Shift Accumulators Right</i>	<i>ASR</i>	<i>n</i>	424	*	No
<i>Left Element Shift Right</i>	<i>LSR</i>	<i>n</i>	440	*	No
<i>Right Element Shift Right</i>	<i>RSR</i>	<i>n</i>	444	*	No
<i>Cycle Left</i>	<i>DCL</i>	<i>n</i>	460	*	No
<i>Cycle Accumulators Left</i>	<i>FCL</i>	<i>n</i>	470	*	No

*Execution time variable with n . Expressed decimally, when $n < 6$, execution time is 6 microseconds; when $n > 6$, execution time is $6 + 0.5(n - 4) \pm 0.5$ microseconds; n is interpreted modulo 64, making maximum execution time 35.5 ± 0.5 microseconds.

the address half of the branch class instruction is transferred from the address register to the program counter. Thus, if the address from which the branch class instruction was obtained is identified as x_1 while the address half of the instruction itself is identified as x_2 , then the branch of program control causes the program to execute the instruction at memory address x_2 rather than the instruction at $x_1 + 1$ that would normally follow the branch class instruction. If the condition specified by the branch instruction is not satisfied, then the instruction at memory address $x_1 + 1$ is executed in normal sequence immediately after the branch class instruction.

Execution of a branch class instruction may be modified by the **BRANCH NORMAL-BRANCH ZERO** switch on the duplex maintenance console when the Central Computer System is in test mode of operation. When this switch is in **BRANCH NORMAL** position, branches are executed as described. When it is in the **BRANCH ZERO** position and the condition for branch of a branch class instruction is satisfied, the command at PT_2 0 which transfers the contents of the address register to the cleared program counter is not generated unless the LS bit of the branch instruction being executed contains a 1. If the LS bit is 0, the program counter is left cleared, causing the next instruction to be taken from core memory location zero. If the LS bit of the branch instruction is 1, the instruction is executed normally; the position of the switch thus has no effect upon branch class instructions whose LS bit is 1. Bit LS therefore has a function only when the Central Computer System is in test mode and the branch control switch is in **BRANCH ZERO** position.

The memory address $x_1 + 1$ stored in the right A register when a branch of program is executed can be used to return to the program sequence that was interrupted by the branch. For example, if the program sequence starting at address x_2 includes seven instructions necessary to its execution, two more instructions can be added to return to the instruction at $x_1 + 1$ upon completion of the branch sequence. The last instruction of the branch sequence is stored at memory address $x_2 + 8$. The instruction at address x_2 can be *STA* $x_2 + 8$, which places $x_1 + 1$ in the address half of the instruction stored at $x_2 + 8$. If the instruction at $x_2 + 8$ is a branch class instruction which calls for an unconditional branch, the program will execute the instructions stored at the following addresses in the following order: $x_1, x_2, x_2 + 1 \dots x_2 + 8, x_1 + 1, x_1 + 2$, and so on. The sequence of instructions from x_2 through $x_2 + 8$ is called a closed routine, since upon its completion the program returns to the instruction sequence interrupted by that routine.

3.7.2 Typical Instruction

The *Branch on Minus (BFM)* instruction is typical of the branch class. The instruction is decoded in a manner similar to that used for all other instructions. At PT 9, the contents of the left and right accumulator sign flip-flops are sensed. If both flip-flops contain 1 (both accumulators are negative), the *BFM* branch condition is satisfied and the branch is executed. The *BFM* instruction is executed within one memory cycle, 6 microseconds, and is not indexable. It should be noted that negative zero is treated as a negative number by the *BFM* instruction.

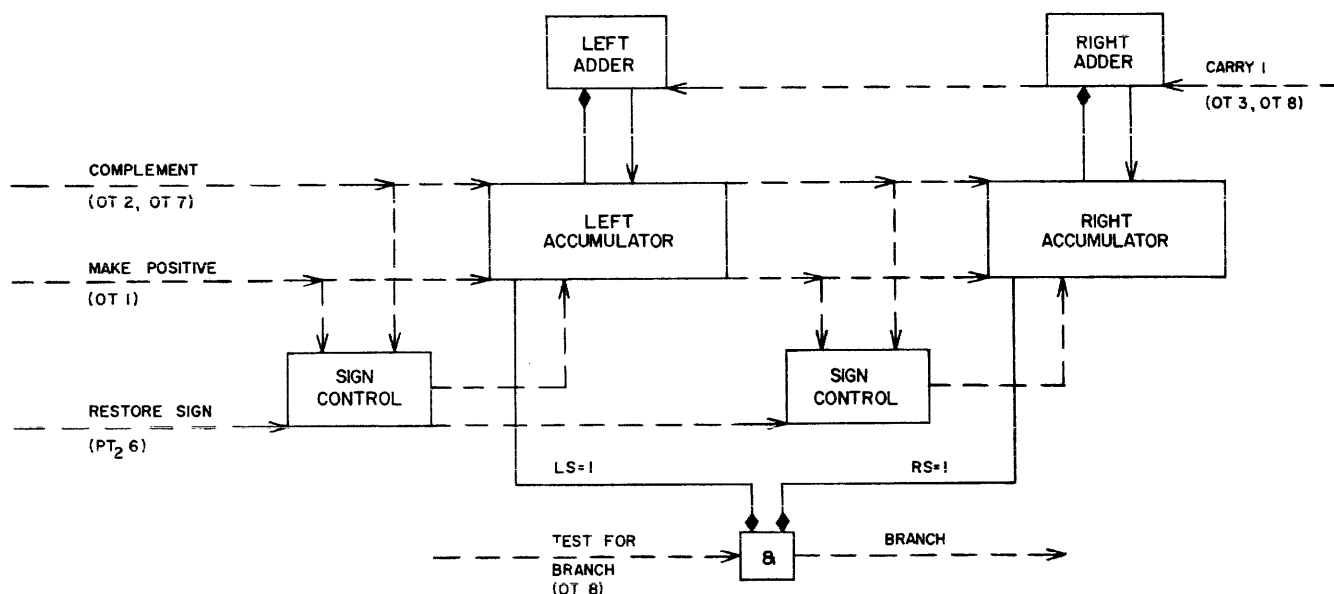


Figure 1-37. Branch on Zero (BFZ) Execution

3.7.3 Similar Variations

Only two of the other five instructions within the branch class have enough similarity to the *BFM* instruction to allow their discussions as variations. These two instructions, *Branch on Left Minus (BLM)* and *Branch on Right Minus (BRM)*, cause a branch of program control if the specified accumulator sign is 1; execution time is 6 microseconds.

3.7.4 Different Variations

3.7.4.1 Branch on Zero

The *Branch on Zero (BFZ)* instruction causes a branch of program control only if both accumulators contain zero. Since both positive and negative zero must be recognized as zero, the instruction must test for the presence of either positive or negative zero or a combination of these in the accumulators and then restore their original contents. The testing and restoration require more time than is available in a one-memory-cycle instruction cycle. Therefore, an OT cycle is included within the *BFZ* instruction cycle. (A word is read from core memory into the memory buffer registers during the OT cycle, but no use is made of this word, nor is the *BFZ* instruction indexable.)

Figure 1-37 illustrates the registers involved in execution of the *BFZ* instruction, and table 1-17 shows the results of this execution on the contents of one accumulator (abbreviated to five significant bits) when starting from each of the three possible contents of the accumulator, positive zero, negative zero, or not zero.

TABLE 1-17. BRANCH ON ZERO TEST OPERATIONS

TIME	POSITIVE ZERO	NEGATIVE ZERO	NOT ZERO
Initial	0.0000	1.1111	1.0010
OT 1	0.0000	0.0000	0.1101
OT 2	1.1111	1.1111	1.0010
OT 3	1←0.0000	1←0.0000	1.0011
OT 7	1.1111	1.1111	0.1100
OT 8	1←0.0000	1←0.0000	0.1101
PT ₂ 6	0.0000	1.1111	1.0010

The sequence of operations necessary to execute the *BFZ* instruction begins with the complementing of either or both accumulators if they are negative at OT 1. Thus, the contents of both accumulators are now known to be positive. (If either accumulator had been negative, the sign control flip-flop would have been complemented

along with the accumulator to maintain a record of its original sign.) At OT 2, both accumulators and the sign control flip-flops are unconditionally complemented. At OT 3, a 1 is added to both accumulators (carry 1 line is pulsed with A registers cleared), ignoring any end-carry generated. At OT 7 (after the correctional shift left which is not shown in figure 1-37), the accumulators and sign control flip-flops are again complemented. After this step, the sign bit flip-flop of each accumulator contains a 1 if that accumulator had contained either positive or negative zero at the start. Thus, at OT 8, the condition for branch is tested and, if it satisfied, a branch of program control is called for.

The restoration of the accumulator contents begins at OT 8 with a subtraction of 1 from the accumulator contents (carry 1 while accumulators are complemented with respect to previous addition of 1); any end-carry generated is ignored. The branch of program control is executed if the conditioning for branch is satisfied at OT 8, with the exception that the branch operation proper begins at OT 11 rather than at PT 11. (See fig. 1-36.) The restoration of accumulator contents is completed at PT₂ 6 with the complementing of each accumulator if its sign control flip-flop contains a 1.

3.7.4.2 Sense

The *Sense (BSN)* instruction provides for a branch of program control if a selected condition is satisfied.

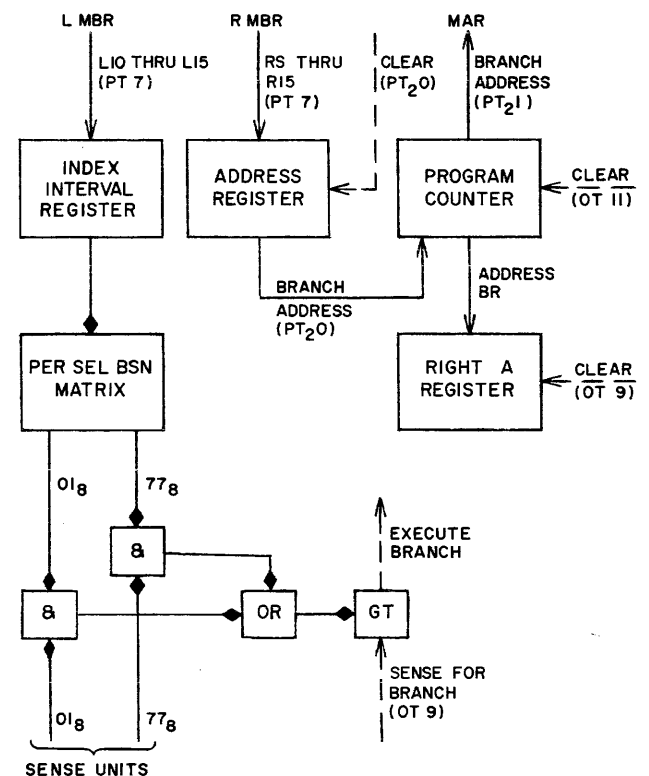


Figure 1-38. Sense (BSN) Execution

The index interval (L10 through L15) of the *BSN* instruction selects the condition which permits or prevents the execution of the branch. This use of the index interval is denoted by the symbol (μ). Each condition which may be sensed for execution of a branch is indicated by a sense unit. If the sense unit selected by the index interval of the *BSN* instruction is on, a branch of program control is executed.

The execution of a *BSN* (μ) instruction requires two memory cycles to provide time for the establishment of selection levels. Thus, as the case for a *BFZ* instruction, an OT cycle is executed with its accompanying memory cycle, although no net effect on core memory or the memory buffer registers results. Therefore, the *BSN* (μ) instruction is not indexable.

The execution of the *BSN* (μ) instruction requires the decoding of its index interval. (See fig. 1-38.) This is performed by the PER SEL *BSN* matrix, conditioning one output line as selected by the contents of the index interval. (Although only two outputs are shown for the PER SEL *BSN* matrix, there are 63 outputs separately selectable by the index interval, with values ranging

from 00₈ through 77₈.) The index interval register is loaded to begin decoding at PT 7. The selection line from the matrix and the condition indicating line from the corresponding sense unit are applied to an AND circuit whose output conditions a gate which is sensed at OT 9. If the selected sense unit is on, the gate is conditioned and the command to execute a branch of program control is generated. Thereafter, the branch is executed as shown in figure 1-38. If the selected sense unit were not on, no branch would result, since the command to execute the branch would not be generated and the next instruction in the program sequence would be executed.

As an example of a *BSN* (μ) instruction, consider the *Sense* (left overflow on) instruction. The index interval of the instruction contains 12₈, specifying the left overflow sense unit. If an overflow has occurred in the left arithmetic element prior to the execution of the *BSN* (12₈) instruction, the left overflow sense unit is on. The execution of the *BSN* (12₈) instruction causes a branch and clears the left overflow sense unit. The sense units are listed in table 1-18.

TABLE 1-18. SENSE UNIT SELECTION CODES

SENSE UNIT	CONDITION FOR BRANCH	CODE INDEX INTERVAL	UNIT OFF BSN TURNS
Condition lights 1-4	On	01-04	Yes
Selected tape unit	Not prepared	10	No
Selected IO unit	Not ready	11	No
Left overflow	On	12	Yes
Right overflow	On	13	Yes
IO interlock	On	14	No
Memory parity error	On	15	Yes
Drum parity error	On	16	Yes
Tape parity error	On	17	Yes
Marginal checking excursion	On	20	No
Sense switches 1-4	On	21-24	No
Duplex switch	Active	30	No
Printer sense 1,2	Energized	31,32	No
Output alarm	On	33	Yes
GFI range signal	On	34	Yes
SD camera	Taking picture	35	No
Display System	Displaying track data	37	No

TABLE 1-18. SENSE UNIT SELECTION CODES (cont'd)

SENSE UNIT	CONDITION FOR BRANCH	CODE INDEX INTERVAL	UNIT OFF BSN TURNS
Other Computer alarms 1,2	On	41,42	Yes
Other Computer intercommunication 1-4	On	43-46	Yes
GFI north or azimuth	On	47	Yes
Output alarms:			
Nonsearch compare	On	50	Yes
Drum parity error	On	51	Yes
Illegal selection or register address	On	52	Yes
Defective transmission	On	53	Yes

3.7.4.3 Branch and Index

The *Branch and Index (BPX)* instruction provides for a conditional branch of program control with the index indicator (L1 through L3) of the instruction word designating the condition upon which the branch is dependent. In addition, the *BPX* instructions can reduce the contents of an index register by the amount specified by the index interval (L10 through L15) of the instruction word. This use of the index interval is designated by the symbol (*s*). Through the use of the *BPX (s)* instruction, a series of indexable instructions can be repeated a fixed number of times to process a series of similar information items by indexing the iterated instructions differently for each iteration. A detailed discussion of the *BPX (s)* application is provided in Section 5.

The *BPX (s)* instruction causes a conditional branch only if the index indicator specifies an index register other than the right accumulator. (Refer to 3.2.3) Thus, if the index indicator is 0₈ or 3₈, (*s*) is meaningless and the branch is unconditionally executed. However, if one of the index registers is selected, the branch is executed and (*s*) is subtracted from the index register only if the sign of the selected index register is 0; i.e., the index register contains a positive number. If the contents of the selected index register are negative (sign = 1), no branch results.

Execution of the *BPX (s)* instruction is accomplished in one memory cycle. (See fig. 1-39.) The instruction word is transferred from the memory buffer registers at PT 7 and decoded as are all other instructions. If the index indicator is 0₈ or 3₈, a branch is executed unconditionally. (Refer to 3.7.1.) The condition for branch is satisfied at PT 9, and the branch is executed. If the index indicator selects one of the index

registers, the condition for branch is satisfied only if the sign of the selected index register = 0.

Although the sensing of index registers 2 and 4 is not shown in figure 1-39, the sensing is identical to that shown for registers 1 and 5. Assuming an index indicator of 1₈, if index register 1 contains a positive number, a branch of program control is called for at PT 9 and, in addition, the contents of index register 1 must be reduced by the amount specified by (*s*). The branch is executed as described in 3.7.1. The reduction of the index register contents starts with the clearing of the address register at PT₂ 0. At PT₂ 1, the complement of (*s*) is placed in the cleared address register. At PT₂ 2, the contents of index register 1 are added to the address register and the result appears in the address register. At PT₂ 4, the selected index register is cleared and, at PT₂ 6, its new contents are transferred from the address register to the index register. The net effect of these steps is the subtraction of the index interval from the selected index register. An index interval of 1 subtracted from an index register containing 1 leaves negative zero in the index register. A summary of information concerning branch class instructions is given in table 1-19.

3.8 IO CLASS INSTRUCTIONS

3.8.1 General

IO class instructions perform the preparatory steps necessary for the initiation of an IO operation. These instructions select an IO unit for operation and set the initial conditions in those registers that control each IO operation. (Refer to 3.1.4.4.) Since the execution of an IO class instruction while an IO operation is in progress will disrupt that IO operation, execution of an IO class instruction is conditional upon the state of the

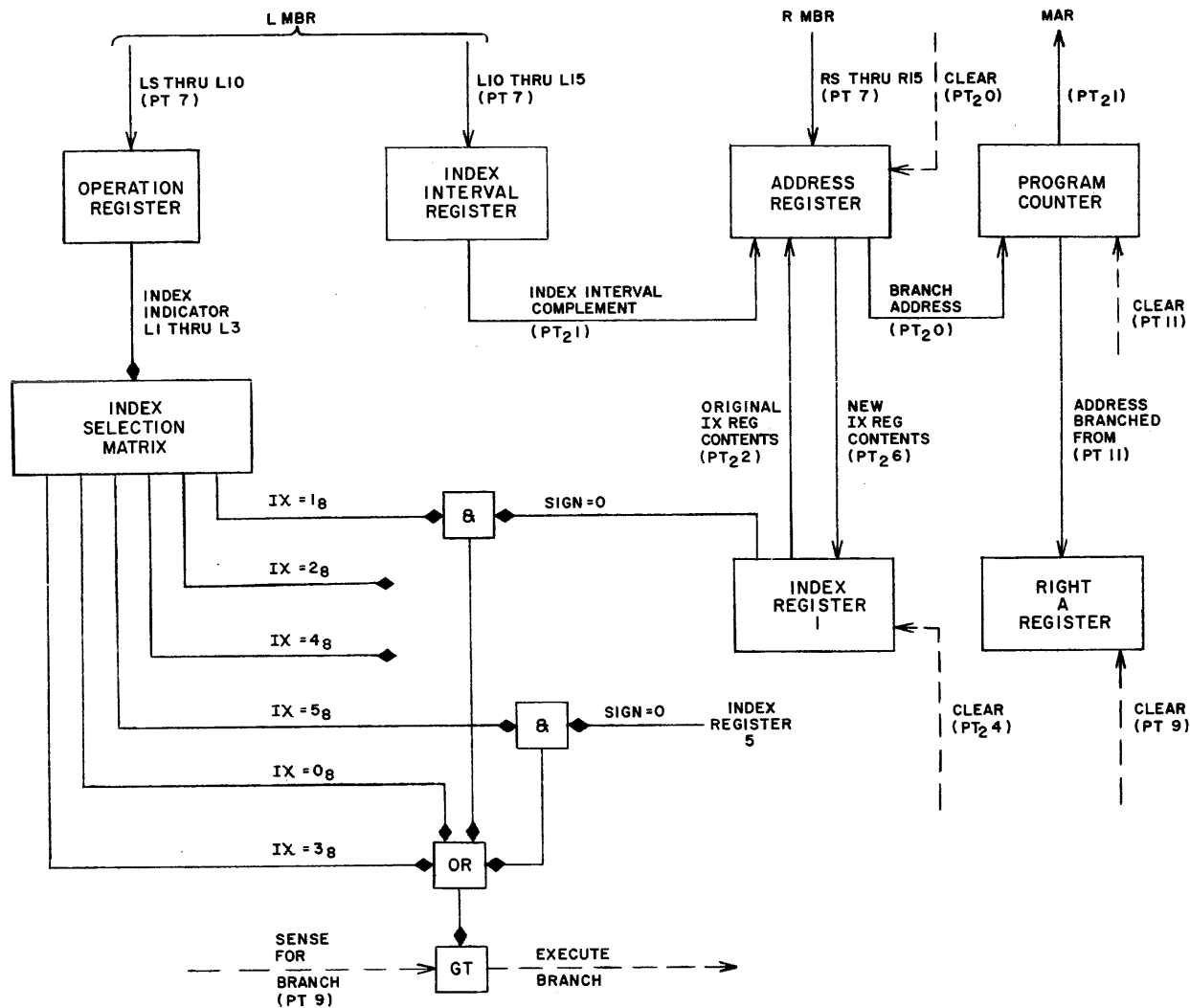


Figure 1-39. Branch and Index (BPX) Execution

TABLE 1-19. BRANCH CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES		EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC	OCTONARY		
Branch and Index	<i>Ix</i> BPX (<i>s</i>) <i>x</i>	51—	6	No
Sense	— BSN (<i>u</i>) <i>x</i>	52—	12	No
Branch on Zero	— BFZ — <i>x</i>	540	12	No
Branch on Minus	— BFM — <i>x</i>	544	6	No
Branch on Left Minus	— BLM — <i>x</i>	550	6	No
Branch on Right Minus	— BRM — <i>x</i>	554	6	No

IO interlock. If the IO interlock is on, indicating that an IO operation is in progress, instructions of the IO class cannot be executed.

All instructions of the IO class are indexable. (Refer to 3.2.3.) However, in the case of one instruction in the

class, indexing is meaningless, since the address half of that instruction is not used. Execution time for three instructions in the IO class is 6 microseconds; for the other two, execution time is 12 microseconds. Each instruction of the IO class is discussed separately in the following text.

3.8.2 Variations

The registers associated with the execution of IO class instructions are shown in figure 1-40. The *Load IO Address Counter (LDC)* instruction prepares the IO address counter for an IO operation by loading into this register the address of the first core memory register to be affected by an IO operation. The address half of the

LDC instruction can be modified by indexing. Execution time is 6 microseconds.

The *Select (SEL)* instruction chooses an IO unit other than a drum field for an IO operation. The index interval of the *SEL* instruction, designated as (*u*), is decoded by the PERSELBSN matrix to select the IO unit. A given *SEL* (*u*) instruction deselects all other IO

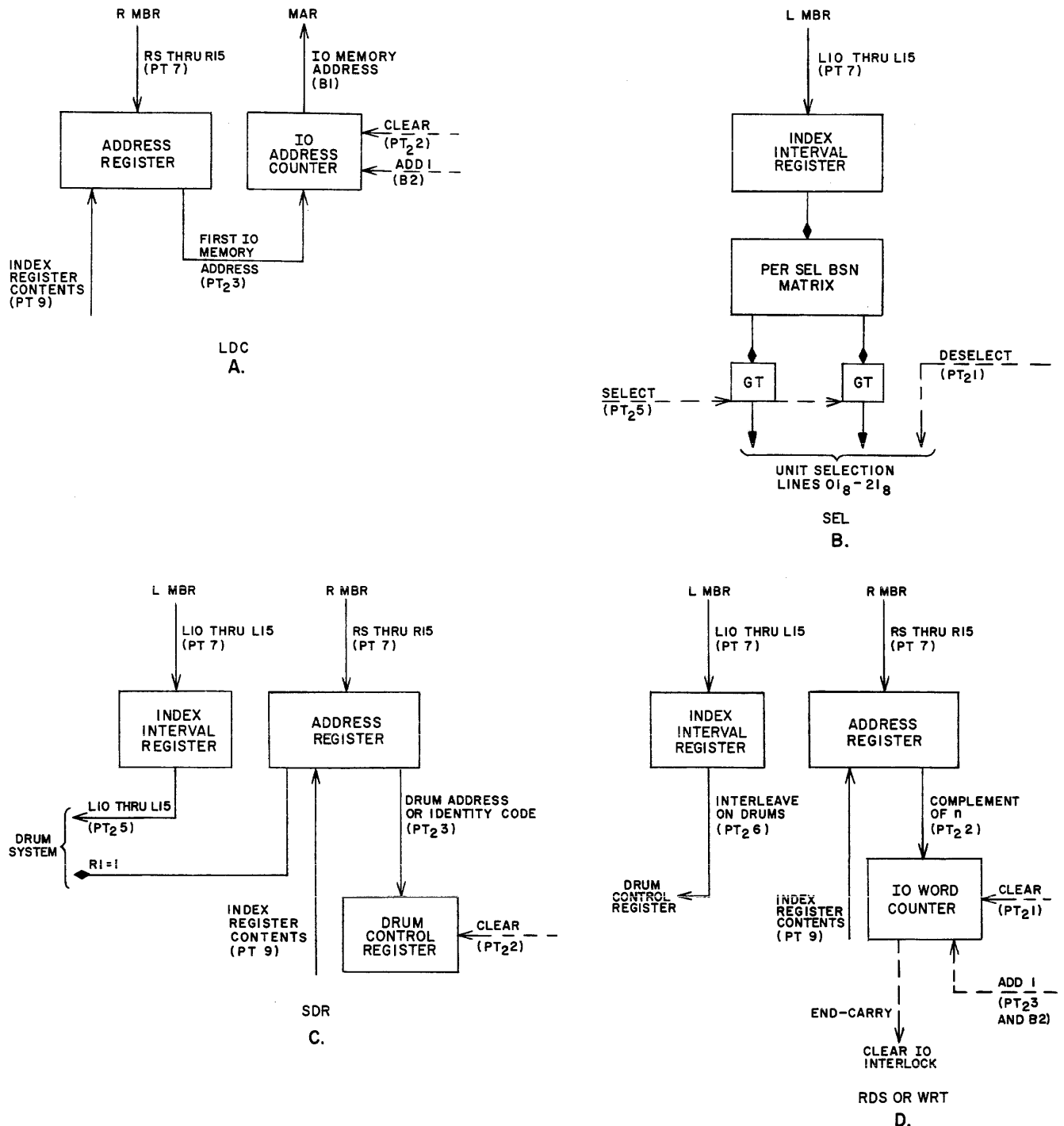


Figure 1-40. IO Class Execution

units when the unit designated by (*u*) is selected. The IO unit selected remains so until a different unit is selected. The address half of the *SEL (u)* instruction is meaningless, thus rendering indexing also meaningless. Execution time for the *SEL (u)* instruction is 12 microseconds. The IO units selectable by the *SEL (u)* instruction are listed in table 1-20. Several of them are described in Section 4.

TABLE 1-20. SELECT UNIT CODES

IO UNIT	SELECTION CODE*	OPERATION
	(OCTONARY L10-L15)	
Card Reader	01	Read
Card Punch	02	Write
Printer	03	Write
IO register, cleared	04	Read
Manual input matrix	06	Read
Warning Light System	10	Write
Magnetic tapes 1-6	11-16	Read, write
Burst time Counters	21	Read

*Codes not listed are spares, not used at present.

The *Select Drum (SDR)* instruction chooses a drum field for use in an IO operation. The index interval of the *SDR* instruction, designated as (*u*), selects a drum field within the group of drum fields selected by bit R1 of the address half of the *SDR (u)* instruction. If R1 = 0, the index interval selects a field within the main drum (MD) group (AM A, AM B, LOG, MIXD, TD, or RD). If R1 = 1, the index interval selects a field within the auxiliary drum (AXD) group (AM C through AM H). Unlike the *SEL (u)* instructions, the PerSelBsn matrix does not decode the index interval of

the *SDR (u)* instruction. Instead, the index interval of the *SDR* instruction is supplied to the Drum System for decoding therein. Those bits of the address half of the *SDR (u)* instruction other than R1 may be meaningless, may specify an address on a drum field (bits R4 through R15), or may contain an identity code. Indexing of the *SDR (u)* instruction is possible, thus changing the contents of the address half-word. Execution time for the instruction is 12 microseconds. (The uses of the *SDR* instruction are discussed in Chapter 4.)

The *Read (RDS)* and *Write (WRT)* instructions are basically quite similar in their execution. The execution of either instruction turns on the IO interlock and starts the word transfers which make up an IO operation. Further, the address half of the instruction word specifies the number of words to be transferred. Thus, an *RDS n* instruction calls for an IO operation in which *n* words are read from a previously selected IO unit into core memory. Conversely, a *WRT n* instruction calls for an IO operation in which *n* words are written onto the storage medium of some IO unit from core memory. If the IO unit selected is a drum field which is written and read by address, the index interval of either an *RDS n* or a *WRT n* instruction may specify an interleaving pattern on the drum field. (Refer to Chapter 4.) This use of the index interval is designated (*i*).

As shown in figure 1-40, the *RDS n* or *WRT n* instruction causes the loading of the complement of *n* into the IO word counter. At PT₂ 3, a 1 is added to the IO word counter, thus converting the contents of the counter into a number which is the complement of *n* - 1. The contents of the IO word counter are thus reduced before any word transfers occur to allow the generation of a dynamic end-carry from the counter at the transfer of the *n*th word. This end-carry is used to clear the IO interlock and terminate the IO operation. Execution time for either of these instructions is 6 microseconds. A summary of information concerning IO class instructions is given in table 1-21.

TABLE 1-21. IO CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES		EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC	OCTONARY		
<i>Load IO Address Counter</i>	<i>LDC x</i>	600	6	Yes
<i>Select Drum</i>	<i>SDR (u) r*</i>	61—	12	Yes
<i>Select</i>	<i>SEL (u) —</i>	62—	12	Meaningless
<i>Read</i>	<i>RDS (i) n</i>	670	6+	Yes
<i>Write</i>	<i>WRT (i) n</i>	674	6+	Yes

*Significance dependent upon mode of operation chosen by (*u*).

3.9 RESET CLASS INSTRUCTIONS

3.9.1 General

The reset class of instructions allow the setting of an index register to a given value or the examination of the contents of an index register. None of the instructions within the reset class can utilize the right accumulator as an index register. Reset class instructions are not themselves indexable. Execution time for each instruction within the class is 6 microseconds.

3.9.2 Variations

The registers involved in execution of reset class instructions are shown in figure 1-41. The *Reset Index Register (XIN)* instruction replaces the right half of the instruction word in the index register selected by the index indicator of the instruction word. Thus, the *XIN n* instruction sets the selected index register to *n*.

The *Reset Index Register from Right Accumulator (XAC)* sets the selected index register to the number contained in the right accumulator. The address half of this instruction is meaningless. The *Add Index Register (ADX)* instruction adds the contents of the selected index register to the address half of the instruction and then transfers the result of the addition to the right A register. It may be seen that if the address half of the *ADX* instruction is 0, the contents of the selected index register are placed unchanged in the right A register. A summary of information concerning the reset class instructions is given in table 1-22.

3.10 MISCELLANEOUS CLASS INSTRUCTIONS

3.10.1 General

The instructions within the miscellaneous class are those which do not fall into any consistent class and are so diverse that no general principles of execution can be given. Instead, each instruction of this class is discussed separately.

3.10.2 Program Stop

The *Program Stop (HLT)* instruction, upon execution, stops all internal operations but allows any IO operation in progress to be completed before stopping the Central Computer System. The *HLT* instruction stops the Central Computer System by halting the instruction pulses which generate commands. The binary coding for the *HLT* instruction is an instruction word containing all zeroes in the operation code part. Thus, an empty register contains an *HLT* instruction, and, if read to obtain an instruction, will cause the Central Computer System to stop.

3.10.3 Extract

The *Extract (ETR)* instruction changes the contents of the accumulator registers in accordance with a control word read from core memory. Each bit position of the control word which contains a 0 clears the cor-

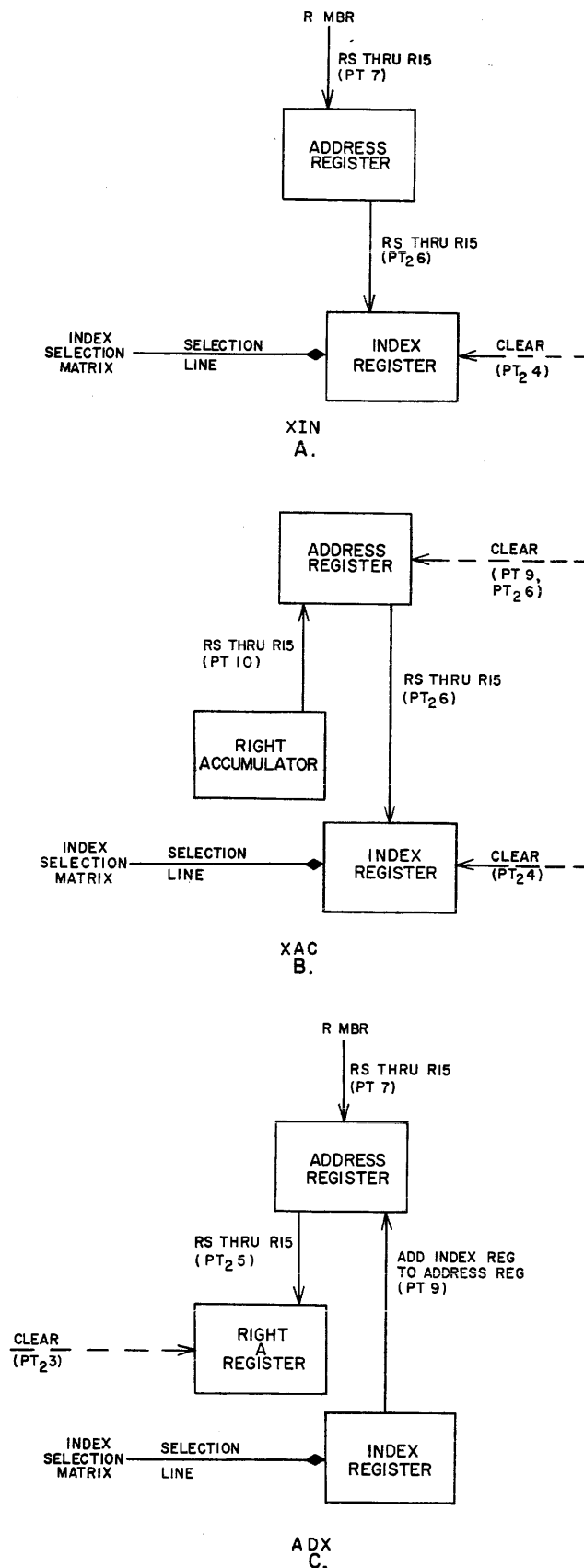


Figure 1-41. Reset Class Execution

TABLE 1–22. RESET CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES		EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC	OCTONARY		
Reset Index Register	<i>Ix</i> XIN <i>n</i>	754	6	No
Reset Index Register from Right Accumulator	<i>Ix</i> XAC –	764	6	No
Add Index Register	<i>Ix</i> ADX <i>n</i>	770	6	No

responding bit position in the accumulator. The *ETR* instruction is decoded in a manner similar to that used for all other instructions. The address half of the instruction word may be modified by indexing. At OT 7, the control word is transferred from the memory buffer registers to the A registers. (See fig. 1–42.) At OT 9,

the remaining portions of the word. Execution time is 12 microseconds.

3.10.4 Operate

The *Operate (PER)* instruction is used to initiate or execute a wide variety of miscellaneous actions. The choice is made possible by using the index interval of the instruction word to select the desired action. The index of the *PER (u)* instruction is decoded by the *PerSelBsn* matrix, which allows the transmission of a pulse to a unit selected by the index interval. The function of the pulse is dependent upon the unit selected. Execution time for *PER (u)* is 12 microseconds. The various uses of the *PER* instruction are listed in table 1–23.

3.10.5 Clear and Subtract Word Counter

The *Clear and Subtract Word Counter (CSW)* instruction transfers the contents of the IO word counter into the right accumulator register without changing the contents of the word counter or of any other register except the right accumulator. (See fig. 1–43.) The *CSW* instruction allows the program to determine the number of words yet to be transferred by an IO operation. The number placed in the right accumulator is the 1's complement of one less than the number of words remaining to be transferred. Execution time for the *CSW* instruction is 6 microseconds.

3.10.6 Shift Left and Round

The *Shift Left and Round (SLR)* instruction provides the means for scaling a small number up in magnitude and for rounding off that number to 15 magni-

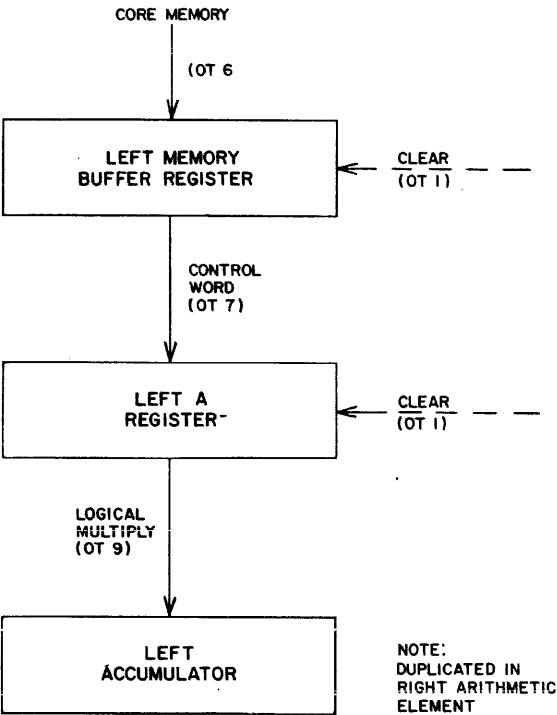


Figure 1–42. Extract (ETR) Execution

the contents of the accumulators are logically multiplied by the control word in the A registers. The logical multiplication sets to 0 those bit positions in the accumulators whose corresponding bits in the control word are 0, leaving unchanged those accumulator bit positions which correspond to bits containing 1 in the control word. It is worthwhile for the reader to compare this action with that of *Deposit (DEP)* instruction. (Refer to 3.5.4.3) The *ETR* instruction can be used to clear unnecessary portions of a word in the accumulators without affecting

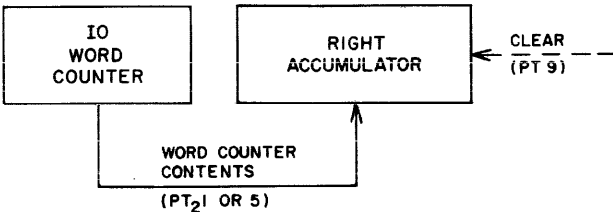


Figure 1–43. Clear and Subtract Word Counter (CSW) Execution

TABLE 1-23. OPERATE UNIT SELECTION CODES

OPERATE UNIT	ACTION	INDEX INTERVAL CODE
Condition lights 1-4	Turn on	01-04
Intercommunication indicators 1-4	Turn on in other computer	10-13
Area discriminator	Turn on for cycle beginning with radar data	20
Marginal checking:	Start excursion	21
	Stop duplex excursion	22
	Stop simplex excursion	23
IO interlock	Clear	27
SD camera:	Start mode 1	31
	Start mode 2	32
Digital display	Start at slot 0, 1st section	35
	Start at slot 107, 2nd section	36
Printer operate hubs 1-10	Energize	51-62
Input System testing:	Start LRI and XTL pattern generator; start GFI continuous pattern generator	63
	Connect GFI pattern generator	64
	Start GFI azimuth	65
	Start GFI target	66
Selected tape unit:	Set prepared	67
	Backspace	70
	Rewind	71
	Write end-of-file	72
Card punch	Punch columns 17-32 in columns 1-16 on card	73
	Gang punch 1-16	74
IO address counter	Lock at current address until IO interlock is cleared	75
Scan counter:	Set to 0	76
	Step by 1	77

tude bits if it is represented with excess precision. The *SLR* instruction shifts the bits in the combined accumulator-B register to the left by the number of positions specified by the address half of the instruction, then rounds off the half-word in each accumulator register in accordance with the contents of the B register sign flip-flop (the 16th and least significant bit). The shifting operation called for by the *SLR* instruction is identical with the shift executed in response to a *DSL n* instruction. (Refer to 3.6.2.) The address half of the *SLR* instruction, designated as *n*, is interpreted modulo 64.

If $n \geq 31$, all significant bits in the accumulator register are replaced by bits that agree with the accumulator sign bit. The result of the round-off is dependent upon the sign of the number and the content of the 16th significant bit. If the number is positive, a 1 in the 16th bit causes the addition of a 1 to the 15th bit. If the number is negative, a 0 in the 16th bit causes the subtraction of a 1 from the 15th bit. Since the *SLR* instruction can cause an overflow, the index interval of the instruction can cause a response to overflow. (Refer to 3.3.4.)

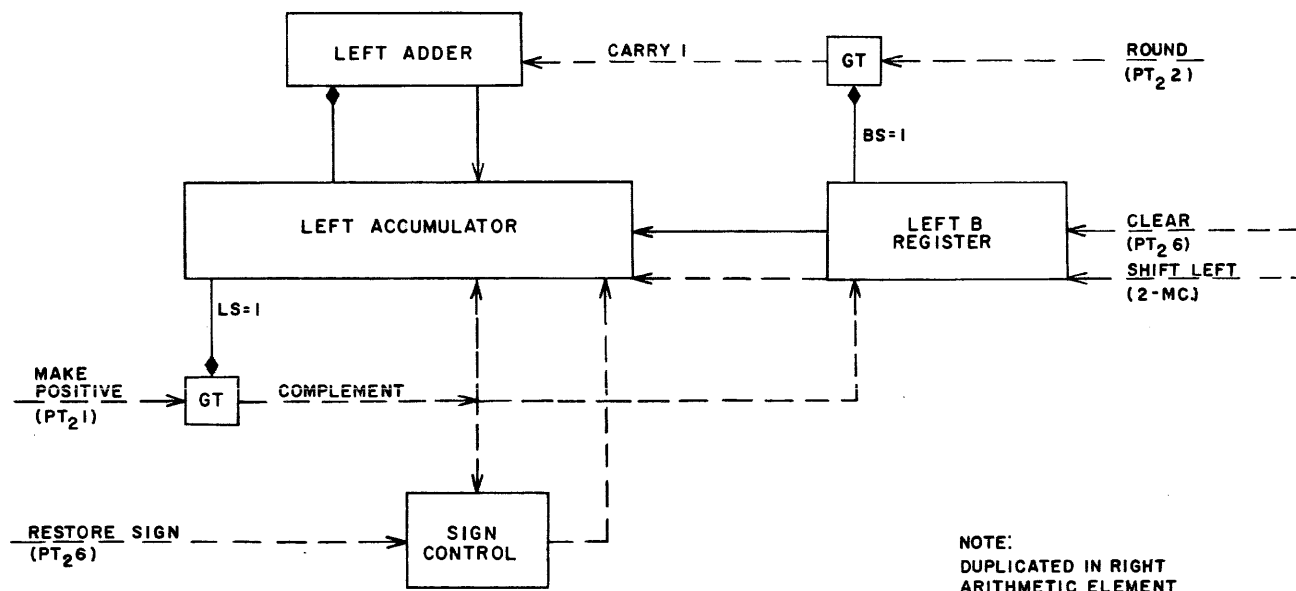


Figure 1-44. Shift Left and Round (SLR) Execution

The *SLR (o) n* instruction is decoded in a manner similar to that used for all other instructions. The shifts called for by *n* are executed at a 2-megacycle rate. Upon completion of all shifts, the round-off step begins. (See fig. 1-44.) At $PT_2 1$, the combined accumulator-B register is complemented if negative with its original sign stored in the sign control flip-flop. At $PT_2 2$, the carry 1 line to the adder is pulsed if the B sign flip-flop contains 1. Since the A register is cleared before this step, pulsing the carry 1 line results in the addition of a 1 to the bit 15 position of the accumulator. At $PT_2 6$, the accumulator sign is restored and the B register is cleared. Execution time for the *SLR (o) n* instruction is dependent upon the value of *n*. If $n \geq 2$, execution time is 6 microseconds. If $n \leq 2$, execution time is 6 microseconds. If $n > 2$, execution time is $6 + 0.5 (n + 1) \pm 0.5$ microseconds.

As an example of the use of the *SLR (o) n* instruction, consider the result of a *DVD* instruction. The sign bit and a 15-bit remainder are held in the accumulator register while the quotient, unsigned and expressed in 16 significant bits, is held in the B register. Execution of the instruction *SLR 17₈* shifts the most significant bit of the quotient to the bit 1 position in the accumulator and utilizes the 16th quotient bit in rounding off the quotient to 15 significant bits.

3.10.7 Load B Registers

The *Load B Registers (LDB)* instruction transfers a word from the core memory location specified by the address half of the instruction into the B registers. The instruction is indexable and requires 12 microseconds for execution. A summary of miscellaneous class instructions is given in table 1-24.

TABLE 1-24. MISCELLANEOUS CLASS INSTRUCTIONS

INSTRUCTION NAME	CODES			EXECUTION (MICROSECONDS)	INDEXABLE
	MNEMONIC		OCTONARY		
<i>Program Stop</i>	<i>HLT</i>	—	000	6	No
<i>Extract</i>	<i>ETR</i>	<i>x</i>	004	12	Yes
<i>Operate</i>	<i>PER (u)</i>	—	01—	12	No
<i>Clear and Subtract Word Counter</i>	<i>CSW</i>	—	020	6	No
<i>Shift Left and Round</i>	<i>SLR (o) n</i>		024	*	No
<i>Load B Registers</i>	<i>LDB</i>	<i>x</i>	030	12	Yes

*Execution time is variable with *n*; if $n \leq 2$, time is 6 microseconds; if $n > 2$, time is $6 + 0.5 (n + 1) \pm 0.5$ microseconds.

SECTION 4

INCLUDED IO UNITS

4.1 INTRODUCTION

There are several IO units logically included within and connected to the Central Computer System. These units include:

- Card reader (IBM type 713)
- Card punch (IBM type 723)
- Line printer (IBM type 718)
- Tape element
- IO register (used as an IO unit)

The first three units listed, generally known as card machines, are used primarily for the initial preparation, insertion, and testing of programs. These card machines provide unlimited memory capacity at the expense of random access time, which is very long for all three machines.

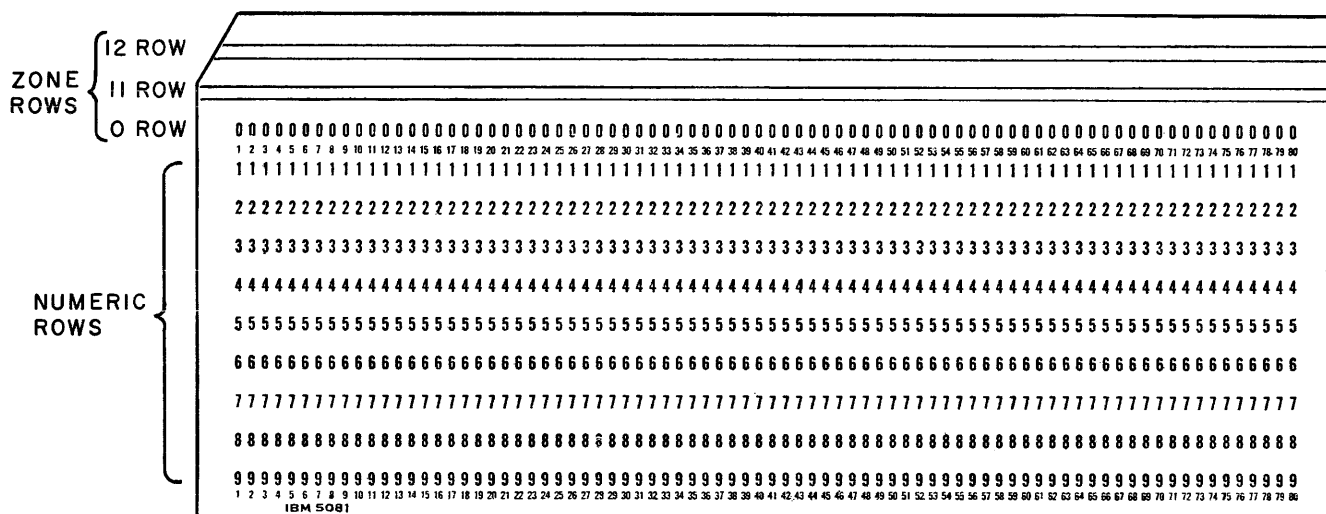
The magnetic tape element provides storage capacity for large blocks of information in a very small space; one reel of magnetic tape can contain about 1.2 million words. Although the random access time for information stored on tape is comparable to that for information on punched cards, the rate of information transfer to or from tape is faster than the rate of transfer to or from cards. The last IO unit, the IO register, can be used for partial or complete erasure of core memory.

4.2 PUNCHED CARDS AND CARD FORMS

4.2.1 General

The basic IBM punch card contains 80 vertical columns of 12 rows each. (See fig. 1-45.) Any information punched on such a card is binary in form since the presence or absence of a punch in a particular position is equivalent to a 1 or a 0 in that position. Although an IBM card can hold 960 binary bits (80×12), its use in storing purely binary information has assumed importance only with the advent of binary digital computers. The card was originally designed for storage of decimal and alphabetical information. The designations of the rows as numeric or zone rows are derived from this original use of the card. For example, a decimal digit can be stored in each column of the card by placing a punch in the numeric row of that column corresponding in value to the value of the decimal digit to be recorded (or in the 0 row if a 0 is to be recorded). In effect, the decimal digit is represented within the column by using a vertical positional notation of its magnitude; i.e., a 6 can be stored in column 10 of the card by punching the 6-row of column 10.

Alphabetical characters are stored on an IBM card with two punches per column; one punch in a zone row



NOTE: CORNER NOTCH INDICATES
CARD THICKNESS.

Figure 1—45. Basic IBM Card Showing Hollerith Code Zones

to indicate selection of one-third of the alphabet and one punch in a numeric row to indicate which letter within that third is represented. Thus, the letter A is indicated by a punch in the 12-row and the 1-row, the letter N by a punch in the 11-row and 5-row, and the letter T by a punch in the 0-row and 3-row. In addition to letters, 14 special characters are indicated by either a single punch in a zone row or by a zone row punch

TABLE 1-25. HOLLERITH CODE FOR PUNCHED CARDS

NUMERIC ROW	ZONE ROW			
	NONE	12	11	0
None		+	—	0
1	1	A	J	/
2	2	B	K	S
3	3	C	L	T
4	4	D	M	U
5	5	E	N	V
6	6	F	O	W
7	7	G	P	X
8	8	H	Q	Y
9	9	I	R	Z
8 & 3	+	.	\$,
8 & 4	—	□	*	%

plus two numeric row punches. The coding for all of these symbols, known as the Hollerith code, is shown in table 1-25.

Using Hollerith code, up to 80 alpha-numeric characters may be represented on a single card with one character per column. The card machine associated with the Central Computer System can utilize only 64 columns on each card for storage of information introduced into, or received from, that system. Specifically, columns 17 through 80 of each card are used for this purpose. (See fig. 1-46.) Column 1 through 16, while not normally available to the Central Computer System, are used for card identification which allows processing of punched cards by card machines independent of the Central Computer System. In general, the information field of a card is further subdivided in accordance with the type of information presented on it. The subdivisions of the information field are discussed in connection with each of the two basic card forms:

- a. Instruction card
- b. Binary card

4.2.2 Instruction Card

The card form on which most programs are prepared for initial insertion into the Central Computer System is the instruction card. (See fig. 1-47.) Each instruction card may contain one instruction in alpha-numeric form plus comments to clarify the function of that instruction within a program. The location field on the card specifies the storage location of the instruction. The instruction field contains the operation code in mnemonic code form and the index indicator and index

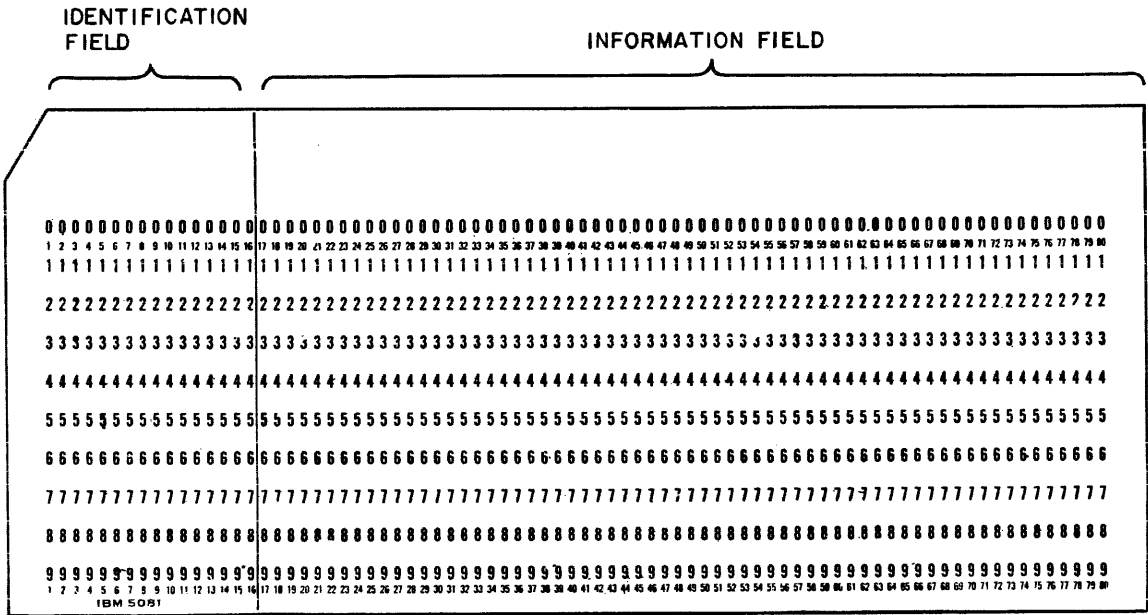


Figure 1-46. Field Division on Punch Cards

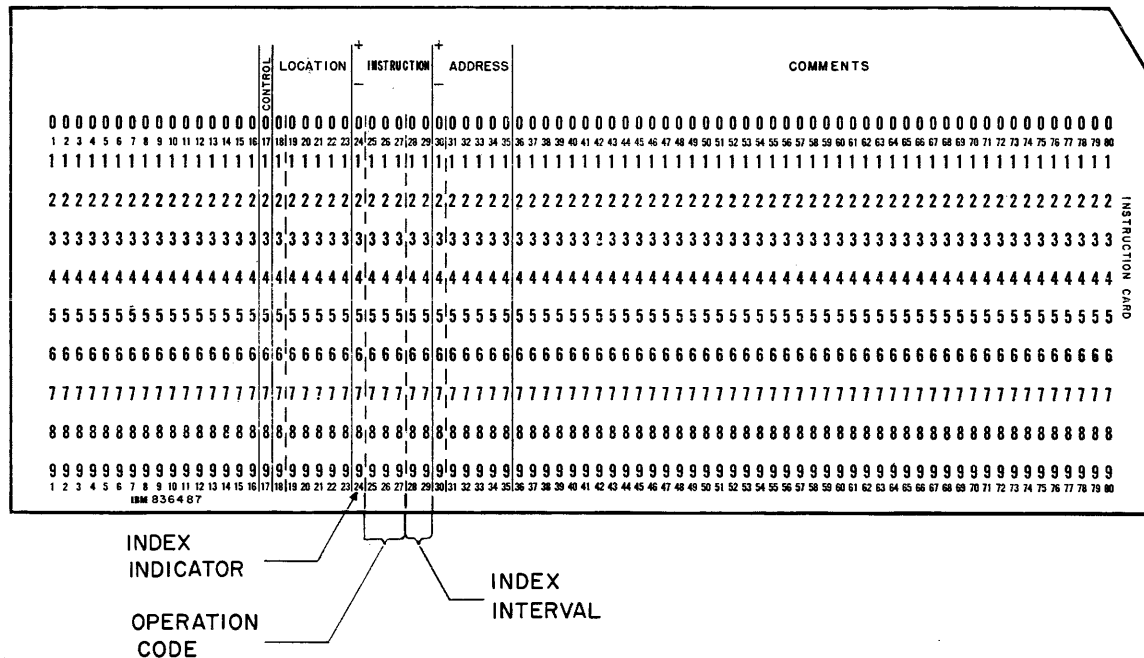


Figure 1-47. Instruction Card

interval in octonary form. The address field specifies the contents of the address half of the instruction. The identification field contains program identification and the preparation date.

The location field and the address field of an instruction card may not specify actual storage locations in core memory. Instead, a symbolic location or a symbolic address may be given. A symbolic location or address is an arbitrary identification of the location or address assigned by the programmer when making up a program. In general, a program is presented to the Central Computer System in symbolic form for assembly into actual form by that system. In this manner, the programmer is relieved of much of the detail work attendant upon preparation of a program. A detailed discussion of this subject is given in Section 5.

It should be noted that a program presented on instruction cards cannot be executed directly by the Central Computer System. Even if actual locations are given on the instruction cards, the program must be translated from its alpha-numeric form into binary form before it can be executed. (The translation process is discussed in connection with symbolic programming in Section 5.) Once the program is translated, it may either be executed and discarded or prepared for reintroduction to the Central Computer System in binary form. When the latter is done, the program is prepared on binary cards.

4.2.3 Binary Card

The binary card utilizes the binary nature of punched cards in storing information. The identification

field, whose contents are identical with the identification field of the instruction card, is the only field on a binary card containing information in Hollerith code. All information on the information field of the binary card is in binary form. (See fig. 1-48.)

Since there are 64 columns within the information field of the binary card, each row contains two Central Computer System words. To distinguish between the two words in a single row, the word in columns 17 through 48 is called the left word while the word in columns 49 through 80 is called the right word. Another convenient terminology describes the former as the word in N-row left and the latter as the word in N-row right where N specifies the row in which the two words are found.

For certain applications, two rows within the information field on a binary card are reserved for control information. Therefore, only 20 program words containing instructions or constants can be stored on a binary card with all control information. These words are located in the 8-row through the 11-row with the first word in 8-row left and the last word in 11-row right. The 12-row, columns 37 through 80, contains the program identification information contained in columns 1 through 8 of the information field, coded in binary form. This binary form of Hollerith code requires 6 bits for an alphabetic character, two for zone, and four for numeric row; a decimal digit requires only the four bits for designation of the number. This binary abbreviation of Hollerith code is given in table 1-26.

As an example of the use of this binary-coded Hollerith code, the letter D is represented as 010100

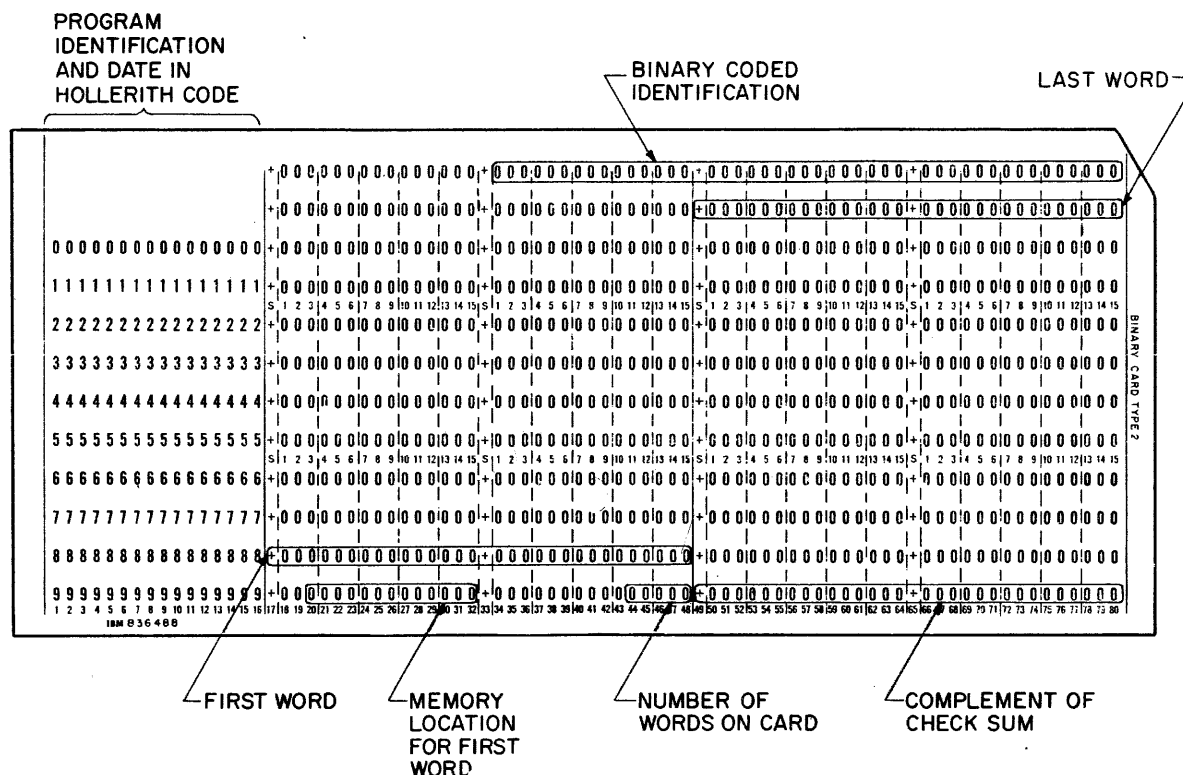


Figure 1-48. Special Punching on Binary Card

TABLE 1-26. BINARY FORM OF HOLLERITH CODE

CODE BITS	CONTENT	MEANING
First two	00	No zone punched
	01	12 zone
	10	11 zone
	11	0 zone
Last four	0001	1-row
	0010	2-row
	0011	3-row
	0100	4-row
	0101	5-row
	0110	6-row
	0111	7-row
	1000	8-row
	1001	9-row

while the digit 7 appears as 0111 if the position it is contained in is known to contain a number rather than a letter; i.e., the 00 indicating no zone is understood.

The control information shown in 9-row left of the binary card may include specification of the memory location for the first word contained on the card (columns 20 through 32) and the number of words on the card (columns 44 through 48). In addition to these items, the word in 9-row left may specify that the program words are to be stored on a drum (column 33), the drum field (columns 37 through 42), and the interleave if any (columns 18 and 19).

The word in 9-row right may contain the complement of a number called the check sum. The check sum, a form of redundancy check on information transfer, is the sum of all words on the card except the words in the 12-row and 9-row. With the check sum in complement form on the binary card, the Central Computer System can check for any errors in reading the words on the card by adding these words to the check sum; a zero result indicates that no error has occurred.

4.2.4 Card Image

When the contents of an IBM card are transferred into the Central Computer System, they appear therein in a form known as a card image. Figure 1-49 shows the relationship between the words on a card and the words in a card image. In effect, the card image in memory contains the words from the card in the order 9-row through 12-row, with the word from the left half of each row preceding the word from the right

12-ROW LEFT	12-ROW RIGHT
11-ROW LEFT	11-ROW RIGHT
0-ROW LEFT	0-ROW RIGHT
1-ROW LEFT	1-ROW RIGHT
2-ROW LEFT	2-ROW RIGHT
3-ROW LEFT	3-ROW RIGHT
4-ROW LEFT	4-ROW RIGHT
5-ROW LEFT	5-ROW RIGHT
6-ROW LEFT	6-ROW RIGHT
7-ROW LEFT	7-ROW RIGHT
8-ROW LEFT	8-ROW RIGHT
9-ROW LEFT	9-ROW RIGHT

IBM CARD

MEMORY LOCATION X	9-ROW LEFT
X+1	9-ROW RIGHT
X+2	8-ROW LEFT
X+3	8-ROW RIGHT
X+4	7-ROW LEFT
X+5	7-ROW RIGHT
X+6	6-ROW LEFT
X+7	6-ROW RIGHT
X+8	5-ROW LEFT
X+9	5-ROW RIGHT
X+10	4-ROW LEFT
X+11	4-ROW RIGHT
X+12	3-ROW LEFT
X+13	3-ROW RIGHT
X+14	2-ROW LEFT
X+15	2-ROW RIGHT
X+16	1-ROW LEFT
X+17	1-ROW RIGHT
X+18	0-ROW LEFT
X+19	0-ROW RIGHT
X+20	11-ROW LEFT
X+21	11-ROW RIGHT
X+22	12-ROW LEFT
X+23	12-ROW RIGHT

CARD IMAGE

Figure 1-49. Relation of Card Image to IBM Card

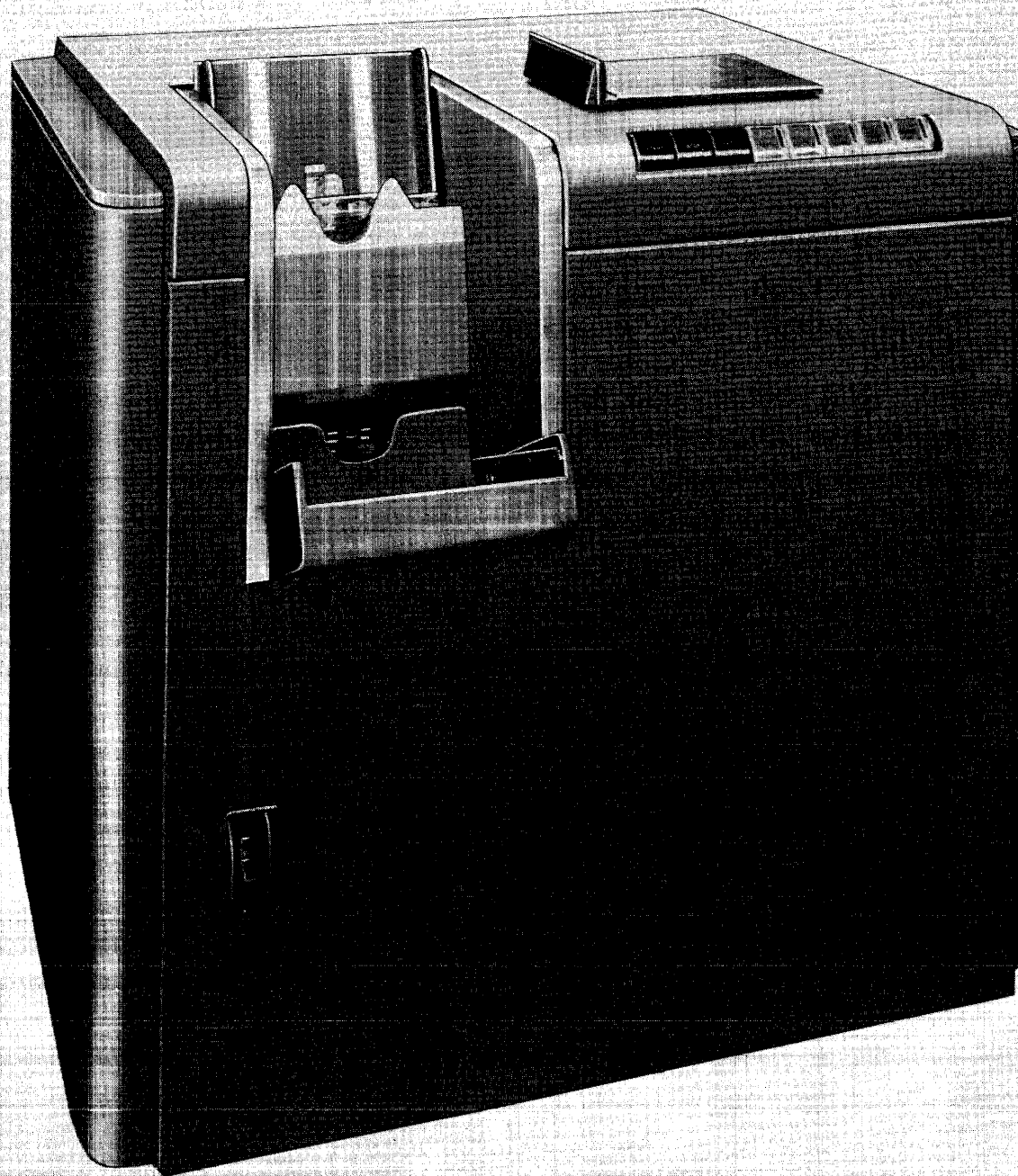


Figure 1—50. Card Reader, Type 713

half of that row. The form of the card image is determined by the modes of operation of the card machines and the Central Computer System; i.e., the card machines can handle 64 bits at a time while the Central Computer System can handle only 32 bits at a time. Since the card image can contain all the information from a card in a known pattern, it is possible to program the Central Computer System to interpret this information just as if the information were still in the pattern present on the card itself.

It is important to realize that programmed interpretation of the information in a card image requires a prior knowledge of the form of that information on the card from which the card image was obtained. If the character A is presented in Hollerith code in column 26 of an instruction card, the card image of that card will contain a 1 in bit L9 of the word from 12-row left and a 1 in bit L9 of the word from 1-row left. A translation program must take account of the relationship between these bits in what would be unrelated words

if not part of a card image. Similarly, a program for the preparation of a card image for delivery to a card machine must be written with the requirements of the card image kept in mind.

4.3 CARD READER

4.3.1 Operation

The card reader (IBM type 713) shown in figure 1-50 allows the insertion of information from punched cards directly into the Central Computer System. The card reader reads a card a row at a time within a cycle of 400 milliseconds, with a maximum reading rate of 150 cards per minute. Only columns 17 through 80 of each row are read, allowing the storage of the information from one row within two Central Computer System words. The reading of a card 9-row first produces the card image in core memory described in 4.2.4. Approximately 15 milliseconds elapse between the reading of successive rows on a card. The remaining time within the card reader cycle ($400 - 11 \times 15$ milliseconds) is used to move the card into and out of the reading section of the card reader.

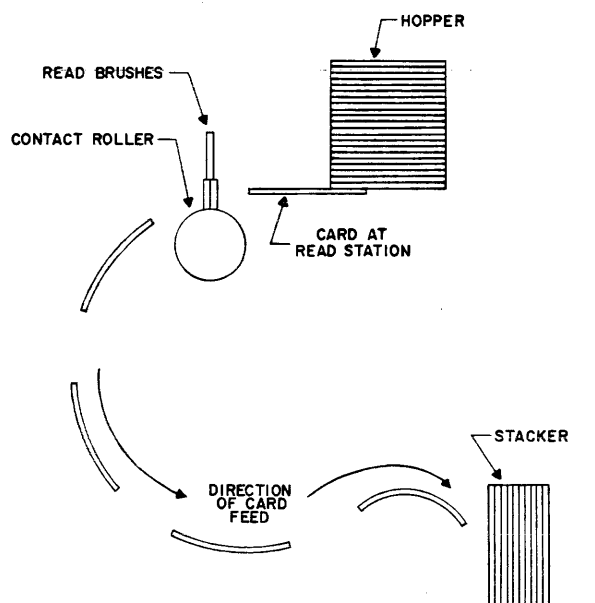


Figure 1-51. Card Feed Unit, Card Reader

The cards to be read are placed in the card reader hopper face down and 9-row to the rear. When the reading of a card is called for by the Central Computer System program, the card is moved bottom edge first, between the read brushes and the contact roller. (See fig. 1-51.) A punch in a particular column of the card row being read is sensed by the brush in that column making contact with the roller. As each row is read, the card is advanced to place the next row under the

read brushes. After all rows on the card have been read, the card is eventually placed in the stacker to be discarded or stored for later reuse.

4.3.2 Controls and Indicators

The controls and indicators on the card reader are shown in figure 1-52. Their functions are listed in table 1-27.

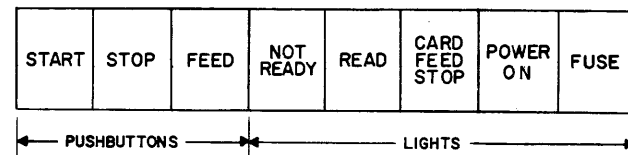


Figure 1-52. Card Reader Controls and Indicators

TABLE 1-27. CARD READER CONTROLS AND INDICATORS

CONTROL OR INDICATOR	FUNCTION
START push-button	If not already done, feeds card from hopper to read station, turns off NOT READY light, and transfers control to Central Computer System; inoperative while card reader is under program control.
STOP pushbutton	Returns card reader to manual control and turns on NOT READY light; action is delayed if pressed during card cycle.
FEED pushbutton	Advances cards through card feed unit while pressed; inoperative unless NOT READY light is on.
NOT READY indicator	On if card reader is not ready; i.e., not under program control, no card at read station, stacker is full, improper power is applied, or fuse is blown.
READ indicator	On while information is being transferred to Central Computer System.
CARD FEED STOP indicator	On if a card feed failure has occurred.
POWER ON indicator	On if a-c power is applied.
FUSE indicator	On for blown fuse if 40-volt a-c power is applied.

TABLE 1-28. CARD READER PROGRAM

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>SEL</i> (01) ₈	—	Selects card reader for use in all subsequent IO operations until another IO unit is selected.
2	<i>LDC</i>	<i>x</i>	Places in IO address counter the address of the first memory location to receive a word from card reader.
3	<i>BSN</i> (11) ₈	<i>y</i>	Senses for not ready condition at selected IO unit; branches if not ready to instruction at address <i>y</i> .
4	<i>RDS</i>	<i>n</i>	Reads <i>n</i> words from card reader.
5			Next program

4.3.3 Information Transfer

Before the card reader can be used to insert information into the Central Computer System, it must be placed in the ready condition; a deck of cards must be placed in the hopper; and the START pushbutton pressed. A program of the form given in table 1-28 then brings information from the card reader into the Central Computer System.

The *Sense* (IO unit not ready) instruction is included within the program to prevent hanging up the Central Computer System on an impossible IO operation. The branch of program control may be either to a *Program Stop* instruction to call attention to the card reader or to another program which will return to this program in the expectation that the card reader will be readied before the return.

The *RDS n* instruction actually initiates the reading operation of the card reader. If $n = 24$ or 30_8 , all the words on one card are read and transferred to the Central Computer System. If $n > 24$, all the words on one card and as many words as specified are read from succeeding cards. When the number of words to be read is fewer than the number of words on the card, only the specified number of words is transferred to the Central Computer System although the entire card is passed by the read brushes of the card reader. The remaining words on that card cannot be read unless the card is manually repositioned for another pass through the card reader. An *RDS 0* instruction thus moves one card past the read brushes without transferring any information. In effect, the card is skipped.

The path followed by information from the card reader is shown in figure 1-53. As each row of a card passes under the read brushes, two words are read. The word from the left half of the row (columns 17 through 48) is placed in the IO register; the word from

the right half of the row (columns 49 through 80) is placed in the IO buffer register. A break request is generated by the card reader and sent to the Central Computer System. At the first convenient point, the Central Computer System executes a break cycle transferring the first word into core memory. (Refer to 3.1.3.1.) During this break cycle, the second word is transferred into the IO register. A second break cycle is initiated by the Central Computer System immediately following the first to place the second word in core memory. The Central Computer System then returns to internal operations until two more words and another break request are received from the card reader, causing the process to repeat.

The IO operation involving the card reader continues until the IO word counter is stepped to zero. At this time, a signal is sent to the card reader stopping the transfer of information but not stopping the card reader read cycle. When the read cycle is completed, the card reader sends a disconnect signal to the Central Computer System, clearing the IO interlock and terminating that IO operation. Thus, although the IO word counter may step to zero after the first word from a card is stored in core memory, no IO class instructions may be performed until the card reader completes its read cycle. Use can be made of this fact to delay the execution of subsequent instructions if these subsequent instructions are to process the information being read by the card reader. The next program referred to in table 1-28 can be preceded by a *SEL* (01)₈ instruction. This instruction cannot be executed while the IO interlock is on. Therefore, the instructions following it will not be executed until the IO interlock is cleared. Although the program delay may be excessive in the case of reading less than a full card, it is a sure means of delaying a program until the information needed by the program is available in core memory.

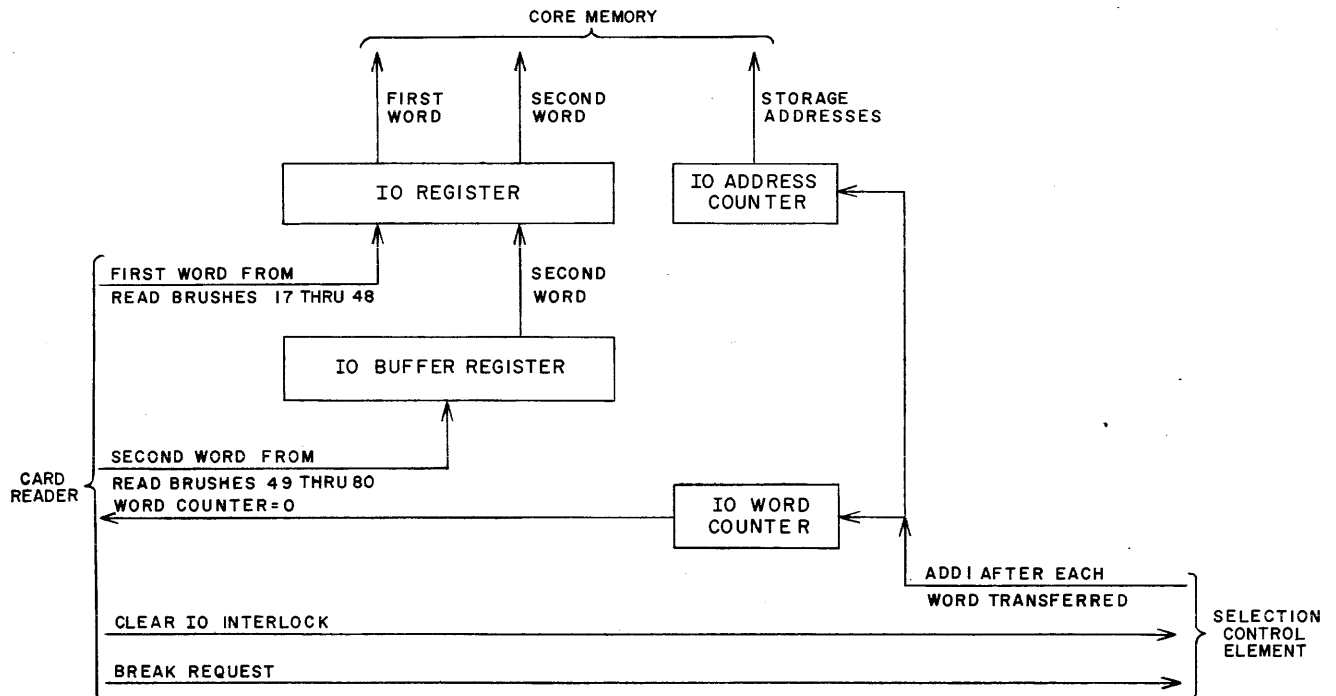


Figure 1-53. Card Reader Information Flow

4.4 CARD PUNCH

4.4.1 Operation

The card punch (IBM type 723) shown in figure 1-54 allows the Central Computer System to present processed information in punched card form. The card punch has an operating cycle of 600-millisecond duration, allowing a maximum punching rate of 100 cards per minute. Since the card punch normally punches only columns 17 through 80 of each row, the Central Computer System must supply two words for each row. The order in which the card punch handles a card, a row at a time and 9-row first, requires the preparation of a card image in core memory in order to have punched information appear correctly on a card. (Refer to 4.2.4.) Approximately 31 milliseconds elapse between the punching of successive rows on a card. The remaining time within the card punch cycle ($600 - 11 \times 31$ milliseconds) is utilized in moving the card into and out of the punching section of the card punch.

Unpunched cards of the desired type are placed in the card punch hopper face down and 9-row to the right. When the punching of a card is called for by the Central Computer System program, the card is moved, bottom edge first, between the punches and the dies. There is one punch per column (See fig. 1-55). The card is punched in a particular column of the row under the punches if the words from the Central Computer System contain a 1 in the bit position corresponding to that column.

After the card passes through the punching station, it goes to a reading station. The card punch can, under program control, connect the punch brushes reading columns 1 through 16 to the punch magnets over the corresponding columns. With this provision, the information in columns 1 through 16 of a card just punched can be duplicated on the card passing through the punch station by gang punching. A programming provision also exists for switching the inputs for the punch magnets over columns 17 through 32 to the punch magnets over columns 1 through 16. Thus, the Central Computer System can place information in the identification field of one card, then have this information gang punched on each succeeding card. After a card has been punched and read by the punch brushes, it is placed in the stacker from which it may be removed for storage or for reinsertion into the Central Computer System through the card reader.

4.4.2 Controls and Indicators

The controls and indicators on the card punch are shown in figure 1-56. Their functions are listed in table 1-29.

4.4.3 Information Transfer

Before the card punch can receive information from the Central Computer System, it must be placed in the ready condition; a deck of blank cards must be placed in the hopper, the various interlocks around the punching station must be in place, and the START push-

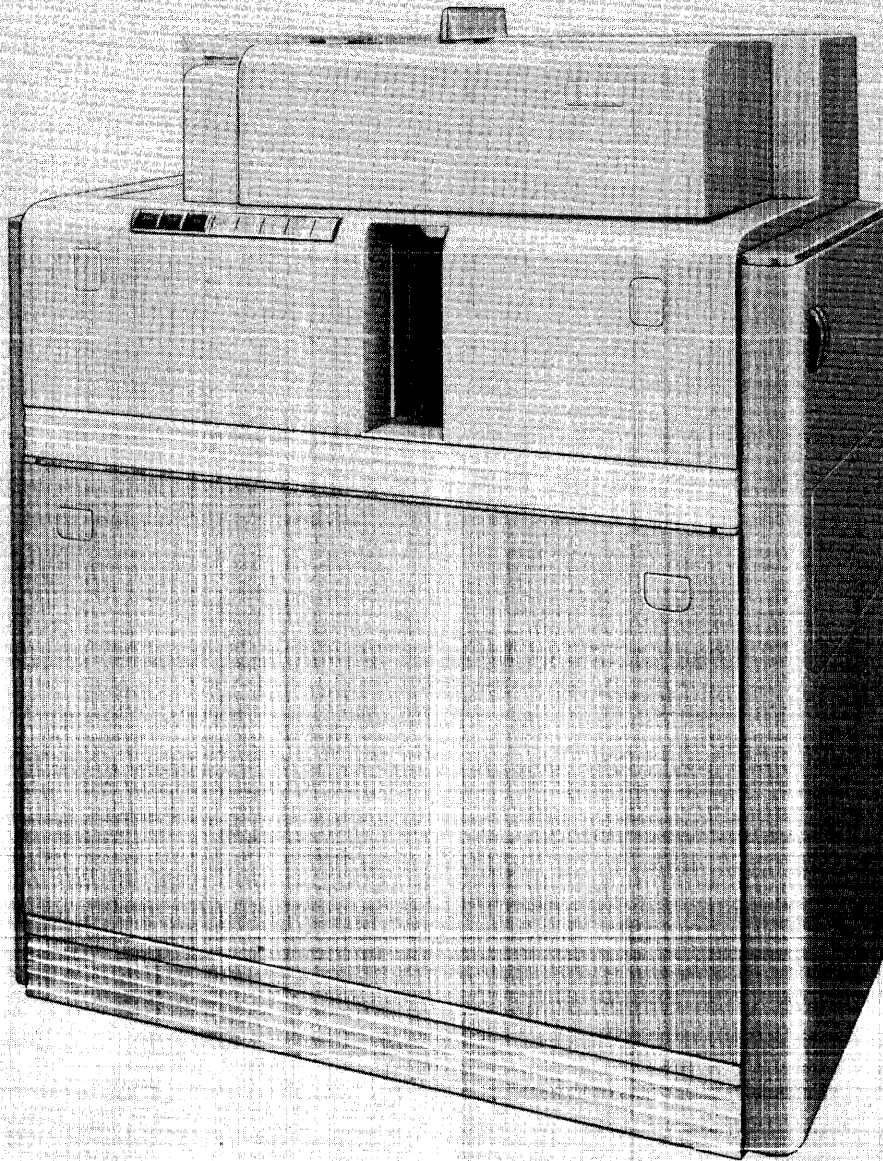


Figure 1-54. Card Punch, Type 723

button pressed. A program of the form given in table 1-30 then brings information from the Central Computer System to the card punch.

The *WRT n* instruction actually initiates the writing operation of the card punch. If $n = 24$ or 30 , the complete card image is transferred to one punched card. If $n > 24$, one card is completely punched for each

group of 24 words while any remaining group of words less than 24 is punched on one card and that card is passed by the punching station. No more words can be punched on this last card unless the card is manually repositioned for another pass through the card punch. A *WRT O* instruction thus passes one card through the punching station without transferring any information to the card, in effect, skipping the card.

TABLE 1-29. CARD PUNCH CONTROLS AND INDICATORS

CONTROL OR INDICATOR	FUNCTION
START pushbutton	If not already done, feeds card from hopper to punch station, turns off NOT READY light, and transfers control to Central Computer System; inoperative while card punch is under program control.
STOP pushbutton	Returns card punch to manual control and turns on NOT READY light; action is delayed if pressed during card cycle.
FEED pushbutton	Advances card through card feed unit while pressed; inoperative unless NOT READY light is on.
NOT READY indicator	On if card punch is not ready; i.e., not under program control, no card at punch station, stacker is full, improper power applied, or fuse is blown.
WRITE indicator	On while information is being transferred from Central Computer System.
CARD FEED STOP indicator	On if a card feed failure has occurred.
POWER ON indicator	On if a-c power is applied.
FUSE indicator	On for blown fuse if 40-volt a-c power is applied.

TABLE 1-30. CARD PUNCH PROGRAM

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>SEL</i> (02) ₈	—	Selects card punch for use in subsequent IO operations until another IO unit is selected.
2	<i>LDC</i>	<i>x</i>	Places in IO address counter the core memory address of first word of first card image.
3	<i>BSN</i> (11) ₈	<i>y</i>	Senses for not ready condition at selected IO unit; branches if not ready to instruction at address <i>y</i> .
4	<i>WRT</i>	<i>n</i>	Writes <i>n</i> words at card punch.
5	<i>SEL</i> (02) ₈	—	Delays next program.
6			Next program.

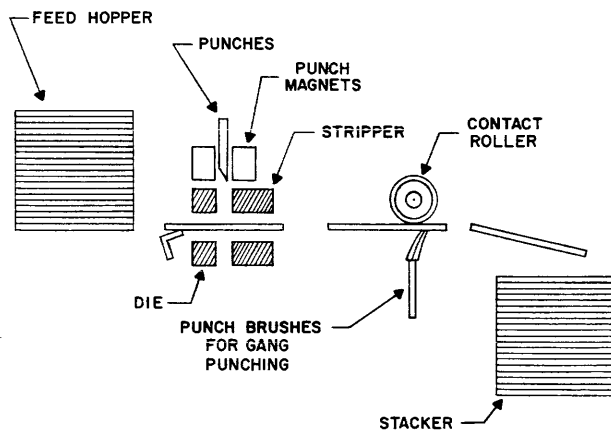


Figure 1-55. Card Feed Unit, Card Punch

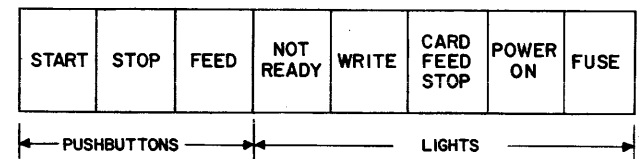


Figure 1-56. Card Punch Controls and Indicators

The path followed by information to the card punch is shown in figure 1-57. The *WRT n* instruction starts the punch cycle of the card punch, moving a blank card into position so that its 9-row is under the punches. At this instant, a break request is sent to the Central Computer System, causing the execution of two break cycles in sequence to supply two words to the card punch.

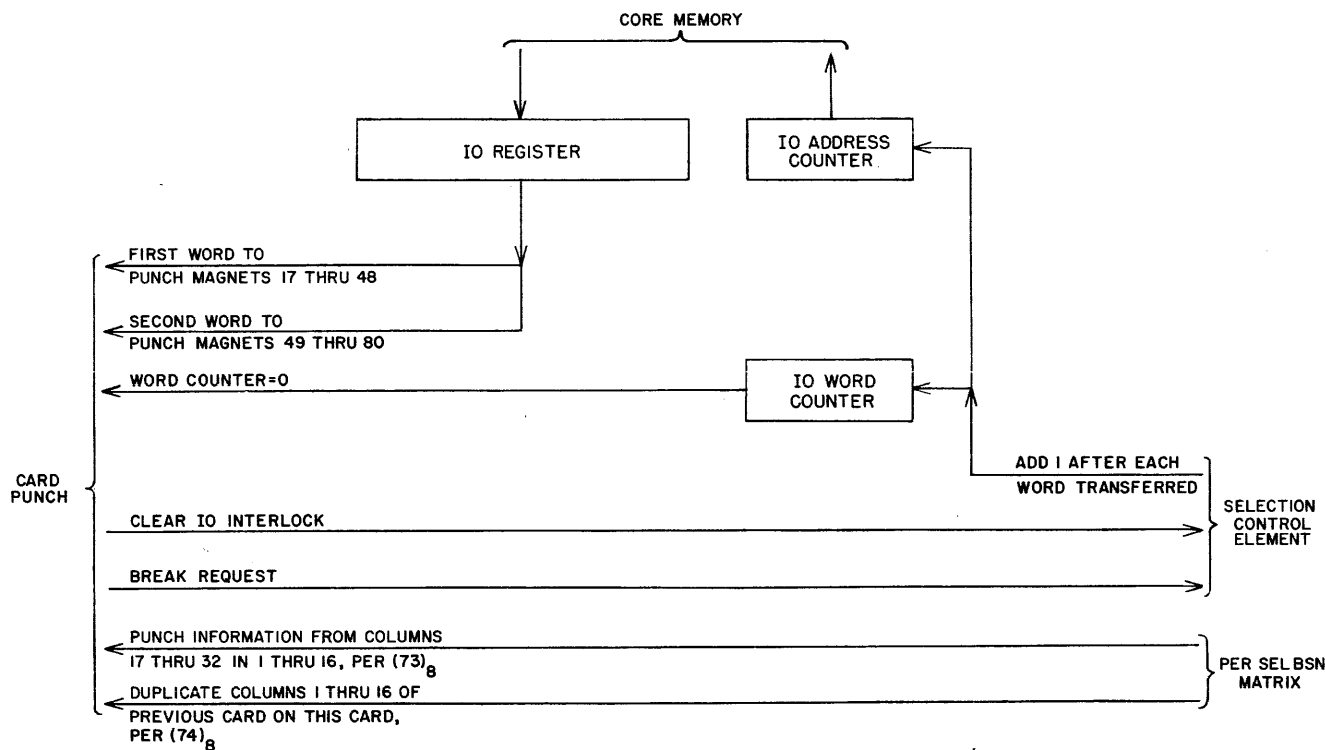


Figure 1-57. Card Punch Information Flow

(Refer to 3.1.3.1.) The first word is normally applied to the punch magnets over columns 17 through 48 while the second word is applied to those over columns 49 through 80. The Central Computer System returns to internal operations after supplying the two words until the card punch moves the next row of the card under the punches, causing the generation of another break request and a repetition of the transfer process.

The IO operation involving the card punch continues until the IO word counter is stepped to zero. When this occurs, a signal is sent to the card punch stopping the transfer of information but not stopping the card punch card cycle. When the punch cycle is completed, the card punch sends a disconnect signal to the Central Computer System to clear the IO interlock and end the IO operation.

In order to prepare a program on a deck of binary cards, it is necessary to have information which would normally be punched within the information field of the card punched in the identification field instead. The instruction *PER (73)₈* following the *WRT* instruction causes the information normally placed in columns 17 through 32 to be punched in columns 1 through 16. Thus, a card can be punched bearing identification information. This information can be duplicated on each subsequent card by programming a *PER (74)₈* instruction just after the *WRT 30₈* instruction which initiates the punching of that card. The *PER (74)₈* instruction causes gang punching of the information from columns

1 through 16 of the card just punched into the corresponding columns of the card being punched during the current card cycle. In order to maintain this gang punching for each card, the *PER (74)₈* instruction must be given for each card cycle.

4.5 LINE PRINTER

4.5.1 Operation

The line printer (IBM type 718) shown in figure 1-58 provides the means for the Central Computer System to prepare information in printed alpha-numeric form. The line printer prints a line at a time by the simultaneous positioning of as many of its 120 type wheels as is possible within one card cycle of 400 milliseconds. The normal printing rate of the line printer is 150 lines of 64 characters each per minute. The Central Computer System must normally supply a full card image for each line of 64 characters. A type wheel is shown in figure 1-59. As can be seen, all characters selectable by Hollerith code are arranged on the type wheel numeric sectors with the four characters within each sector selectable by the zone indication.

At the start of a print cycle, all 120 wheels are lined up with the character 0 in printing position. All the wheels begin rotating as information is supplied to their positioning controls. The information applied to the positioning controls of each type wheel arrives in the order used in reading one column on a punched card; i.e., 9-row, 8-row, etc., then 0-row, 11-row, and

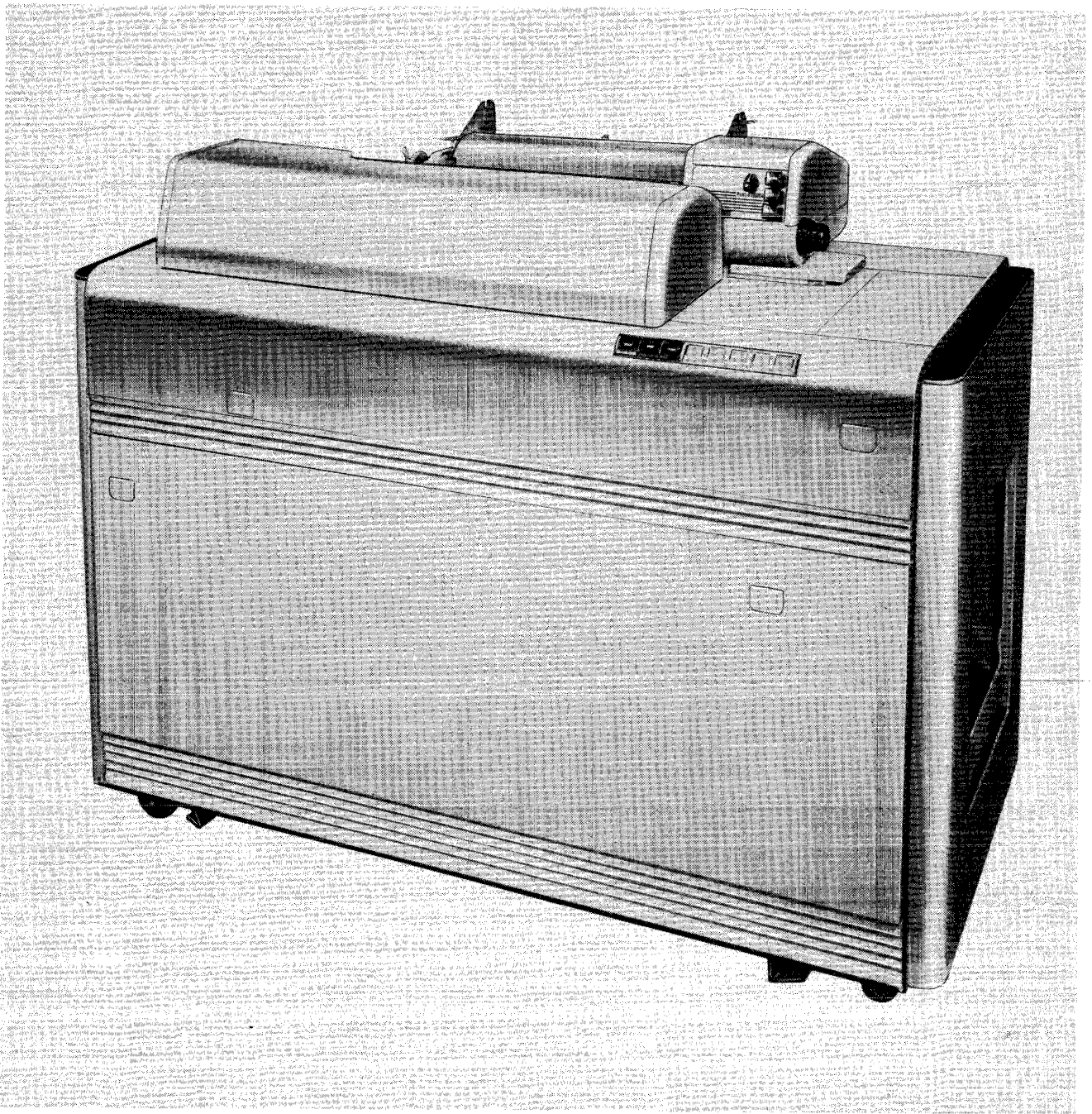


Figure 1-58. Line Printer, Type 718

12-row. A bit arriving at a time corresponding to the reading of a given numeric row stops the wheel with the 0-zone character of that numeric sector in printing position. (A double numeric punch, 8-3 or 8-4, selects the 0-zone character within the indicated sector.) When the zone bits are applied, the type wheel is allowed to advance one character position for each nonpresent zone bit or locked when a zone bit is supplied. Thus, a type wheel positioned at the 0-zone character of the 7-row sector by a bit received at 7-row time is either locked

in that position if a 0-zone bit is received or advanced to the 11-zone character if no 0-zone bit is applied. Again, the wheel is advanced one character for each subsequent missed zone bit until 12-zone time. If no 12-zone bit is applied, the wheel advances to the no zone character, 7, and locks. Just after 12-zone time, those print wheels which are energized strike the paper to print out the information supplied to the line printer. It should be noted that, in normal operation, the paper form is spaced while the print wheels are being posi-

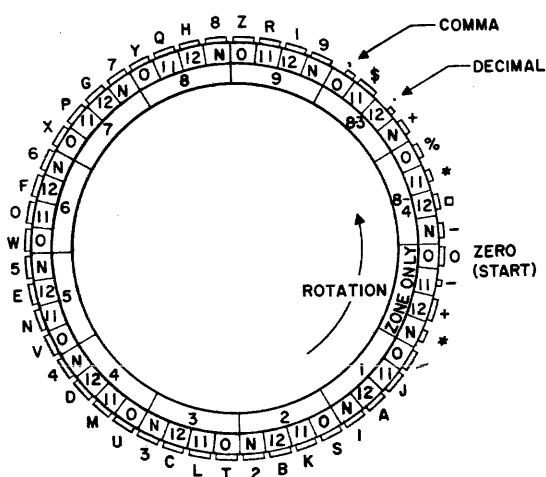


Figure 1-59. Type Wheel, Pictorial Diagram

tioned. Thus, the printed line remains under the print wheels until the next print cycle is begun.

This discussion of the normal operation of the line printer is based on the assumption that the wiring of the line printer control panel is of the simplest nature. Overall operation of the line printer is extremely flexible when the control panel is taken into account.

4.5.2 Controls and Indicators

4.5.2.1 Control Panel

The control panel for the line printer is shown in figure 1-60. As shown on the right half of that panel, the information from the Central Computer System appears at the CALC EXIT hubs. The ODD WORDS exit hubs receive the bits of the words corresponding to the left half of a card image while the EVEN WORDS hubs receive the words corresponding to the right half of a card image. For normal printing of 64 characters per line with no variations, these CALC EXIT hubs are wired to the PRINT ENTRY hubs. There are 120 PRINT ENTRY hubs, one for each print wheel. The information from the Central Computer System can be applied to any 64 of the 120 hubs, thus allowing printout with any number of characters per printed column and in any desired order within each line. If it is desired to insert a character in a particular column, the exit hubs within the box labelled CHARACTER EMITTER can be used for that purpose. Each hub within this group emits pulses at the correct times to generate the designated characters. For example, the L emitter provides a pulse at 3-time and at 11-time. The BUS hubs shown on both halves of the panel are provided for multiple connections, each bus is a set of four hubs wired together. The FILTER hubs are simply one-way circuits, usable for isolation purposes.

The CO-SELECTORS 1 through 32 are relay-controlled transfer switches. Each CO-SELECTOR provides

five single-pole double-throw transfer circuits. The five hubs in the C row are the switch arms, the hubs in the N row are the normal connections, and the hubs in the T row are the transfer connections. Each CO-SELECTOR is controlled by a CO-SELECTOR PICKUP (shown on the left half of the panel). By applying a pulse to a given CO-SELECTOR PICKUP, all five switches of the corresponding CO-SELECTOR transfer, breaking their C-N connections and making their C-T connections.

The PILOT SELECTORS 1 through 10, top and bottom group and their corresponding PILOT SELECTOR PICKUPS allow the transfer of two single-pole double-throw relay contacts. These PILOT SELECTORS can be used to control a group of CO-SELECTORS simultaneously. The PILOT SEL COUPLING EXITS provide a d-c level at each hub when the corresponding PILOT SELECTOR PICKUP is energized. The PILOT SEL COUPLING EXITS can thus be used in place of the original pickup signal to avoid overloading its source.

An interlock is provided on the control panel in the hubs labelled PR ON. Unless these two hubs are connected, the line printer is inoperative. Certain hubs provide pulses at specified times for control purposes. The PRINT CYCLES hubs are energized once each print cycle. ALTER hubs 1 through 4 are energized at the same time as the PRINT CYCLES hubs, providing that the corresponding ALTERATION switch is on. The SPLIT COLUMN CONTROL hubs are energized sequentially during a print cycle in such a manner that a hub between two numbers on the panel is energized between the times of reception of the rows designated by those numbers. The OPERATE EXITS are energized when the corresponding PER (*u*) instruction is executed by the Central Computer System. Similarly, an impulse wired to SENSE ENTRY hubs 1 or 2 can be detected by the appropriate BSN (*u*) instruction. OPERATE EXITS 1 through 10 correspond to PER (51)₈ through PER (62)₈ and SENSE ENTRY hubs 1 and 2 are examined by BSN (31)₈ and BSN (32)₈ respectively.

The hubs that have not been described are all used in controlling paper feed through the carriage of the line printer. The SHORT SKIP hubs on the right half of the control panel may be wired to skip up to seven lines on a form without delaying printing. If more than seven lines are to be skipped, a separate cycle is required to allow time for the paper feeding through the carriage. Normal spacing is provided by wiring SPACE hubs 1 or 2 for a single or double space between each line. If the EXTRA hubs are plugged, the effect of the SPACE hubs is doubled. The SUP hubs, if impulsed during a print cycle, suppress all normal spacing during that print cycle, thus allowing printing of more than 64 characters per line. The SEL hub provides for selective spacing under control of a paper tape on the printer carriage. This control tape contains 12 columns, each

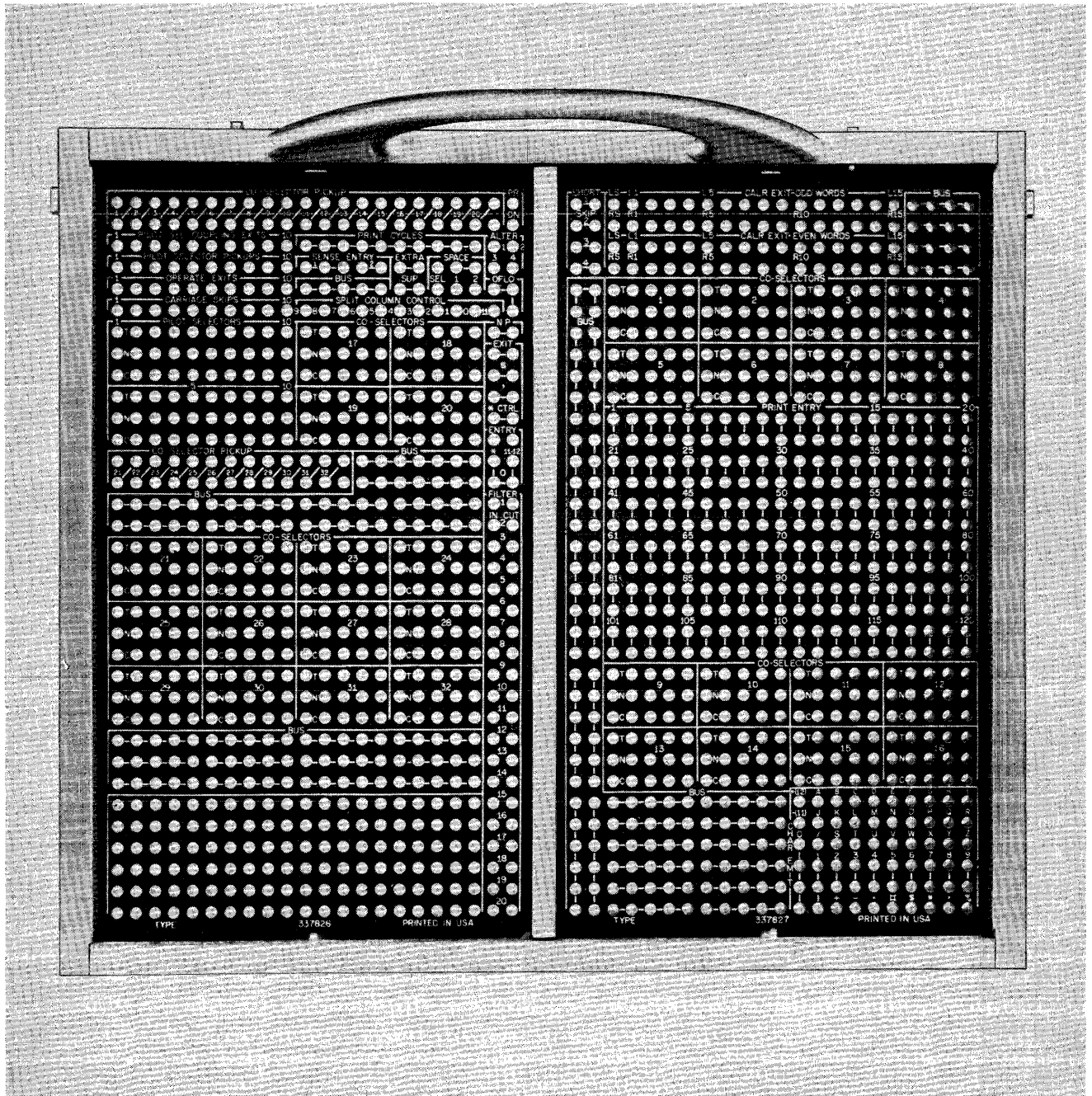


Figure 1-60. Line Printer Control Panel

sensed for punches by a separate brush. If the SEL hub is impulsed, the carriage advances the paper until a punch is sensed in column 11 of the control tape. Similarly, CARRIAGE SKIP hubs 1 through 10, when impulsed, advance the paper form until a punch is sensed in the corresponding column on the control tape. In addition, a punch is placed in column 12 to correspond to a position just short of the end of a paper form. When this punch is sensed, the OVFL hubs are energized to indicate that printing is about to overflow the end of the form. A signal from the OVFL hubs can be used

to initiate a carriage skip to the beginning of the next paper form. Some provision must be made to prevent printing if the carriage skip exceeds seven lines. Although the NP hubs will suppress printing during the print cycle in which they are energized, they will also suppress spacing and cannot, therefore, be used to prevent printing during a long skip.

4.5.2.2 Manual Controls and Indicators

The controls and indicators on the line printer are listed in table 1-31 and shown in figure 1-61.

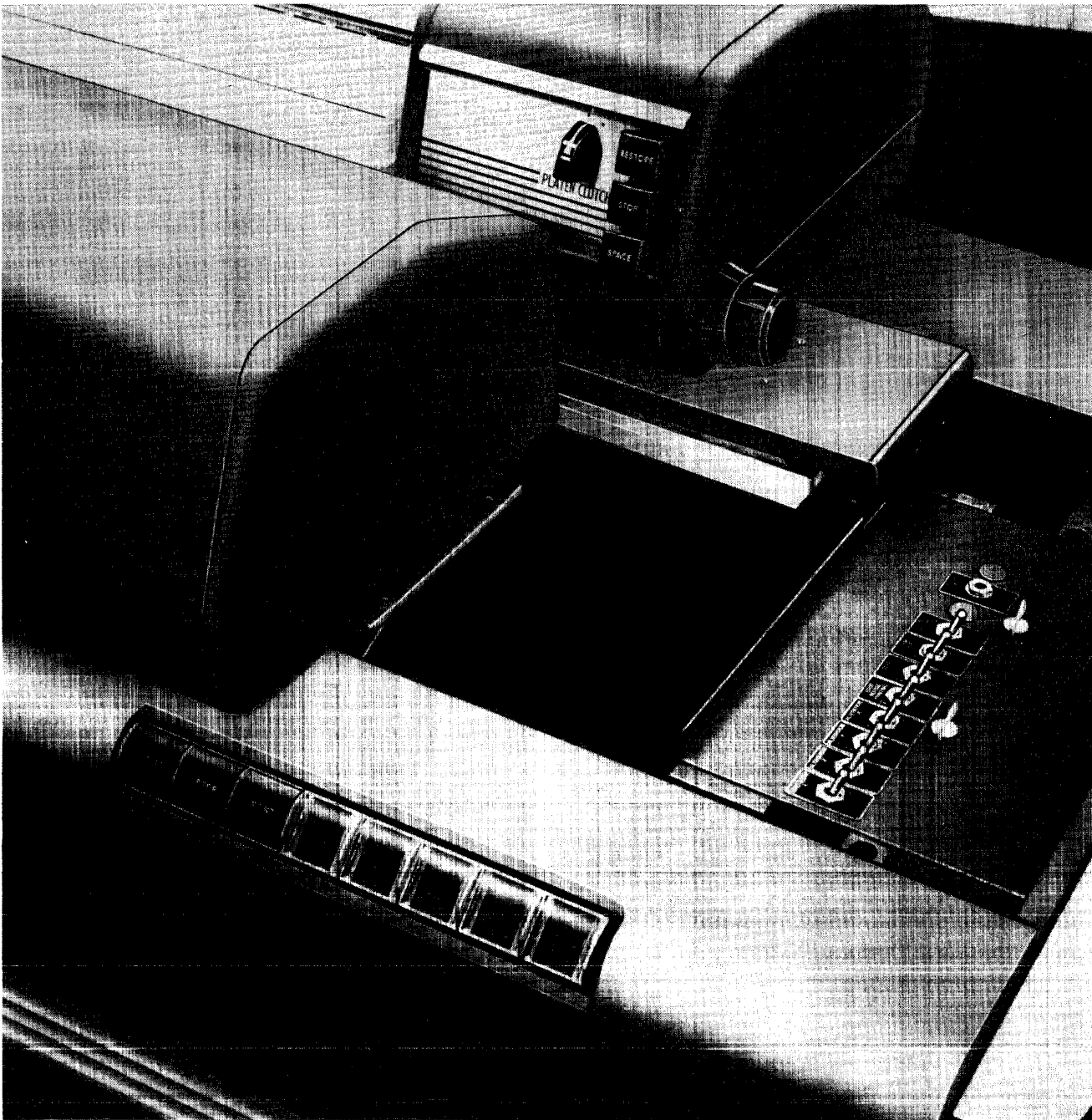


Figure 1-61. Line Printer Controls and Indicators

More complete information on the line printer is available in the IBM Manual of Instruction for the Accounting Machine, Type 407.

4.5.3 Information Transfer

Before the line printer can receive information from the Central Computer System, it must be placed in the ready condition; the paper forms must be fed into the carriage, the control panel must be properly wired and in place, and the START pushbutton pressed. A pro-

gram of the form given in table 1-32 then brings information from the Central Computer System to the line printer.

The basic program for use of the line printer may be much modified depending upon the degree of complexity of control panel wiring. The only limitation upon the programming of this unit is the level of ingenuity applied by the programmer. For this reason, no detailed description of line printer information transfer is offered.

TABLE 1-31. LINE PRINTER CONTROLS AND INDICATORS

CONTROL OR INDICATOR	FUNCTION
START pushbutton	If line printer is in ready condition, transfers control to Central Computer System and turns off NOT READY indicator.
STOP pushbutton	Returns line printer to manual control and turns on NOT READY indicator.
PRINT CYCLE pushbutton	Causes the line printer to print one line; inoperative unless NOT READY light is on and TEST switch is on.
NOT READY indicator	On if line printer is not ready; i.e., fuse is blown, control panel not in place, d-c power not applied, paper form not in place, or line printer not under program control.
WRITE indicator	On during a print cycle initiated by Central Computer System.
FUSE indicator	On when fuse is blown.
FORM STOP switch	Controls stopping of line printer when out of paper; if off, form stop feature inoperative.
FORM STOP indicator	When FORM STOP switch is on, indicator goes on and line printer is rendered inoperative by end of paper form in carriage; cleared by feeding additional forms into line printer.
TEST switch	Equivalent in action to STOP pushbutton.
STOP BEFORE PRINTING switch	If TEST switch is off, this switch can suppress printing without removing the printer from program control.
ALTERATION switches	When on, an impulse is available during each print cycle at the corresponding ALTER hub.

TABLE 1-32. LINE PRINTER PROGRAM

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	SEL (03) ₈	—	Selects line printer for use in subsequent IO operations until another IO unit is selected.
2	LDC	<i>x</i>	Places in IO address counter the core memory address of first word of card image.
3	BSN (11) ₈	<i>y</i>	Senses for not ready condition.
4	WRT	<i>n</i>	Writes <i>n</i> words at line printer.
5	SEL (03) ₈	—	Delays next program.
6			Next program.

4.6 TAPE ELEMENT**4.6.1 Introduction**

The storage medium handled by the tape elements is 1/2-inch oxide-coated plastic tape, similar to that used in sound tape recorders. Information is recorded on the tape in the form of small magnetized areas, each area containing one binary bit. A character, a group

of seven bits on a line perpendicular to the direction of tape travel, is the smallest unit of information handled by the element. A complete character is written or read as a unit by the recording heads of a tape drive unit. Only six of the seven bits in a character contain information; the middle bit is a synchronizing bit used to ensure correct tape reading.

The next larger unit of tape information, known as a word, corresponds in capacity to a Central Computer System word, holding 32 bits plus a parity bit. (See fig. 1-62.) Each tape word includes six characters, five full ones and one (the first character) containing only three information bits. Each word nominally occupies 0.0242 inch of tape, thus allowing storage of 41.3 words per inch. One reel (the physical unit) of tape, 2400 feet long, can hold about 1.2 million words.

A group of words written consecutively on a tape in one IO operation is known as a record. The end-of-record is indicated by a 3/4-inch blank space on the tape (produced by erasing the tape that far beyond the last word in a record). If a number of IO transfers is carried out on one tape producing a number of records written consecutively on the tape, the group of records is known as a file. The end-of-file is indicated by a one-word last record in which the three bit positions, normally left blank, are written. (See fig. 1-63.)

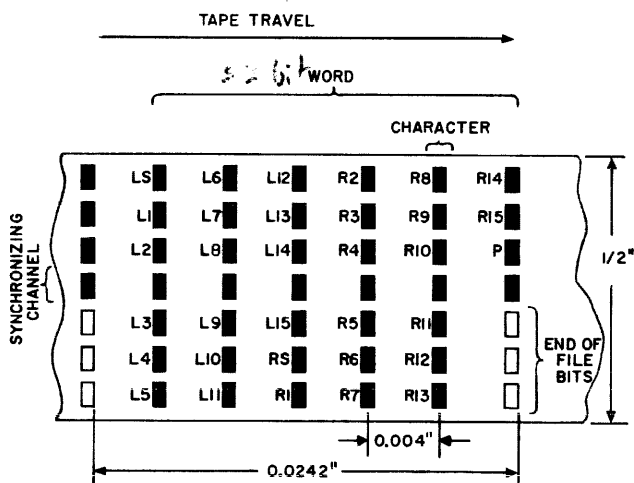


Figure 1-62. Tape Word Bit Positions

In addition to its information content, each tape reel carries two types of control markings in the form of reflective spots on the tape. The spots are sensed by a photoelectric cell in the tape drive unit. One spot, the load point, is placed 10 feet from the beginning of the tape. The length of tape between the load point and an end-of-file mark is designated as the file area. The second spot, the end-of-tape mark, is normally 14 feet from the end of the reel but may be placed anywhere on the tape. Reading or writing a tape starts from the load point. Although a tape is outside the file area if the end-of-tape mark passes the photoelectric cell during a writing or reading operation, that operation may continue for another 4000 words if there are 14 feet of tape beyond the mark. However, a new writing or reading operation cannot be started past the end-of-tape mark. A summary of tape information and control units is given in table 1-33.

The three major components of the tape element are:

- The tape power supply unit
- The tape drive units
- The tape adapter unit

The tape power supply unit provides the non-standard voltages required by the other two components of the tape element. It performs no other logical function.

Six tape drive units are provided for in the tape element. (See fig. 1-64.) Each unit includes read, write, and erase circuits, tape speed control circuits, a tape transport mechanism, and photoelectric cell sensing circuits. The operating specifications of the tape drive unit are summarized in table 1-34.

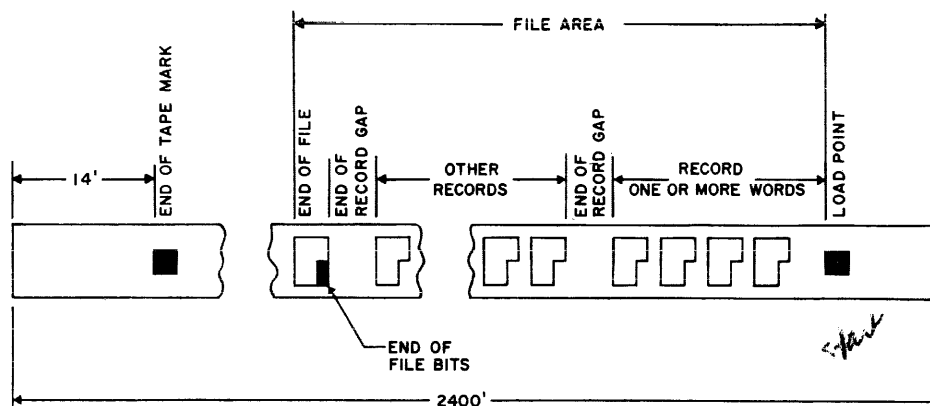


Figure 1-63. Tape Reel Information and Control Units

TABLE 1-33. TAPE INFORMATION AND
CONTROL UNITS

UNIT	DESCRIPTION
Character	Six information bits and one synchronizing bit written in one line across the width of the tape.
Word	A group of six consecutive characters, five full and one containing only three information bits.
Record	A group of consecutively written words.
End-of-record	Blank space on tape $\frac{3}{4}$ inch long.
File	A group of consecutively written records.
End-of-file	Special one-word record indicating the end of a file.
Load point	Reflective spot at beginning of usable tape.
End-of-tape mark	Reflective spot near end of tape.
Reel	Physical tape unit containing 2,400 feet of tape.

TABLE 1-34. TAPE DRIVE UNIT OPERATING
SPECIFICATIONS

CHARACTERISTIC	SPECIFICATION
Maximum reading or writing rate	3,125 words per second
Character density	248 characters per inch
Word density	41.3 words per inch
Tape speed	75 inches per second
Rewind speed	500 inches per second (average)
Reel change time	1 minute (average)
Reel capacity	2,400 feet

Tape speed across the recording heads is maintained by a special motor drive that is independent of the reel drive system. On either side of the recording heads, the tape is fed into vacuum columns which maintain constant tension on the tape. Rotation of the feed reel and take-up reel is controlled by the depths of the tape loops in the vacuum columns.

The tape adapter unit connects the tape drive units with the remainder of the Central Computer System. Functionally, the tape adapter unit receives and interprets command signals from the Central Computer System and, in a sense, performs the change in information form necessary between the tape drive units and the Central Computer System.

4.6.2 Tape Drive Unit

4.6.2.1 Manual Operations

Although mechanical operation of a tape drive unit is normally under control of the Central Computer

System, an operator at the unit may assume manual control when necessary. For example, to change tape reels, the operator takes control of the unit by pressing the RESET pushbutton. The tape is rewound completely onto the feed reel when the operator presses the LOAD REWIND pushbutton. Pressing the UNLOAD pushbutton then unloads the tape from the transport mechanism. The operator may then remove the feed reel and replace it with another reel of tape. To allow the reels to turn freely while threading the new tape through the transport mechanism, the operator presses the REEL RELEASE switches. When threading is completed, the operator releases the REEL RELEASE switches and presses the LOAD REWIND pushbutton which winds the tape to the load point. Before the LOAD REWIND pushbutton is pressed, the load point must be wound onto the take-up reel and beyond the photoelectric sensing unit. Otherwise, when the LOAD REWIND pushbutton is pressed, the load point will

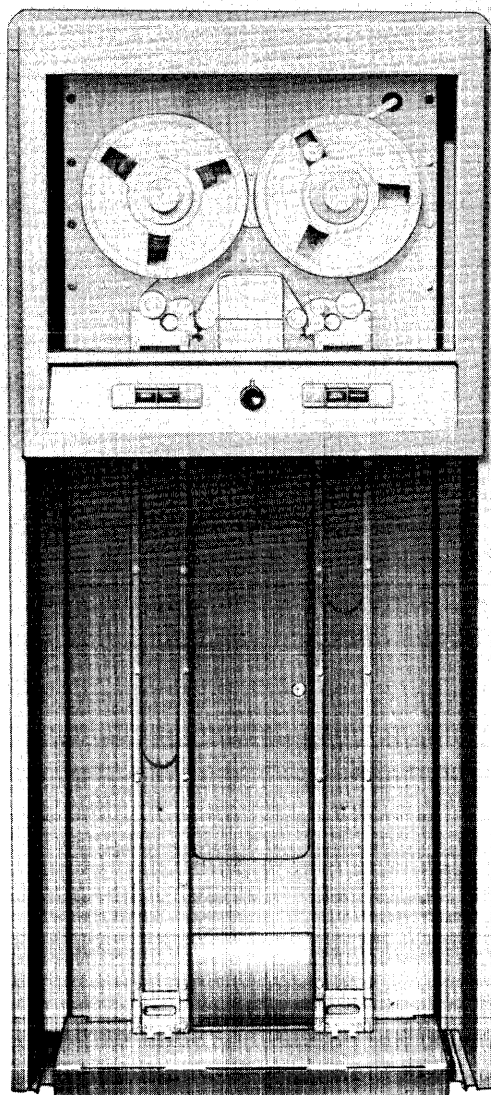


Figure 1-64. Tape Drive Unit

not be sensed to stop rewind and the tape will be torn off the take-up reel. The reel change is now completed. The operator returns control of the tape drive unit to

the Central Computer System by pressing the START pushbutton.

If the operator wishes to protect the contents of the new reel from accidental writing or erasure, the file protection interlock may be energized by removing a special ring from the inner side of the reel. A file protection interlock sensing pin on the tape drive unit slips into the resulting groove, thus preventing writing on that reel. A summary of controls on the tape drive unit and their functions is contained in table 1-35.

In addition to the controls on the tape drive unit, a number of control lights indicate the status of the unit. The control lights are described in table 1-36.

4.6.2.2 Programmed Operation

Three mechanical operations of the tape drive unit, selectable by the program of the Central Computer System, are:

- a. Move tape forward (read or write)
- b. Backspace
- c. Rewind.

The tape must be accelerated to the correct speed before either writing or reading can begin. A normal delay of 10 milliseconds is allowed (with an additional 40-millisecond delay allowed if writing or reading begins from the load point). Reading each word requires 320 microseconds with a 150-microsecond delay after the last word of a record. Writing requires an extra 150 microseconds, totalling to 300 microseconds after the last word. This time which includes the time taken by the tape in coasting to a stop, produces the blank space denoting the end of the record.

A backspace operation moves the tape back over one record, stopping the tape at the beginning of that record. If the tape is in write status when the backspace is given, the tape will move forward for 6 milliseconds and take another 25 milliseconds to shift into reverse before moving backward. The record is read backward but is not sent to the Central Computer System. Only the synchronizing track is read to sense the beginning of the record. When the beginning of the record is

TABLE 1-35. TAPE DRIVE UNIT CONTROLS

CONTROL	FUNCTION
START pushbutton	Places unit under program control.
RESET pushbutton	Places unit under manual control.
LOAD REWIND pushbutton	Loads tape onto transport mechanism, then rewinds to load point.
UNLOAD pushbutton	Unloads tape from transport mechanism.
REEL RELEASE pushbutton	Frees tape reels for loading.
File protection interlock	Prevents accidental writing on permanent tape records.

**TABLE 1-36. TAPE DRIVE UNIT CONTROL
LIGHT FUNCTIONS**

LIGHT	INDICATION
SELECT	Unit selected for reading or writing by the Central Computer System.
NOT READY	Unit mechanically unready for use.
FILE PROTECTION	Write circuit locked out of use.
NOT IN FILE AREA	Tape beyond end-of-tape mark or end-of-file mark has been read or written; tape heads not over usable area of tape.

reached, 1 millisecond elapses before the stop-tape signal is generated, and 2 milliseconds elapse before the mechanism shifts into forward. Twenty-five milliseconds are required to complete the shift.

The rewind operation causes the tape to rewind onto the feed reel until the photoelectric cell senses the load point. The tape stops, then shifts to forward status. The timing of each operation is shown in table 1-37.

A tape drive unit must satisfy both physical and program requirements before it can be operated under Central Computer System control. If the physical requirements listed in table 1-38 are satisfied, the tape drive unit is said to be ready. Each tape drive unit, if ready, sends a ready signal to the tape adapter unit.

If the program requirements are met by a tape drive unit, it is said to be prepared. The tape unit must be ready before it can be used by the Central Computer

System. If not prepared, it can be made prepared by the program. The conditions which can cause a tape drive unit to be not-prepared are shown in table 1-39.

A not-prepared condition due to the not-in-file-area condition is determined by the status of the not-in-file-area flip-flop. If this flip-flop is cleared, although the tape unit is not-prepared during rewind, the tape unit becomes prepared when rewind is completed. If the tape unit is not-prepared because the not-in-file-area flip-flop is set, it is still not-prepared after a rewind operation. A backspace operation, however, clears the not-in-file-area flip-flop, moves the tape back over one record, and leaves the tape drive unit in the prepared condition.

4.6.3 Tape Adapter Unit

The tape adapter units acts as an intermediary between the tape drive units and the Central Computer

TABLE 1-37. TAPE DRIVE UNIT OPERATION TIMING

OPERATION	STEP	TIME	
		(MILLISECONDS)	(MICROSECONDS)
Read or write; or write end-of-file:	1. Delay from load mark.	40	
	2. Delay from other than load mark.	10	
	3. Read or write each word.		320
	4. Delay after last word.		150
	5. Write end-of-record delay.		150
	6. Minimum total, one word.	10,620	(plus above 150)
Backspace; maximum total elapsed time to first word:	1. If in write status	73.9	
	2. If in read status.	61.9	
Rewind		Variable	

TABLE 1-38. READY VERSUS NOT-READY CONDITIONS

READY	DESCRIPTION	NOT-READY
START pushbutton pressed.	Unit, if ready, under Central Computer System control.	RESET pushbutton pressed.
Tape in both vacuum columns.	Transport mechanism loaded.	Tape not in both vacuum columns.
Tape intact.	Tape is complete through transport mechanism.	Tape broken.
Sensing light on.	Light for photoelectric sensing of reflective spots.	Sensing light off.
Power on.	Unit power switch on.	Power off.
A-c fuse OK.	Primary power applied.	A-c fuse blown.
D-c fuse OK.	Power applied to unit.	D-c fuse blown.
Door closed.	A-c interlock complete.	Door open.

TABLE 1-39. NOT-PREPARED CONDITIONS

CONDITION	CAUSE
During rewind.	Tape not yet rewound to load point.
Not-in-file-area.	End-of-tape mark senses or end-of-file mark just read or written.

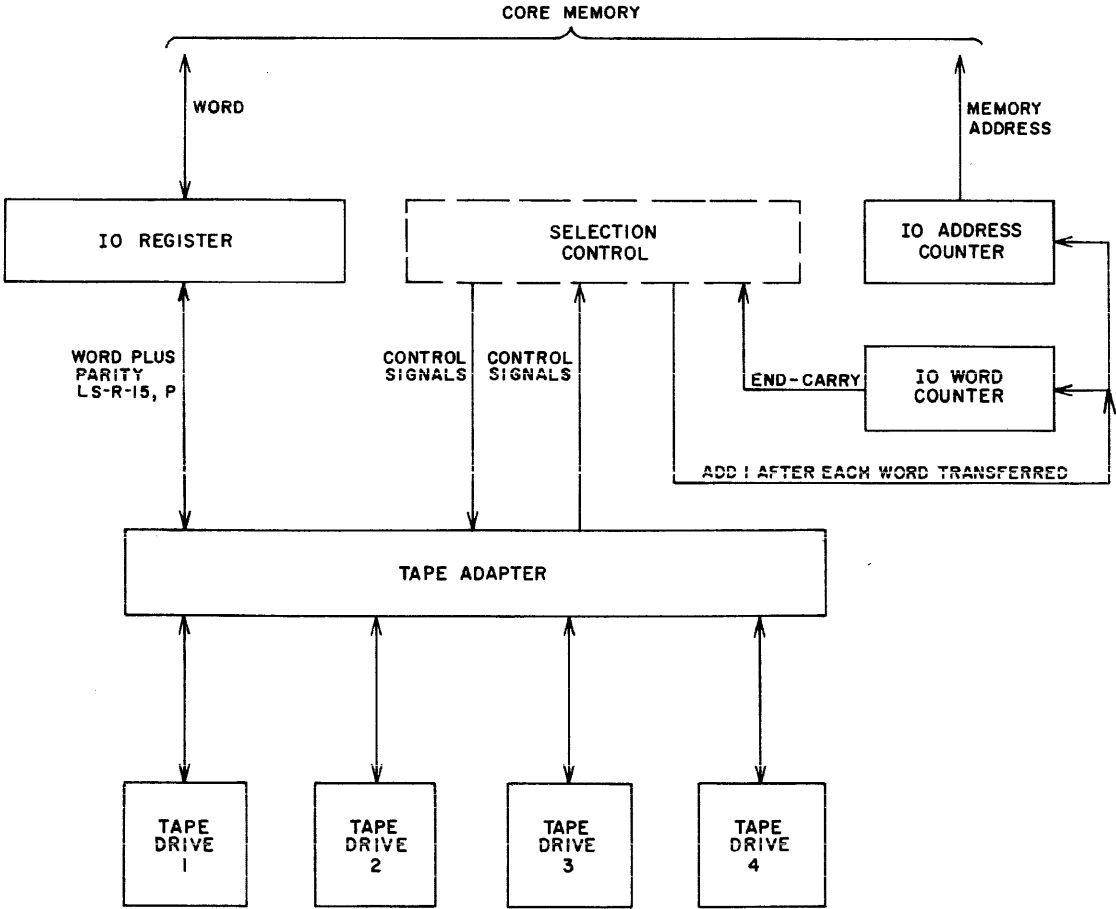


Figure 1-65. Tape Element Information Flow

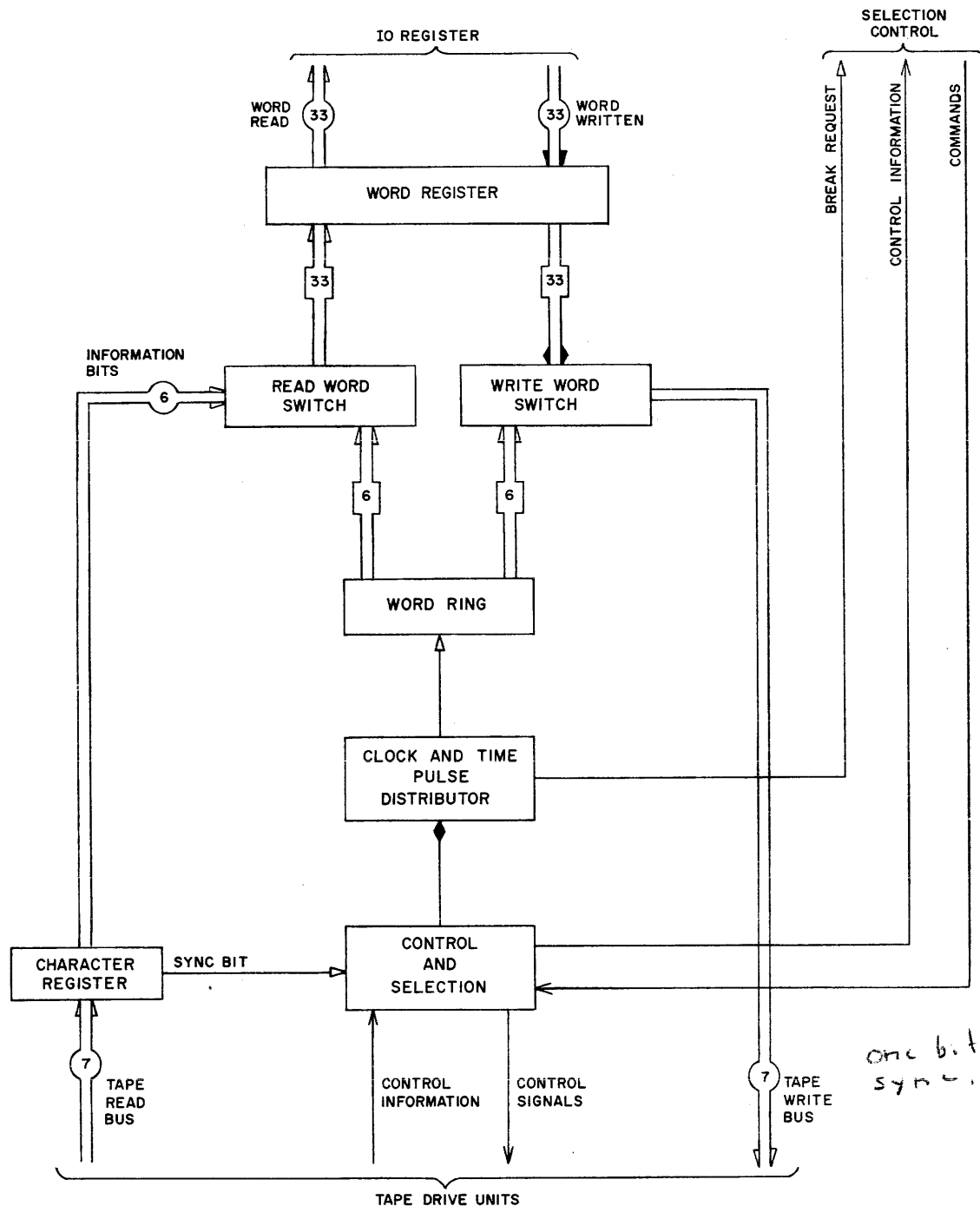


Figure 1-66. Tape Adapter Unit Information Flow

System. (See fig. 1-65.) Its operation may be considered in two parts, the execution of Central Computer System commands and the transfer of information to or from the Central Computer System.

In order to have any other instruction affect the tape element directly, the Central Computer System must first select the tape drive unit to be used. A *SEL* (u) instruction (where u specifies a tape drive unit) causes the selection and IO control element to send a select

signal to the tape adapter unit. The tape adapter unit then allows all subsequent instructions to affect only the selected tape drive unit until a new *Select* instruction deselects that unit. The assignment of unit addresses is accomplished by a patch panel in the tape adapter unit. All other tape element instructions cause the selection and IO control element to supply commands to the tape adapter unit, which in turn initiates and controls the operation of the selected tape drive unit.

Information transferred between the tape element and the Central Computer System passes between the IO register and the word register. Both registers can hold 32 bits plus a parity bit. Since the tape drive units handle only six information bits at a time, the word register must supply them with only six bits at a time during writing. The write word switch accomplishes this by splitting the contents of the word register into 6-bit characters and supplying the tape drive unit with one character at a time. (See fig. 1-66.) The clock and time pulse distributor section pulses the word ring which sequences the entire writing operation. The word ring causes the write word switch to gate bits R14 and R15 and the parity bit out of the word register first, then bits R8 through R13, bits R2 through R7, and so on, at the proper intervals for writing each character on the tape. After the bits of the last character are read from the word register, a break request starts the transfer of the next word from core memory to the word register via the IO register.

The reading operation of the tape adapter units is very similar to its writing operation. Each character

(six bits plus a synchronizing bit) read from the tape passes through shaping circuits into the character register. The sync bit is sent to the tape control and selection section and causes that section to start the clock and time pulse distributor section. This section pulses the word ring to gate the character bits through the read word switch into the word register. Again, the parity bit and bits R14 and R15 are read first. Each character is read into the word register by the same process, the word ring determining its correct placement in the word register. After the word is finally assembled in the word register, a break request initiates its transfer into core memory via the IO register.

4.6.4 Programming

4.6.4.1 Pertinent Instructions

Those instructions of direct pertinence to the tape element are listed in table 1-40.

The conditions listed in tables 1-38 and 1-39 determine the status of the selected tape drive unit. Each tape drive unit supplies the tape adapter unit with control information signals reflecting its status.

TABLE 1-40. TAPE ELEMENT INSTRUCTIONS

INSTRUC- TION	CODE	INSTRUCTION	REMARKS
<i>SEL</i> (11) ₈ through	—	<i>Select</i> (tape drive unit)	Deselects previously selected units.
<i>SEL</i> (16) ₈	—		
<i>BSN</i> (10) ₈	<i>x</i>	<i>Sense</i> (tapes not prepared)	Branch if not prepared.
<i>BSN</i> (11) ₈	<i>x</i>	<i>Sense</i> (IO unit not ready)	Branch if not ready.
<i>BSN</i> (17) ₈	<i>x</i>	<i>Sense</i> (tape parity)	Branch if incorrect.
<i>RDS</i>	<i>n</i>	<i>Read</i>	Reads <i>n</i> words into memory; sets IO interlock which is cleared at end-of-record pause; skips one record if <i>n</i> =0.
<i>WRT</i>	<i>n</i>	<i>Write</i>	Writes <i>n</i> words from memory; sets IO interlock which is cleared at end-of-record delay; <i>WRT</i> 0 illegal, can cause hangup.
<i>PER</i> (67) ₈	—	<i>Operate</i> (set prepared tapes)	Refer to 4.6.4.6.
<i>PER</i> (70) ₈	—	<i>Operate</i> (backspace)	Backs up one record; sets IO interlock which is cleared when beginning of record is reached and places unit in prepared condition.
<i>PER</i> (71) ₈	—	<i>Operate</i> (rewind)	Backs up to load point; IO interlock is on for 60 milliseconds; unit then not-prepared until end of rewind.
<i>PER</i> (72) ₈	—	<i>Operate</i> (write end-of-file)	Writes end-of-file mark; IO interlock set until completed.

(See fig. 1-66.) The variations of the *Sense* instruction can interrogate the tape adapter unit concerning the status of the selected tape drive unit. If that unit is prepared and ready, a reply is given and the next instruction is executed. If, however, that unit is not-prepared or not-ready, no signal is returned to the selection and IO control element which initiates a branch of program control.

The *Sense* (tapes not prepared) instruction, *BSN* (10)₈, senses for the conditions of tapes not-prepared. The *Sense* (IO unit not ready) instruction, *BSN* (11)₈, is a precautionary operation used to prevent hanging up the Central Computer System in an impossible IO operation involving a not-ready unit. Since a not-ready condition requires operator intervention to clear the condition, the *Sense* (IO unit not ready) instruction could branch program control to a *Program Stop* (*HLT*) instruction, calling attention to the not-ready condition and allowing the operator to clear it. The *Sense* (tape parity) instruction, *BSN* (15)₈, will check the parity of information transferred either to or from the tape element. If the parity is incorrect, the program should branch to take cognizance of the incorrect information transfer.

The *RDS* *n* instruction starts the reading process. Information transfer continues until either the number of words specified in the instruction is transferred or until an end-of-record gap is reached. In either case, the tape continues to move past the reading heads until the end-of-record gap is reached, stopping the tape drive unit and sending a disconnect signal to the selection and IO control element.

The *WRT* *n* instruction starts the writing process. After the *n*th word has been written, the end-of-record gap is produced as the tape stops and a disconnect signal is returned to the selection and IO control

element. Each *Write* instruction produces a separate record on the tape. Writing normally starts from the load point. A write operation can continue past the end-of-tape mark. However, after the *n*th word of that operation is written, the tape drive unit becomes not-prepared, preventing a new writing operation from starting. A *WRT 0* instruction is illegal. The four codes for the *Operate* instruction in table 1-40 are discussed in specific programs.

4.6.4.2 General Rules

If the contents of a reel of tape are to be preserved, the file protection interlock should be activated by removing the file protection ring on that reel. Writing on that reel is thus positively prevented.

~~To prevent hanging up the Central Computer System, a read, write, or write end-of-file operation should be preceded by a *Sense* (IO unit not ready) instruction, *BSN* (11)₈. The branch provided could be to a *Program Stop* (*HLT*) instruction, thereby allowing the operator to ready the selected unit.~~

The *Sense* (tapes not prepared) instruction, *BSN* (10)₈, permits a branch of program control after an end-of-file mark has been written or read, or the end-of-tape mark reached. In either case, some operation to prepare that tape unit is required before it can be programmed for reading or writing. If, after reading a complete file, the reel is to be rewound and then used for writing, the *Sense* (tapes not prepared) instruction may be used with a branch back to itself to delay the transfer of information to the tape until the rewind is completed. A read operation may not follow a write operation without special programming precautions.

4.6.4.3 Read Operation

The program sequence required for a read operation is shown in table 1-41.

TABLE 1-41. PROGRAM FOR READ OPERATION

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>SEL</i> (<i>u</i>)	—	Specifies which unit is to be used.
2	<i>LDC</i>	<i>x</i>	Loads IO address counter with address <i>x</i> in core memory to which first word is transferred.
3	<i>BSN</i> (11) ₈	<i>y</i>	Branches to location <i>y</i> if not ready.
4	<i>BSN</i> (10) ₈	<i>z</i>	Branches to location <i>z</i> if sensing end-of-file or end-of-tape.
5	<i>RDS</i>	<i>n</i>	Reads either <i>n</i> words or one record from tape.
<i>y</i>	<i>HLT</i>	—	Allows operator to ready tape unit.
<i>z</i>			Branch program depends on whether sensing for end-of-file or end-of-tape (refer to table 1-44).

The first two instructions of table 1–41 must be given in the sequence shown to allow settling before sensing for not-ready. Steps 3, 4, and 5 must be given consecutively. Step 5, the *Read* instruction, turns on the IO interlock, which remains on until the end-of-record gap is read. Each *Read* instruction will therefore pass one record through the reading station. If n is greater than the number of words in the record, the IO word counter will not contain 0 when the read operation ends. If n is less than the number of words in the record, only n words will be transferred to core memory although the entire record passes the reading heads; the words beyond the n th word in the record being read are lost. If n is 0, a record is skipped. The IO interlock is cleared and tape movement is stopped only when the end-of-record gap is read by the tape heads whereas the break requests which initiate word transfers to core memory are not honored after the IO word counter reaches 0. If n specified in the *Read* instruction is less than the number of words in the record being read as the result of a program error, that error can be rectified by backspacing through the record, then executing another *Read* instruction with n equal to, or greater than, the number of words in the record.

A second *Read* instruction may follow step 5 with out any intervening instructions. The execution of the second *Read* instruction will be delayed by the IO interlock until the first *Read* instruction is completed. A complete file can be read from the tape by programming the sequence of instructions shown in table 1–41 and including as many *Read* instructions as there are records within the file.

If accuracy of transfer is important during tape reading, two methods of detecting tape parity errors are available. One method uses the TAPE PARITY

switch on the duplex maintenance console. With this switch in the ACTIVE position, the occurrence of a parity error during tape reading can halt both the Central Computer System and the tape unit, generally not at an end-of-record gap. To reread the record in which the parity error occurred, a backspace operation must be executed before the *Read* instruction which initiates rereading.

The second method of detecting parity errors during tape reading makes use of the *Sense* (tape parity) instruction, *BSN* (17)₈. This instruction can be used to provide a branch of program control if a parity error is detected. The manner in which the branch is used is at the discretion of the programmer. The branch might provide for a backspace operation, then a read operation to reread the record in which the parity error was detected. The branch might also provide for a cumulative count of parity errors detected, with a resultant branch to a *Program Stop* (*HLT*) instruction if the number of parity errors exceeds some arbitrary number.

4.6.4.4 Write Operation

The program sequence required for a write operation is shown in table 1–42.

Writing on a reel of tape should start at load point. Therefore, the first write operation should be preceded by a rewind operation if it is possible that the reel is not at load point.

Each *Write* instruction causes the writing of a record containing n words onto the tape. The IO word counter steps to 0 as the n th word is written, sending a stop signal to the tape drive unit. However, the tape drive unit continues erasing for 300 microseconds after the n th word is written, producing the end-of-record gap and clearing the IO interlock.

TABLE 1–42. PROGRAM FOR WRITE OPERATION

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>SEL</i> (u)	—	Specifies which tape unit is to be used.
2	<i>LDC</i>	x	Loads IO address counter with address x in core memory from which first word is transferred.
3	<i>BSN</i> (11) ₈	y	Branches to location y if not-ready.
4	<i>BSN</i> (10) ₈	z	Branches to location z if sensing end-of-file or end-of-tape.
5	<i>WRT</i>	n	Writes n words in one record on tape.
y	<i>HLT</i>	—	Allows operator to ready tape unit.
z			Branch program depends on whether sensing for end-of-file or end-of-tape.

TABLE 1-43. PROGRAM FOR BACKSPACE OPERATION

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>BSN</i> (14) ₈	1	Branch to self, if on, to delay next instruction.
2	<i>PER</i> (70) ₈	—	Move tape backward over one record.
3			As program requires, generally read or write operation.

It is possible but not desirable for two *Write* instructions to be programmed with no intervening instructions since the second instruction is not executed until the first is completed and the IO interlock cleared. However, since the first *Write* instruction may carry writing beyond the end-of-tape mark, making the tape unit not-prepared, the second *write* instruction (and each succeeding *Write* instruction) should be preceded by the instruction listed as step 4 in table 1-42. The Central Computer System will then not be hung up on an impossible IO operation.

Parity checking may be performed during a writing operation but is useful only during reliability or diagnostic programs that test core memory. Accordingly, it is not discussed in this Section.

The last record on a reel of tape should be followed by an end-of-file word unless the end-of-tape mark has been passed. The end-of-file word is written by an *Operate* (write end-of-file instruction, *PER* (72)₈, which writes the three end-of-file bits in a one-word record. Since execution of the *Operate* (write end-of-file) instruction is not delayed by the IO interlock being on, it is imperative that this instruction be preceded by a *Sense* (IO interlock) instruction, *BSN* (14)₈, which can branch back to itself to delay the write end-of-file operation until the IO interlock is cleared. The *Operate* (write end-of-file) instruction sets the IO interlock until the end-of-file word is written, then puts the tape drive unit in the not-prepared condition. Writing an end-of-file word (if the tape has been stopped) requires approximately 10.6 milliseconds. A rewind operation generally follows the reading or writing of an end-of-file word.

4.6.4.5 Backspace

If it is desired to read a record just written or, for example, to rewrite the last record written because of a parity error, a backspace operation can be performed. The programming sequence required for a backspace operation is shown in table 1-43.

Since the execution of the *Operate* (backspace) instruction is not delayed by the setting of the IO interlock, the IO interlock must be sensed to prevent

the backspace operation from interfering with an IO operation in progress. The *Operate* (backspace) instruction initiates three actions. In addition to backspacing the tape over one record, the instruction sets the IO interlock while the backspace operation is being executed and puts the tape drive unit in the prepared condition if it was not-prepared before the backspace operation was executed. Thus, the backspace operation need not be followed by the set prepared tapes operation when it is desired to write more records into a file by backspacing after reading the end-of-file or to read a just-written record which carried past the end-of-tape mark by backspacing and then reading the record.

Although reading or writing an end-of-file or sensing the end-of-tape mark puts the tape drive unit in the not-prepared condition by setting the not-in-file-area flip-flop, the backspace operation clears this flip-flop and thus puts the unit in the prepared condition.

4.6.4.6 Rewind

A rewind operation moves the tape back to the load point. Since its execution is not affected by the setting of the IO interlock, the IO interlock should be sensed before the instruction to rewind is given, thus preventing interference with an IO operation in progress. The *Operate* (rewind) instruction holds the IO interlock on only long enough to switch the relays necessary to initiate rewind (approximately 60 milliseconds). At the end of the 60 milliseconds, the IO interlock is cleared, but the tape unit becomes not-prepared during rewind. Unlike the backspace operation, the rewind operation does not clear the not-in-file-area flip-flop. If an end-of-file word was read or written or an end-of-tape mark was read before the rewind operation, that operation must be followed by the *Operate* (set prepared tapes) instruction, *PER* (67)₈, to clear the not-in-file-area flip-flop. If this instruction is programmed along with the rewind, the tape unit will be in the prepared condition when rewind is completed.

If all the information on a reel of tape is read, using an iterative version of the program sequence given in table 1-41, and if it is desired to use the same tape reel to write new information, then, when program

TABLE 1-44. PROGRAM FOR REWIND OPERATION AFTER
READING END-OF-FILE

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>BSN</i> (14) ₈	1	Prevents interrupting IO operation.
2	<i>PER</i> (71) ₈	—	Moves tape back to load point, sets IO interlock for 60 milliseconds.
3	<i>LDC</i>	<i>x</i>	Loads counter with address <i>x</i> in core memory from which first word will be transferred and delays subsequent steps for 60 milliseconds.
4	<i>PER</i> (67) ₈	—	Clears not-in-file-area flip-flop set by reading end-of-file.
5	<i>BSN</i> (10) ₈	5	Branches to self until rewind completed and tape unit prepared.
6	<i>WRT</i>	<i>n</i>	Writes <i>n</i> words starting at load point.

step 4 of that sequence senses the end-of-file, the program could branch to a sequence of instructions similar to that of table 1-44.

Step 5 is used to delay the writing operation until the rewind is completed. The *Write* instruction would not be delayed by the IO interlock, which remains on only for 60 milliseconds after the rewind operation is initiated. The *Sense* (tapes not prepared) instruction branching back on itself will provide for the necessary delay of subsequent program steps until the tape reaches the load point. However, the *Sense* (tapes not prepared) instruction will not cause a branch if programmed immediately after the *Operate* (rewind) instruction, since the tape unit does not become not-prepared until shortly before the IO interlock is cleared (60 milliseconds after the rewind operation is initiated). The program given in table 1-44 meets this problem by including the *LDC* instruction to delay the succeeding program steps until the IO interlock is cleared. A rewind operation should generally be programmed before a reel change and prior to the start of operation with a tape drive unit.

4.7 IO REGISTER

The IO register may be used as a miscellaneous IO unit for the purpose of clearing registers in core memory. Essentially, this clearing action is accomplished by loading the IO register with positive zero (0.00000)₈ and then executing a series of break-in cycles. During these break-in cycles, the 0's in the IO register are transferred to the memory buffer register and written into specified locations in core memory. By this means, all or a specific portion of the memory locations in core memory may be cleared of extraneous information.

The program shown in table 1-45 will clear a specific number of memory locations in core memory by utilizing the IO register as an IO unit. This program will clear a number of memory locations starting with the address given by the *LDC* instruction. Thus, if *n* memory locations are to be cleared and the IO address counter is loaded with the address *x*, the Central Computer System will proceed to clear core memory locations *x* through *x + n - 1*.

A break is requested during the second program time (PT₂ 0 through 6) cycle of the *Read* instruction. As a result, the Central Computer System will stop its arithmetic operations at the end of this PT cycle and institute a break-in cycle. During this break-in cycle, the IO register is cleared and its contents transferred to the memory buffer register for storage in the memory location specified by the *LDC* instruction. Since the IO register is cleared, this stores positive zero (0.00000)₈ in this memory location, thus effectively clearing it. A continuous series of these break-in cycles will be executed until the number of memory locations specified by the *Read* instruction have been cleared. At this time, the IO interlock, which was turned on by the *Read* instruction, is turned off and the Central Computer System proceeds with the execution of its program.

Although core memory is usually completely cleared of all information by depression of the CLEAR MEMORY pushbutton on the duplex maintenance console, the clearing of all 8,192 memory locations may also be accomplished by the program given in table 1-46. Execution of this program by the Central Computer System will result in the execution of 8,192 break-in cycles, which transfer all 0's to the memory locations

of core memory. A final *Select* instruction after the *Read* instruction holds up subsequent instructions until all of the 8,192 break-in cycles are completed. Since all memory locations are now cleared (unless this program is in test memory and contains further instructions),

the next instruction will be a *Program Stop* instruction, causing the Central Computer System to halt unconditionally. Thus, this program completely clears core memory of all information and then causes the Central Computer System to halt.

TABLE 1-45. PROGRAM TO CLEAR SPECIFIC MEMORY LOCATIONS

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>LDC</i>	x	Loads IO address counter with address x of first memory location to be cleared.
2	<i>SEL</i> (04) ₈	—	Selects IO register.
3	<i>RDS</i>	n	Clears n memory locations in n break-in cycles; requests break cycle.

TABLE 1-46. PROGRAM TO COMPLETELY CLEAR CORE MEMORY

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>LDC</i>	0.00000 ₈	Places first core memory address in IO address counter.
2	<i>SEL</i> (04) ₈	—	Selects IO register.
3	<i>RDS</i>	0.20000 ₈	Clears all 8,192 ₁₀ memory locations.
4	<i>SEL</i> (u)	—	Holds up subsequent instructions.

SECTION 5

APPLICATIONS

5.1 INTRODUCTION

This Section discusses some programs for a number of operations that the Central Computer System can perform. The first few examples are relatively simple. They serve to clarify basic operations. Each sample program is preceded by a brief discussion of the operation to be performed. The sample programs themselves are given in the normal form of a printed-out program, specifying the location of each instruction and the storage locations for initial data and for the result of the program.

In a program involving arithmetic operations, the arithmetic element limits an operation to the manipulation of only two operand words at a time; one operand word is held in the accumulators and the other is brought from memory by the instruction designating the nature of the operation. However, to satisfy the dual nature of the arithmetic element, each operand word contains two separate numbers, one in each half of the operand word. Therefore, an arithmetic operation manipulates two pairs of numbers, a left pair and a right pair; this is done simultaneously without any necessary relation between the two pairs.

5.2 STRAIGHT-LINE PROGRAMMING

5.2.1 Basic Arithmetic Operations

5.2.1.1 Addition

The addition of two operands requires two distinct

steps. First, one operand containing two numbers is placed in the accumulators. Then, the other operand is added to the first, leaving the sums of the numbers in the accumulators. The result may then be left in the accumulators for immediate manipulation by a subsequent program or may be stored in core memory for later use. Mathematically, the addition may be described as follows:

$$\begin{array}{rcl} \text{First operand:} & n_l & n_r \\ \text{Second operand:} & m_l & m_r \\ \hline & n_l + m_l & n_r + m_r \end{array}$$

The program and the initial storage and result storage locations for this addition operation are given in table 1-47.

If more than two operands were to be added, another *ADD* instruction would be required for each extra operand before the *FST* instruction. Numerous other variations are possible, but this example is sufficient to demonstrate the basic program for addition.

5.2.1.2 Subtraction

The subtraction of one operand from another with both operands initially in core memory may be accomplished in one of two ways. The second operand may be subtracted from the first or the first operand may be made negative and the second operand added to it.

TABLE 1-47. PROGRAMMED ADDITION

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	10	Places n_l , n_r in accumulators.
2	<i>ADD</i>	11	Adds m_l , m_r to contents of accumulators.
3	<i>FST</i>	12	Stores sums at location 12.
4	<i>HLT</i>	—	End of program.
10	n_l	n_r	Initial data storage.
11	m_l	m_r	
12			Result storage.

The first method appears mathematically as follows:

$$\begin{array}{r} \text{First operand:} \quad n_l \quad n_r \\ \text{Second operand:} \quad m_l \quad m_r \\ \hline n_l - m_l \quad n_r - m_r \end{array}$$

The program for this operation is shown in table 1-48.

Note that the results stored at 12 are $n_l - m_l$, $n_r - m_r$. Comparison of this result with that produced by the program shown in table 1-49 reveals a significant difference.

This second subtraction program yields $m_l - m_l$, $m_r - n_r$ as the result stored in location 12. It should be apparent from the comparison of these two sub-

traction programs that the order in which operations are performed is sometimes significant to the result.

5.2.1.3 Multiplication

The multiplication of two operands is performed in two steps. However, the products so obtained must be further processed before they can be stored in core memory. Multiplication of two half-words yields a signed product of 30 significant bits, smaller in absolute value than either the multiplier or the multiplicand. (Refer to 3.4.2.) Further, two such products are obtained in the dual arithmetic element. In order to store such a result in core memory, the products must be rounded off to 15 significant bits and, in some cases, scaled up

TABLE 1-48. PROGRAMMED SUBTRACTION

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>CAD</i>	10	Places n_l , n_r in accumulators
2	<i>SUB</i>	11	Subtracts m_l , m_r from assumulators
3	<i>FST</i>	12	Stores result
4	<i>HLT</i>	—	End of program
10	n_l	n_r	Initial data storage
11	m_l	m_r	
12			Result storage

TABLE 1-49. ALTERNATE SUBTRACTION PROGRAM

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	<i>CSU</i>	10	Places n_l , n_r in accumulators
2	<i>ADD</i>	11	Adds m_l , m_r to accumulators
3	<i>FST</i>	12	Stores result
4	<i>HLT</i>	—	End of program
10	n_l	n_r	Initial data storage
11	m_l	m_r	
12			Result storage

in value to retain sufficient precision. Thus, a multiplication operation is accomplished by a program of the type shown in table 1–50.

The factor by which the *SLR* n instruction may scale up the magnitude of the products obtained by multiplication can be determined if the range of values of multiplier and multiplicand are known. A rough calculation with those values yielding the largest product indicates the largest scale factor that will not destroy the significance of the product. For example, if $n < 2^{-8}$ and $m < 2^{-5}$, then $nm \leq 2^{-13}$ and can be scaled up in magnitude by a maximum factor of 2^{12} . However, certain other facts must be considered when deciding upon a scale factor. The scale factor may be modified to include the effect of a prior or subsequent operation on the numbers. In addition, the size of the scale factor may be limited by the desired scale for presentation of the final result of calculation. In general,

the choice of scale factor is determined by these three considerations:

- The desire to preserve maximum precision at a given point in a calculation
- The effect of prior or subsequent operations on the scale of the number
- The desired scale factor of the final result

5.2.1.4 Division

The division of two operands requires two instructions, as do all other arithmetic operations. As in the case of multiplication, further processing is required after the quotients are obtained to put them in a form suitable for storage in core memory. In addition, the dividends may require manipulation prior to the division operation proper to meet the requirement that the dividend be smaller in absolute magnitude than the divisor. If this restriction is observed, the *DVD* x

TABLE 1–50. MULTIPLICATION PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	10	Places $-n_1$, $-n_r$ in accumulators
2	<i>MUL</i>	11	Multiplies m_1 by n_1 and m_r by n_r
3	<i>SLR</i>	n	Scales products up in magnitude by 2^n and rounds them off to 15 significant bits
4	<i>FST</i>	12	Stores products
5	<i>HLT</i>	—	End of program
10	n_1	n_r	Initial data storage
11	m_1	m_r	
12			Result storage

TABLE 1–51. DIVISION PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	10	Places n_1 , n_r in accumulators
2	<i>DSR</i>	n	Scales dividends down to magnitude smaller than divisors
3	<i>DVD</i>	11	Divides scaled n_1 by m_1 , scaled n_r by m_r
4	<i>SLR</i>	15	Moves quotients into accumulators and rounds them off to 15 significant bits
5	<i>FST</i>	12	Stores result
6	<i>HLT</i>	—	End of program
10	n_1	n_r	Initial data storage
11	m_1	m_r	
12			Result

instruction leaves an unsigned 16-bit quotient in each of the B registers and a signed 15-bit remainder in each of the accumulators. (Refer to 3.4.3.) If the quotients are to be stored, they must be shifted into the accumulators to join their sign bits and rounded off to 15 significant bits. The program shown in table 1-51 accomplishes this result.

As shown, scaling in division should be done prior to the DVD instruction. An attempt to use the SLR instruction requires an extra step, clearing the bits of the remainders from the accumulators to prevent their interpretation as significant bits of the quotients. In general, the amount of scaling to be done on the dividends can be determined by making a rough calculation with the limits on the values of divisor and dividend. The same overall considerations apply to scaling within a division operation as were described in connection with multiplication.

5.2.2 Combined Operations

5.2.2.1 Co-ordinate Conversion

An example of a combined operation can be given in terms of the radar co-ordinate conversion discussed mathematically in Chapter 1, 3.1.3. The equations set forth there for the conversion of a radar report on a target from polar co-ordinate form based on the radar

set to rectangular co-ordinate form based on some reference axes common to all radar sets are as follows:

$$X_t = R \sin \theta + X_r \quad Y_t = R \cos \theta + Y_r$$

Where R = range from radar set to target

θ = angle, measured clockwise, between radar north line and radar target line

X_r = X component of radar location measured on common reference

Y_r = Y component of radar location measured on common reference

X_t = X component of target location

Y_t = Y component of target location

The problem of finding $\sin \theta$ and $\cos \theta$ when θ is given is considerably simplified by preparing a table of sines and cosines in core memory arranged so that adding the represented value of θ to the address of the location in which $\sin 0$, $\cos 0$ is stored produces the address of the location containing $\sin \theta$, $\cos \theta$. Such a table can be prepared in advance if the minimum increment of measurement of θ is known. In addition, the problem of scaling can be eliminated if the scale factors for R and X_r , Y_r are suitably chosen. If these two assumptions are granted, the program for co-ordinate conversion can be written as shown in table 1-52.

TABLE 1-52. CO-ORDINATE CONVERSION PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	CAD	30	Places θ in right accumulator (and R in left accumulator)
2	3 CAD	100	Places $\sin \theta$, $\cos \theta$ in accumulators by a table look-up procedure
3	TMU	30	Multiplies $\sin \theta$ and $\cos \theta$ by R
4	SLR	0	Rounds off products to 15 significant bits
5	ADD	20	Adds X_r to $R \sin \theta$ and Y_r to $R \cos \theta$
6	FST	30	Replaces R, θ in memory with X_t , Y_t
7	HLT	—	End of program
20	X_r	Y_r	Initial constant data storage
30	R	θ	Initial data, then final result storage
100 through 100 + N	$\sin 0$ $\sin N$	$\cos 0$ $\cos N$	Sine, cosine table

The table look-up procedure accomplished by the instructions at locations 1 and 2 is dependent upon the presence of the table of sines and cosines in locations 100 through $100 + N$. The instruction at location 1 places θ in the right accumulator for use in indexing the instruction at location 2. This second instruction, 3 *CAD* 100, obtains the sine and cosine of θ if not modified by indexing. The indexing of this instruction causes the transfer of the word at location $100 + \theta$ which, by prearrangement, contains $\sin \theta$, $\cos \theta$. (Refer to 3.2.3.)

The *TMU* 30 instruction allows the simultaneous generation of $R \sin \theta$ and $R \cos \theta$ in the two accumulators. It is followed by the *SLR* 0 instruction to round off the product to 15 significant bits. This instruction could also be used to scale the products, if that were necessary. After the addition of X_r , Y_r , the result is stored in the location from which R , θ was taken. This procedure can be used if R , θ is no longer needed for subsequent calculations and if it is desired to save space in core memory.

5.2.2.2 Evaluation of a Function

The evaluation of the function $y = ax^2 + bx + c$

can be used as another example of an operation requiring the use of several basic operations. The programming of such an operation requires some investigation of the scaling to be performed as well as of the arithmetic operations required. The specific scaling operations will be pointed out in this discussion without being explained in detail. However, some mention must be made of one basic rule of scaling: two numbers to be combined by addition or subtraction must be scaled identically if the results are to be valid. In addition, the end result of scaling, the retention of maximum precision information after processing of that information, must always be kept in mind.

The equation to be evaluated, $y = ax^2 + bx + c$, can be rewritten to facilitate coding in the form $y = x(ax + b) + c$. The calculation must therefore generate the following terms: ax , $ax + b$, $x(ax + b)$, and finally $x(ax + b) + c$. The coded program for this calculation is given in table 1–53. For convenience in following this program, only the left half of the arithmetic element is considered.

This program could be modified to use the right arithmetic element in evaluating y for another value

TABLE 1–53. FUNCTION EVALUATION PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	20	Places a in accumulator
2	<i>MUL</i>	23	Multiplies a by x
3	<i>DSR</i>	n	Scales ax in accumulator-B register down by 2^{-n} to match scale of b
4	<i>SLR</i>	0	Rounds off scaled product to 15 significant bits
5	<i>ADD</i>	21	Adds b to properly scaled ax
6	<i>MUL</i>	23	Multiplies sum by x
7	<i>SLR</i>	m	Scales second product up by 2^m to match scale of c and rounds off to 15 significant bits
10	<i>ADD</i>	22	Adds c to properly scaled second product, yielding y
11	<i>FST</i>	30	Stores result y
12	<i>HLT</i>	—	End of program
20	a	—	Initial data storage
21	b	—	
22	c	—	
23	x	—	
30			Result storage

of x , thus performing two evaluations simultaneously. In addition, the program could be repeated without this modification to calculate the value of y for each of several values of x .

5.2.3 Logical Operations

As an example of a logical operation (that is, a nonarithmetic operation), consider the problem of finding the number of largest absolute magnitude in a group of four numbers. The program shown in figure 1-67 will accomplish this result. Each comparison is made by the *DIM* x instruction. (Refer to 3.3.3.) In the program given, only the left arithmetic element is used.

The number sorting program is designed to provide two alternatives after each comparison, the choice of an alternative being determined by the result of the comparison. The program can be modified to obtain the smallest rather than the largest number in the group simply by interchanging the alternatives; i.e., by interchanging the *DCL* 20₈ and *CAD* x instruction in the parallel paths of the program.

This type of program, although it uses arithmetic operations, is described as a logical operation in the sense that its objective is logical rather than arithmetic. The result of this program, finding the largest number within a group, differs from the result of an arithmetic program wherein a product or the value of a function is found.

5.2.4 IO Operations

An examination of the programs given in Section 4 reveals a basic pattern in all IO programs; every IO program uses the instructions *SEL* (u) or *SDR* (u) r , *LDC* x , and *RDS* (i) n or *WRT* (i) n to set up a block transfer of words between a single IO unit and the Central Computer System. However, these instructions do not comprise the IO program proper. Instead, they set parameters with the basic IO program designed into the Central Computer System. Once these parameters are set and the IO program initiated, the IO operation is completed without any need for further programmed instructions. The IO operation has priority over internal program operations in that a break cycle is performed rather than an internal machine cycle whenever a choice exists. In this way, internal machine operations can be continued within the interval in which an IO operation is performed. The internal machine operations are interrupted only when an IO word transfer is executed and then only for one machine cycle at any one time. In addition, provision is made to prevent changing the parameters of the IO program while an IO operation is in progress; this is done by making the execution of any IO class instruction dependent upon the setting of the IO interlock. The IO interlock is turned on by the *RDS* or *WRT* instruction which initiates a given

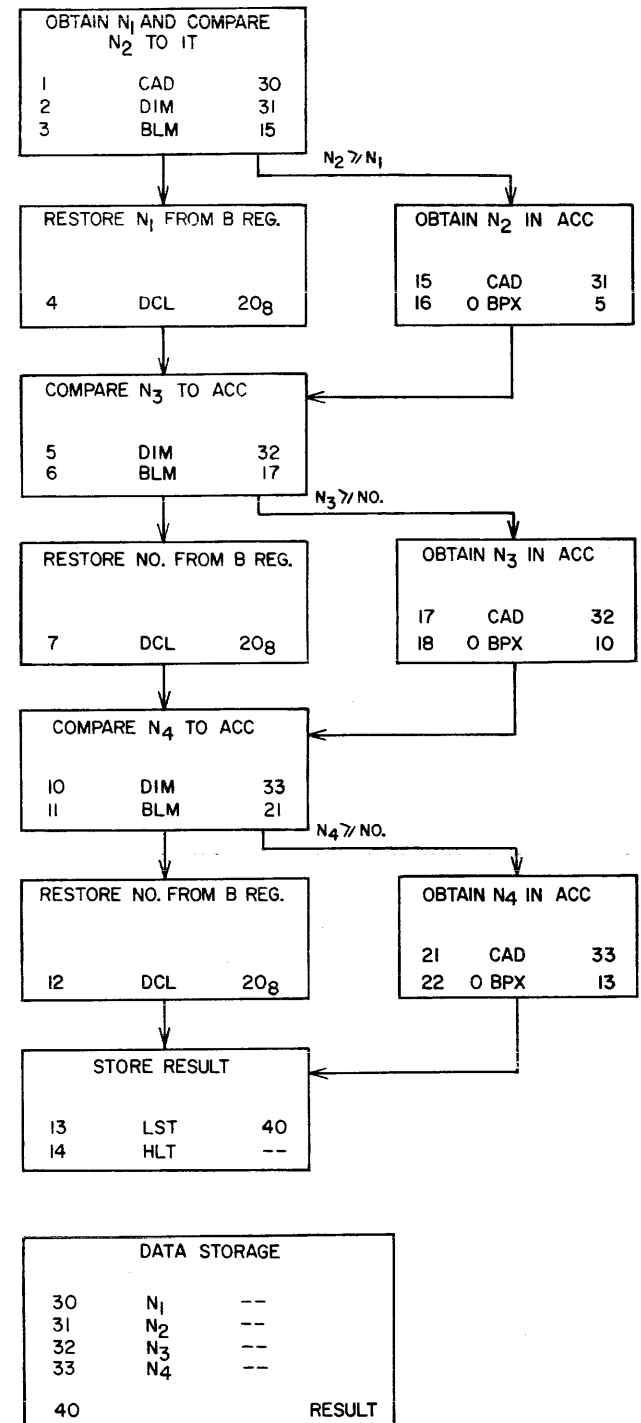


Figure 1-67. Number Sorting Program,
Flow Chart Form

IO operation and is turned off by the completion of the IO operation.

The parameters which must be set to execute an IO operation include the selection of the IO unit (either those discussed in Section 4 or the drum fields whose operation is described in Chapter 4), the designation of the register in core memory which is to receive or

supply the first word to be transferred in the IO operation, and the specification of the direction of transfer and of the number of words to be transferred. The exact sequence in which the first two parameters are set is not usually important. However, their setting must precede the RDS or WRT instruction which sets the last two parameters and initiates the IO operation. Initiation of the IO operation sets the IO interlock until completion of the operation. No IO class instructions can be performed while the IO interlock is set. Further, the programming of an IO class instruction while the IO interlock is on will prevent the execution of any instructions until the IO interlock is cleared and the programmed IO class instruction is executed.

The IO interlock is turned on by an RDS or WRT instruction. It is normally cleared either by the IO word counter stepping to zero (indicating the transfer of all n words) or by a disconnect signal from the IO unit indicating that all available words have been read or all available registers have been written. However, the IO interlock will never be turned off automatically if any of the following conditions exist:

- a. A nonexistent IO unit is selected.
- b. The selected IO unit is unable to deliver or accept words due to lack of power, failure of any component, or inoperative condition caused by a previous operation.
- c. The direction of transfer specified by an RDS or WRT instruction is impossible for the selected IO unit (for example, reading from the line printer or writing at the card reader).

If one of these conditions does exist, the Central Computer System will hang up on the next IO class instruction.

Hangups can be guarded against by including, in each IO program, instructions which check the readiness or preparedness of the selected IO unit. The instructions *Sense* (IO unit not ready), $BSN (11)_8 x$, and *Sense* (tapes not prepared), $BSN (10)_8 x$, can be used where appropriate. (Refer to Sec. 4.) In addition, the IO interlock can be sensed by a $BSN (14)_8 x$ instruction which causes a branch of program control if the IO interlock is on. A program which can hang up the Central Computer System may contain provision for sensing the IO interlock and branching, if it is on, to an *Operate* (clear IO interlock) instruction, $PER (27)_8$. The $PER (27)_8$ instruction is intended for use only when the IO interlock is not likely to be cleared automatically; its execution during an IO operation may produce results which are both unpredictable and undesirable.

The number of words to be transferred in any one IO operation is determined by the contents of the right half-word of the RDS or WRT instruction which initiates the operation after the right half-word is modi-

fied by indexing of the instruction, if any. This right half-word must be interpreted as a 16-bit binary integer rather than as a signed 15-bit binary fraction. If, for example, the right half-word after any indexing is 1.77777_8 (negative zero), the number of words to be transferred is $177,777_8$ or $65,535_{10}$ words, considerably more than the capacity of core memory. At the start of a transfer, the right half-word of the RDS or WRT instruction is indexed, if this is called for, and placed in the IO word counter in complement form. A 1 is immediately added to the least significant bit position to sense for an RDS 0 or WRT 0 instruction. At this point, before any words have been transferred, the IO word counter contains a 16-bit integer in 1's complement form with negative sign understood; this integer is the 1's complement of a number 1 less than the number of words specified for transfer by the instruction which loaded the IO word counter. Stated mathematically, the IO word counter contains $-n + 1$ just before the first word is transferred.

As each word is transferred during an IO operation, a 1 is added to the least significant bit of the IO word counter, making its contents less negative by 1 after each transfer. After all words except the last word have been transferred ($n - 1$), the IO word counter contains negative zero; any number added to its complement produces negative zero as the result. As the last word is transferred, the IO word counter generates a disconnect pulse which, in most cases, terminates the IO operation.

In certain cases, such as reading from a drum field by status or status identification, reading a magnetic tape record, or writing on a drum field by status, the number of words to be transferred is limited either by the number of words available for reading or by the number of drum registers available for writing. In these cases, the IO unit generates a disconnect pulse when it has supplied all the words it has available or when it has accepted all the words it has space to write. If t words are transferred in such an operation and $t < n$, then the IO word counter will contain the quantity $-n + t + 1$ when the IO operation is terminated. The number of words already transferred, t , can be found by adding $n - 1$ to the contents of the IO word counter. This is made possible by the *CSW* instruction, which places the contents of the IO word counter in the right accumulator register.

Certain special considerations are necessary when attempting to compute the number of words written onto a drum field using the *CSW* instruction after the IO operation is terminated. Writing onto a drum field requires two break-out cycles to move two words into writing position before any word is actually written onto the drum field. However, the IO word counter is not stepped for these two break cycles. For this reason,

the IO word counter contains $n + 1$ just before the first transfer is completed. Similarly, at the end of a writing operation in which t words are actually written onto a drum field, the IO word counter contains $n + t + 1$. In addition, if an IO transfer is interrupted by an alarm, the contents of the IO word counter may have no predictable relation to the number of words transferred and, therefore, cannot be used to determine that number.

Like the IO word counter, the IO address counter is stepped by 1 as each word is transferred in an IO operation. At any point in an IO operation, the IO address counter contains the core memory address from which or into which the next word is to be transferred. At the end of an IO operation, the IO address counter contains an address one higher than the last address involved in the IO operation. In the case of writing onto a drum field, the address in the counter is three higher than the address of the last word written onto the drum field; two words are in writing position but are not written onto the drum field.

The IO address counter contains 16 flip-flops connected so that only bit positions R2 through R15 operate as a counter. The IO address counter thus counts modulo $40,000_8$. In any transfer of more than $20,000_8$ ($8,192_{10}$) words, the transfer will be to or from core memory and test memory alternately. The IO address counter can be stepped beyond the memory unit selection code for core memory to the code for test memory and from test memory will step back to core memory. However, the IO address counter cannot be stepped beyond the test memory code to select the clock register nor, if the clock register selection code is placed in the IO address counter by an *LDC x* instruction, can it be stepped from the clock register code back to core or test memory. (Refer to 2.4.)

In certain applications, it may be desirable to execute a transfer to or from a single register in core or test memory. This can be done by executing the *Operate* (lock address counter) instruction, *PER* (75)₈, before the *RDS* or *WRT* instruction which initiates the IO operation. The IO address counter is not stepped during the operation; therefore, the same memory register is used for each transfer executed during the IO operation. The IO address counter is unlocked when the IO interlock is cleared at the end of the IO operation. The *PER* (75)₈ instruction is not to be given during an IO operation; i.e., while the IO interlock is on.

5.3 ITERATIVE PROGRAMMING

5.3.1 General

The programming examples discussed so far have all been straight-line programs; i.e., programs which proceeded in a straight line to accomplish a given result with a single item, or limited block, of data. If any

of these programs were to be used in processing more data in a straight-line manner, the programs would have to be lengthened. Those instructions which refer to data in core memory would be rewritten with new address halves and added to the original program. For example, in order to use the function evaluation program given in table 1-53 to evaluate the function for a second value of x , all nine instructions of the program would be written out again, changing only the address halves of those instructions referring to the value of x used in the calculation and those referring to the storage location for the result. The use of straight-line programming in this type of operation is inefficient in terms of memory space.

A digital computer capable of performing a branch of program control can be programmed iteratively as well as in a straight line. Iterative programming reuses a small group of instructions for each datum to be processed in the same manner by changing the address half of each instruction referring to the datum before that instruction is reused. The Central Computer System not only meets the requirement for iterative programming but exceeds it, having facilities for indexing and automatic address modification.

5.3.2 Indexing

The Central Computer System contains four registers (index registers 1, 2, 4, and 5) used solely for address modification in iterative programs. In addition, the right accumulator register can be used in certain types of address modification such as table look-up procedures. (Refer to 5.2.2.1.) When so used, the right accumulator register is identified as index register 3.

An indexable instruction (one whose address half can be modified by the contents of an index register) is indexed if its index indicator bits, L1 through L3, contain the binary equivalent of one of the octonary numbers from 1 through 5; when the instruction is executed, the contents of the index register whose number corresponds to the index indicator of the instruction is added to the address half of the instruction in the address register before the address half is used either to obtain an operand or to set a control register. (Refer to 3.2.3.) If the index indicator of an indexable instruction is 0, no address modification is performed.

The contents of index register 1, 2, 4, or 5 can be set by either of two instructions, *XIN n* or *XAC*. (The right accumulator register contents can be set or modified by so many instructions that no special instruction is necessary to set it for indexing purposes.) The *XIN n* instruction sets the index register specified by its index indicator to the value n . For example, *4 XIN 12304₈* sets index register 4 to the value 12304_8 . This instruction is most useful in setting an index register to a value known at the time the program is written or to a value calculated by a

previous program and left in core memory. The *XAC* instruction sets the index register specified by its index indicator to the value contained in the right accumulator register at the time the instruction is executed. For example, 2 *XAC* sets index register 2 to the value in the right accumulator register. This instruction is most useful in setting an index register to a value obtained as the result of some calculation. Neither of these instructions can be used to set the right accumulator register; i.e., an index indicator of 3 in either of these instructions invalidates them.

Once the contents of an index register are set, the index register may be used to modify the address halves of instructions referring to operands or storage locations as these instructions are executed in a program routine processing one item of data. When the processing of that item is completed, the contents of the index register may be reduced by an appropriate amount and a branch of program control executed to repeat the program routine for a new item of data by the use of a *BPX* (*s*) *x* instruction whose index indicator specifies the index register being used in the routine. (Refer to 3.7.4.3.) Since the *BPX* (*s*) *x* instruction causes a branch of program control each time it is executed until the specified index register becomes negative (because of the successive sub-

tractions of *s* from its contents), the number originally placed in the index register controls the number of repetitions of the iterative loop processing the several items of data.

The advantages of iterative programming over straight-line programming and the simplifications possible through the use of the indexing facilities available in the Central Computer System are shown by three different programs that can be written to perform the same operation.

5.3.3 Program Comparisons

5.3.3.1 Straight-Line Program

A straight-line program to find the sum of 10 numbers can be written as shown in table 1–54. The program uses 12 instructions in all.

The entire program can be executed in 248 microseconds (exclusive of the time required to load the program and the data to be processed). If more than 10 numbers are to be added, an *ADD x* instruction must be added for each extra number to be added. Therefore, if more than 10 numbers are to be added by a straight-line program, more memory space is required to store the program.

5.3.3.2 Nonindexed Iterative Program

A nonindexed iterative program can be written to add

TABLE 1–54. STRAIGHT-LINE ADDITION

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	30	Places first number in accumulator.
2	<i>ADD</i>	31	Adds second number to first number.
3	<i>ADD</i>	32	Adds succeeding numbers to sum of previous numbers in accumulator.
4	<i>ADD</i>	33	
5	<i>ADD</i>	34	
6	<i>ADD</i>	35	
7	<i>ADD</i>	36	
10	<i>ADD</i>	37	
11	<i>ADD</i>	40	
12	<i>ADD</i>	41	
13	<i>FST</i>	150	Stores result.
14	<i>HLT</i>	—	End of program.
30	<i>n</i> ₁	<i>m</i> ₁	Initial data storage.
through 41	<i>n</i> ₁₀	<i>m</i> ₁₀	
150			Result storage.

10 numbers. A program of this type requires fewer instructions but more execution time to duplicate the results of an equivalent straight-line program. However, the iterative program shown in table 1-55 can be used without requiring more instructions to add more than 10 numbers.

This program takes 678 microseconds to find the sum of 10 numbers. However, if more than 10 numbers are to be added, only the right half-word in location 147 need be changed to contain the address of the last number to be added.

Explanation of this program follows. The first instruction simply places the first number in the accumulator. The 0 *BPX* 4 instruction causes the program to enter the iterative loop contained in location 3 through 10, at the addition step. The first partial sum is produced by the *ADD* instruction and is stored temporarily by the *FST* instruction while the operations necessary to make the loop iterative are performed. The *AOR* 4 instruction modified the contents of location 4 by adding a 1 in the least significant position of the right half-word. Therefore, after the first execution of the *AOR* instruction,

the word in location 4 contains *ADD* 32. Thus, when the *ADD* instruction is repeated, it will obtain the third number from core memory rather than the second.

The *AOR* instruction leaves a duplicate of the right half-word in location 4 in the right accumulator. The *SUB* 147 instruction subtracts the last address from the right accumulator. (The operation of the left arithmetic element resulting from this instruction is unimportant since the next instruction ignores the left accumulator.) The *BRM* 3 instruction tests the sign of the right accumulator; if negative, a branch of program control to location 3 results. The *CAD* 150 instruction at location 3 takes the partial sum from temporary storage and replaces it in the accumulator in preparation for the next addition. (This instruction was bypassed by the 0 *BPX* instruction on the first iteration since it was unnecessary.) The iterative loop then executes the next addition and test as described, repeating itself until all of the additions are completed. The program then halts with the result in the proper storage location.

As the next to the last pass through the iterative loop is started, the *ADD* instruction contains 40 as its address

TABLE 1-55. NONINDEXED ITERATIVE PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	<i>CAD</i>	30	Places first number in accumulator.
2	0 <i>BPX</i>	4	Unconditionally branches to first addition.
3	<i>CAD</i>	150	Places last partial sum in accumulator.
4	<i>ADD</i>	31	Adds next number to partial sum (or to first number in first iteration).
5	<i>FST</i>	150	Places partial sum (or final sum in last iteration) into storage.
6	<i>AOR</i>	4	Adds 1 to address half of <i>ADD</i> instruction in location 4 and leaves modified address half in right accumulator.
7	<i>SUB</i>	147	Compares new address half to final data address.
10	<i>BRM</i>	3	Tests for completion of addition; if not complete, branches to repeat addition sequence with next number.
11	<i>HLT</i>	—	End of program.
30 through 41	n_1 n_{10}	m_1 m_{10}	Initial data storage.
147	—	41	Address of last number to be added.
150			Temporary and final result storage.

half. After the addition is performed and the partial result stored, the address half of the *ADD* instruction is changed to 41 by the *AOR* instruction. When the *SUB* instruction is executed, the result in the right accumulator is negative zero (1.77777_8) since any number added to its complement gives negative zero as the result. The *BRM* instruction recognizes the result as negative and therefore causes another repetition of the iterative loop. The last addition is performed and the final sum is stored in location 150. The *AOR* instruction changes the address half of the *ADD* instruction to 42. When the *SUB* instruction is executed, the result in the right accumulator is 0.00001_8 , a positive number. Therefore, the *BRM* instruction does not cause a branch and thus allows execution of the *HLT* instruction which ends the program.

5.3.3.3 Indexed Iterative Program

An indexed iterative program can also be written to add 10 numbers. (Refer to table 1–56.) This type of program requires only 6 instructions which, with repetition, takes 204 microseconds to produce the desired result. Thus, this program is shorter and faster than either of the other two programs.

This iterative program can be used in processing more than 10 numbers with the same number of instructions; only the right half of the *XIN* instruction must be changed. A description of the entire program follows for changed. A description of the entire program follows for an understanding of the exact significance of this right half-word.

The 1 *XIN* 10_8 instruction sets index register 1 to the

decimal value 8, one less than the necessary number of repetitions of the *ADD* instruction. The *CAD* instruction moves the first number into the accumulator. The 1 *ADD* 31_8 adds a number to the contents of the accumulator. However, this instruction does not initially obtain the number at location 31. Instead, the indexing of this instruction causes it to obtain the number at location $31_8 + 10_8$ which is the number at 41, the last number in the group of 10. The 1 *BPX* (01) 3 instruction tests the sign of index register 1. If the sign is positive (a 0 bit), the program branches to location 3 and the contents of the index register are reduced by 1. Index register 1 now contains 7_8 ; therefore, the second execution of the 1 *ADD* 31_8 instruction will add the number at location 40 ($31_8 + 7_8$) to the contents of the accumulator. The next execution of the 1 *BPX* (01) 3 instruction reduces the contents of index register 1 to 6_8 and branches back to location 3. The third execution of the 1 *ADD* 31 instruction obtains the number in location 37 ($31_8 + 6_8$).

On the next to the last pass through the iterative loop made up of the instructions at locations 3 and 4, index register 1 contains 1. The 1 *ADD* 31 instruction adds the number at location 32 ($31_8 + 1$) to the contents of the accumulator. The 1 *BPX* (01) 3 instruction senses the sign of index register 1; finding it positive, the instruction causes the program to branch again to location 3. Before the branch is completed, the contents of index register 1 are again reduced by 1. This operation causes the contents of index register 1 to change from 1 to negative zero (the addition of a number to its comple-

TABLE 1–56. INDEXED ITERATIVE PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	1 <i>XIN</i>	0.00010_8	Sets index register 1 to decimal 8.
2	<i>CAD</i>	30	Places first number in accumulator.
3	1 <i>ADD</i>	31	Adds number at location 31 + contents of index register 1 to accumulator.
4	1 <i>BPX</i> (01)	3	Branches back to location 3 if index register 1 is positive and subtracts index interval from contents of index register 1.
5	<i>FST</i>	150	Stores result.
6	<i>HLT</i>	—	End of program.
30 through 41	n_1 n_{10}	m_1 m_{10}	Initial data storage.
150			
			Result storage.

ment yielding negative zero as its result). When the branch is executed, the 1 *ADD* 31 obtains the number at location 31; the addition of negative zero to the address leaves the address unchanged. Thus, this execution of the *ADD* instruction obtains the last number to be added, completing the addition; the result remains in the accumulator. The 1 *BPX* (01) 3 instruction detects the negative sign of index register 1 and therefore does not execute a branch of program control. The program proceeds to the *FST* instruction to place the result in storage and then halts.

Some justification is necessary for the initial loading of index register 1 with the number decimal 8. In the addition of 10 numbers, 9 additions must be performed; the first operation of placing a number in the accumulator is not itself an addition operation. The last execution of the iterative loop in the program of table 1-56 is performed while the index register is negative; i.e., the loop controlled by the index register is repeated one time more than the number set into the index register. Therefore, in order to repeat the loop 9 times, the index register is set to decimal 8 or 10.

The indexed iterative program offers several advantages over both the straight-line program and the interactive program without indexing. The program with indexing is much simpler to set up than is the nonindexed iterative program, although both programs offer advantages over the straight-line program if more than 10 numbers are to be added. The indexed iterative program is also shorter (in terms of memory space) and faster (in terms of time of execution) than either the straight-line program or the nonindexed iterative program. Finally, the indexed iterative program may be reused on another group of numbers without any special precautions, whereas the nonindexed iterative program must have the address half of the *ADD* instruction changed back to its original value before the program can be reused in processing a new set of numbers put into the original data storage locations. Indexing does not change the instruction words stored in core memory, whereas the nonindexed iterative program does alter them.

5.3.4 Applications of Indexing

5.3.4.1 Indexed Number-Sorting Program

An indexed number-sorting program can be written to check through a block of numbers to find the largest number in that block. In addition, a program of this type can check for the existence of at least one other number as large as the largest number found. (See fig. 1-68.) The indexed number-sorting program can be compared with the straight-line number-sorting program given in figure 1-67. Table 1-57 shows initial data storage for the program.

The first block within the indexed number-sorting program sets index register 1 to a value two less than the amount of numbers to be sorted. This value is one less than the number of iterative comparisons to be performed, as required by the operation of an index register in controlling the number of repetitions. The value is two less than the amount of numbers to be sorted, since the number of comparisons to be performed is one less than the total of numbers. The second instruction within the first program block places the first number in the left accumulator. (For convenience, only the numbers in the left half-words are considered. All right half-words contain zero.)

The next block within the program compares a second number (N') to the number in the left accumulator (N). The left accumulator will be positive if $N > N'$; it will be negative if $N' \geq N$. When $N > N'$, the program continues to the instruction at location 5, which restores N from the left B register to the left accumulator. When $N' \geq N$, the program branches to the instruction at location 12, which tests for $N = N'$. If they are not equal, $N' > N$; therefore, N' is placed in the left accumulator by the instruction at location 13. This instruction, 1 *CAD* 101, obtains the same number from core memory as does the 1 *DIM* 101 instruction, since the index register contents have not yet been changed; i.e., the program is still executing the same pass through the iterative routine. When $N = N'$, the program branches to the instruction sequence starting with location 15. The first instruction

TABLE 1-57. DATA STORAGE FOR INDEXED NUMBER-SORTING PROGRAM

LOCATION	CONTENTS	COMMENTS
21	<i>CAD</i> x	x is address of number equal to largest number found if CONDITION LIGHT 1 is on at end of program.
40	0	Storage for largest number found.
100 through $n+99$	N_1 0 N_n 0	Initial data.

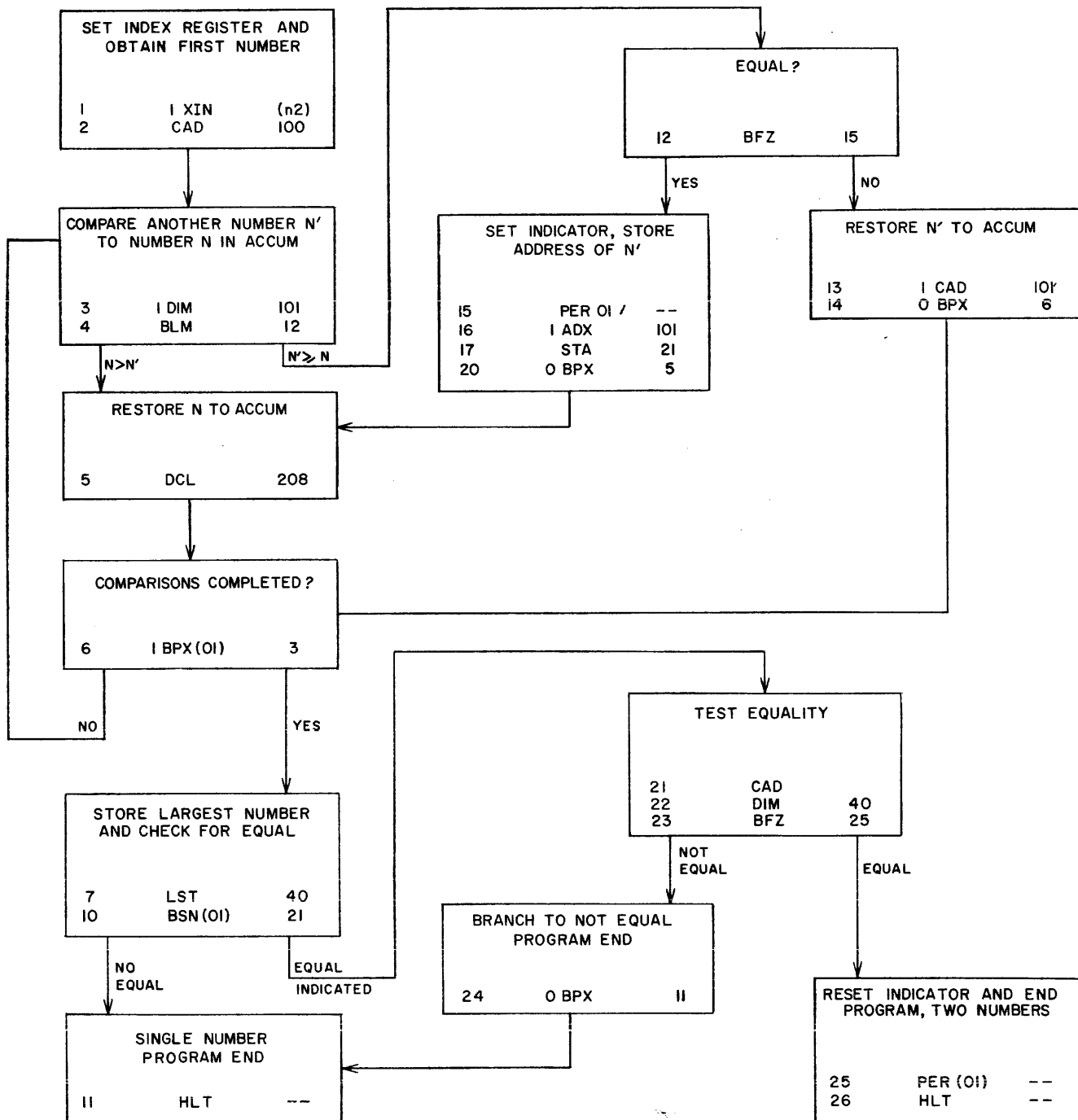


Figure 1-68. Indexed Number-Sorting Program., Flow Chart Form

in this sequence, *PER (01)*₈, turns on CONDITION LIGHT 1 to indicate that a number has been found equal to the largest number discovered so far in the program. The next instruction, *1 ADX 101*, places the address of *N'* in the right A register; i.e., the instruction places in the right A register the sum of the address half of the instruction, 101, and the contents of index register 1; this sum designates the address from which *N'* was taken. The instruction at location 17 stores this address as the

right half-word in location 21. (Location 21 contains an instruction which will be used in a terminal routine to test the equality of the largest number found and the last equal number found.) The program then branches to restore *N* to the left accumulator. Thus, when *N=N'*, *N* is retained for the next comparison while the address of *N'* is stored.

The first pass through the iterative portion of the program is performed in one of the three ways just de-

scribed, depending upon whether $N > N'$, $N = N'$, or $N < N'$. If either of the first two paths is followed, the number restored to the left accumulator is N ; if the third path is followed, N' is placed in the accumulator in preparation for the next comparison. In any case, the number found to be larger is left in the left accumulator for the next comparison.

The block of the program containing the check for completion of comparisons is simply a 1 *BPX* (01)₈ 3 instruction. If the comparisons are not completed, the program branches back to location 3 to begin another pass through the iterative routine. A new number is obtained to compare with the largest number found in preceding passes, since the index register is reduced by 1 by the 1 *BPX* (01)₈ 3 instruction.

After the last number comparison is completed, the program stores the largest number in location 40. The program then checks for the indication that a number was found which was equal to the largest number found by the program at some particular point in its execution. This check is performed by the *Sense* (CONDITION LIGHT 1) instruction, *BSN* (01)₈; this instruction causes a branch of program control if CONDITION LIGHT 1 is on and turns off the CONDITION LIGHT. If the CONDITION LIGHT is off when this instruction is executed, the program proceeds to the *HLT* instruction at location 11. Thus, if no number equal to the largest number is found, the Central Computer System stops with the largest number stored in location 40, CONDITION LIGHT 1 off, and the program counter containing the address 12.

If CONDITION LIGHT 1 is on when the *BSN* (01)₈ 21 instruction is executed, the CONDITION LIGHT is turned off and the program branches to the sequence starting at location 21. This terminal routine tests the equality of the number whose address is stored as the right half-word in location 21 and the largest number stored in location 40. This test is necessary, since the number which was equaled by the number whose address was stored may have been rejected by the program if a larger number were found in a later comparison. The program changes the contents of the address half of location 21 each time a number is found which is equal to the number retained in the left accumulator register; this address is not removed if another number is found which is larger than the number previously re-

tained in the left accumulator. Thus, the number whose address is stored in location 21 after all comparisons have been completed is the last number equal to the largest number found up to that point in the program.

The terminal routine starting at location 21 compares the magnitude of the largest number (in location 40) with the magnitude of the number whose address has been stored in location 21. If the two numbers are not equal, the program branches to the program end that indicates no equal number. If the two numbers are equal, CONDITION LIGHT 1 (turned off by the *Sense* instruction in location 10) is turned on again and the program halts the Central Computer System. Thus, if at least one number equal to the largest number is found, the Central Computer System stops with the largest number stored in location 40, address of the equal number in location 21, CONDITION LIGHT 1 on, and the program counter containing the address 27. Therefore, there are two indications of the result of the program; either the state of the CONDITION LIGHT or the contents of the program counter can be examined to discover whether an equal number was found.

Interpretation of a program result by examining the contents of the program counter after the Central Computer System halts is particularly important in maintenance programs. A program halt caused by detection of an error is identified by determining where in the program the halt occurred. The use of the two program terminations in the indexed number-sorting program is not critically important since CONDITION LIGHT 1 also indicates the result of the program.

5.3.4.2 Indexed Function Evaluation Program

The function evaluation program given in table 1-53 and described in 5.2.2.2 can be modified by the addition of indexing facilities to evaluate the function $y = ax^2 + bx + c$ for several discrete values of x . In addition, by using a second index register, the program can evaluate the function for each value of x with a series of different values for a , b , and c . This indexed function evaluation program must use one index register to control the iterative routine which determines the value of y for each value of x with a single set of values for a , b , and c . The second index register controls iterations of the indexed iterative routine using different values of a , b , and c . Table 1-58 presents this indexed function evaluation program.

TABLE 1-58. INDEXED FUNCTION EVALUATION PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	2 <i>XIN</i>	3(K-1)	Sets index register 2 to a value 3 (K-1), where K is the number of sets of a, b, and c.
2	1 <i>XIN</i>	(1)	Sets index register 1 to a value L, which is the number of values of x to be used in evaluating y for each set of a, b, and c.
3	1 <i>BPX</i> (01)	4	Reduces contents of index register 1 by 1.
4	2 <i>CAD</i>	200	Places value of a selected by indexing in left accumulator.
5	1 <i>MUL</i>	100	Multiplies a by x.
6	<i>DSR</i>	<i>n</i>	Scales product down by 2^{-n} to match scale of b.
7	<i>SLR</i>	0	Rounds off product to 15 significant bits.
10	2 <i>ADD</i>	201	Adds value of b selected by indexing to scaled product.
11	1 <i>MUL</i>	100	Multiplies sum by x.
12	<i>SLR</i>	<i>m</i>	Scales product up by 2^m to match scale of c and rounds off to 15 significant bits.
13	2 <i>ADD</i>	202	Adds value of c selected by indexing to second scaled product.
14	1 <i>LST</i>	$377+L(K-1)$	Stores y.
15	1 <i>BPX</i> (01)	4	Has y been evaluated for all values of x with one set of values for a, b, and c? If so, continue with next step; if not, evaluate y for next value of x.
16	<i>CAD</i>	14	Places first storage address of last storage block in right accumulator.
17	<i>SUB</i>	2	Subtracts number of items in block of storage from first storage address of last block.
20	<i>RST</i>	14	Places new starting storage address in instruction word in location 14.
21	2 <i>BPX</i> (03)	2	Has y been evaluated for all sets of a, b, and c? If so, continue to next program; if not, repeat iterative evaluations with next set of a, b, and c.
22			Next program.
100	x_1	—	Storage for values of x.
through			
$77+L$	x_L	—	
200	a_1	—	
201	b_1	—	Storage for sets of values of a, b, and c.
202	c_1	—	
through			
$177+K$	a_K	—	
$200+K$	b_K	—	Result storage locations.
$201+K$	c_K	—	
400		—	
through			
$377+KL$		—	

This indexed function evaluation program actually contains two iterative routines, one within the other. The routine contained in locations 4 through 15 is an iterative version of the program given in table 1-53; it is indexed and controlled by index register 1 to evaluate y for all values of x , using one set of values for the constants a , b , and c . This routine is contained within an indexed iterative routine which repeats this routine once for each set of values of a , b , and c , and changes the storage location for each iteration of the included iterative routine. The program thus develops a set of values for y , one for each value of x , using one set of values for a , b , and c ; then develops another set of values for y , one for each value of x , using another set of values for the three constants; and so on.

The first step in the indexed function evaluation program sets index register 2 to control the larger iterative routine. The value to which this register is set is three times a number which is one less than the number of sets of values of a , b , and c to be used in the program. The multiple of three is required since the values of a , b , and c are stored in successive locations in core memory with successive values of a (for example) spaced three registers apart in one table of values. In effect, the values of a , b , and c are stored in three tables interleaved by three to comprise one table. If three different tables, one for each constant, were used in this program, the value set into index register 2 would not be three times the number of sets less one.

The second instruction in the program sets index register 1 to the number of values of x to be used with each set of constants in evaluating y . This value is reduced by the third instruction, 1 *BPX* (01)4, to the proper value for controlling the number of iterations of the included evaluation routine. This third instruction could be eliminated by changing the value of the right half-word in the 1 *XIN* instruction; however, this value is used in modifying the storage address later in the program.

Each evaluation is performed by the routine contained in locations 4 through 13. The instruction 2 *CAD* 200 is indexed by the contents of index register 2 to obtain the value of a stored in the highest memory location, $177 + K$. The next instruction multiplies this value of a by the value of x in location $77 + L$. The instruction in location 10 adds the value of b stored in memory location $200 + K$; although this instruction is indexed by the same amount as was the instruction in location 4, this instruction obtains a value of b , since its address half is one larger than that of the instruction in location 4. Similar reasoning applies to the instruction at location 13, which obtains the value of c in location $201 + K$.

The procedure used in storing the result of each iteration deserves some special comment. The entire indexed function evaluation program will produce K tables, each containing L values of y . It has been noted that the

program first produces the value of y for the last value of x with the last set of values of a , b , and c . This value of y can be stored in the L th position in the K th table by writing the *LST* instruction with the address half specifying the first address of the K th table and indexing the instruction by index register 1. Since index register 1 is reduced by 1 after each iteration of the included evaluation routine, each value of y obtained with the K th set of values for a , b , and c is stored in a memory location whose address is one lower in value than the address of the preceding storage location; i.e., the values of y are stored in an order which corresponds to the order of storage of values of x used in calculating them. When a new set of values for a , b , and c is selected by stepping index register 2, the starting storage address for the next table of values of y must be changed. If the $K-1$ table of values is to be stored (physically) before the K th table, the address half of the instruction in location 14 must be reduced by the value L when index register 2 is stepped to obtain a new set of constants. In this way, the order of storage of the K tables of values corresponds to the order of storage of the sets of constants.

The same order of storage could be attained by using a third index register to control the *LST* instruction. This third index register would be set initially to the $KL-1$ and reduced by one after the storage of each result. The *LST* instruction would then have the value 400 as its address half. The resulting program using three index registers would be shorter by two instructions than the program given in table 1-58.

5.3.5 Program Precautions in Indexing

5.3.5.1 Number of Cycles of Iterated Routine

In general, the number of repetitions of an indexed iterative routine which will be performed is one greater than the value contained in the controlling index register at the start of the first iteration. This is true if the *Branch and Index* instruction controlling the loop is placed at the end of the loop; the loop is repeated once after the index register is stepped negative. In this situation, the minimum number of repetitions of the loop is 1. Table 1-59 lists the number of repetitions of an iterative loop for different starting values in the controlling index register when the 1 *BPX* (s) instruction is placed at the end of the iterative loop and (s) equals 1.

If only one pass through the loop is required, the controlling index register must be loaded with negative zero (1.77777₈) rather than positive zero. If the index register is loaded with positive zero and (s) equals 1, the index register will be positive at the end of the first pass causing the program to branch and perform a second pass through the routine. At the end of the second pass, since the index register contains 1.77776₈, no branch is executed.

TABLE 1–59. CONTROL OF LOOP ITERATIONS

INITIAL VALUE IN INDEX REGISTER	NUMBER OF LOOP REPETITIONS
N	$N + 1$
2	3
1	2
+0	2
–0	1

If a program must be written to allow for the possibility that no iterations of an indexed loop are needed on some runs through a program, the requirement can be met by placing the *BPX* instruction controlling the iteration of the loop apart from the loop and using an unconditional branch at the end of the loop to the controlling instruction. Table 1–60 gives an example of this type of program.

With an iterative program of the form given in table 1–60, the number of passes through the routine is equal to the value n set into the controlling index register. If n equals negative zero, no passes are made through the iterative loop. Again, positive zero is a special case; if n equals positive zero, the program will make one pass through the iterative loop.

5.3.5.2 Use of Zero Address with Indexable Instructions

When an indexed instruction is to obtain the word in memory location 0.00000_8 on the last pass through an iterative routine, it would appear that the instruction should be written with an address half which is 0.00000_8 .

No difficulty is encountered with this value of address half until the last iteration of the loop. At that point, the index register used in controlling the loop contains negative zero (1.77777_8). When negative zero is added to positive zero, the result is negative zero. Thus, the modified address half of the indexed instruction will refer not to the first address in core memory but rather to the clock register.

The indexing error resulting from use of an indexed instruction with a zero address half can be avoided by writing the zero address half as 1.00000_8 . When the last iteration of the program loop is performed, the indexed address half of the instruction is still 1.00000_8 since the addition of negative zero to a number (other than positive zero) leaves the number unchanged. Therefore, the instruction will obtain the word in the first core memory location rather than the contents of the clock register.

When IO class instructions are to be indexed and a zero address half is desired as part of the instruction, similar precautions are required. The *SDR* instruction must have a zero address half with the value 1.00000_8 ; otherwise, when indexed by negative zero, the drum control register will be loaded with 0.03777_8 rather than the desired 0.00000_8 . Similarly, a zero address half for the *LDC* instruction must be of the form 1.00000_8 or, after indexing by negative zero, the instruction will load the IO address counter with 0.600000_8 , selecting the clock register as the source or destination of an IO transfer. In the latter case if reading the words read from an IO unit will be lost; if writing, all positive zeros will be supplied to the IO unit.

When an *RDS* or *WRT* instruction is to be indexed, it must be remembered that the number of words to be transferred in the IO operation initiated by one of these instructions is specified by the address half of the instruction.

TABLE 1–60. ZERO REPETITION INDEXED ROUTINE

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
1	1 <i>XIN</i>	n	Sets index register 1 to value n .
2	1 <i>BPX</i> (01)	4	Controls number of repetitions of loop starting at location 4.
3	0 <i>BPX</i>	y	Branches to next program.
4 through 4+k			Iterative loop.
5+k	0 <i>BPX</i>	2	Unconditionally branches to controlling instruction at location 2.

tion after indexing interpreted as a 16-bit binary integer; i.e., the sign bit of the address half is interpreted as the most significant bit rather than as a sign bit. Therefore, if the address half of one of these instructions is negative zero (1.77777_8) after indexing, the transfer of $1.77,777_8$ words is called for. Further, an *RDS 0* or *WRT 0* instruction (where the address half is positive zero) may be illegal for the IO unit involved. (Refer to 5.2.4.)

5.4 PROGRAM PREPARATION

5.4.1 Program Organization

5.4.1.1 General

A program may be considered simply as a series of instructions which accomplishes a given function. However, when a long or complicated program is to be handled, it is often desirable to consider this program as a series of blocks of instructions wherein each block performs a single function subordinate but necessary to the accomplishment of the function of the entire program. Each block of instructions within the larger program is called a subprogram, routine, or subroutine of the larger program. These terms are variously used, since what is called a program in one application may be only a subroutine of a still larger program.

In order to plan a program efficiently, it is often necessary to prepare a flow chart of the program being prepared. The flow chart should show the logical blocks and their functions as well as the branches of program control, whether unconditional or conditional, upon the results of preceding operations. With this flow chart, it is possible to recognize the existence of equivalent routines within a program and to thus eliminate duplication within the final program.

The degree of detail shown within a flow chart can vary from the extreme of one block for each instruction to the other extreme wherein a block represents what will eventually be a subroutine long enough to be called a program in its own right. While the degree of detail can be determined anywhere between these two extremes, no simplification results if a block is presented for each instruction. On the other hand, a flow chart showing the entire program as a single box is simplified to the point of uselessness.

If figure 1-68 were redrawn and the instructions omitted, the figure would represent a reasonable flow chart for use in writing the indexed number-sorting program. (The presentation of the written program in flow chart form makes it easier to follow than presentation in tabular form.) However, if the entire indexed number-sorting program were to be used as a subroutine within a larger program, the flow chart for that larger program might show the number-sorting routine as a single box.

5.4.1.2 Master Programs

A large program may be made up of several smaller programs, each performing a portion of the overall function of the large program. In some cases, the large program does not properly contain these smaller programs; instead, the large program selects each smaller program in the proper order to perform necessary data processing without performing any data processing itself. A program of this type, which controls other programs without doing any processing itself, is called either a master program, an executive routine, or a sequence selection program. In effect, a master program can do no processing except through the use of other programs which cannot perform the entire processing task without having the master program to control the sequence in which they are performed.

The smaller processing programs may themselves contain blocks of instructions which can be called programs or routines. For example, if a given program requires the performance of a particular operation, a routine may be written to perform that operation and placed in core memory with the processing program. The processing program may then be written to refer to the routine whenever the operation performed by that routine is required by the processing program.

5.4.1.3 Subroutines

A subroutine is a group of instructions which perform a distinct function and it may be written in one of two ways; the subroutine may be open or closed.

An open subroutine is a sequence of instructions performing a particular function but with no special provision for incorporation into a longer program. Any of the programs discussed up to this point can be considered open subroutines since they end either with an *HLT* instruction or at the point where the next program or subroutine may begin. Open subroutines are useful in assembling a long program; if subroutines are already written for each function to be performed in a program, it is more efficient to assemble these routines as required into a complete program rather than to write new routines to accomplish the same functions. In addition, a program assembled from previously written subroutines is more reliable. A written subroutine has usually been tested before being placed in a program library and can therefore be depended upon to perform its assigned function. One caution must be observed when using library routines in a program: the restrictions on data magnitude and location established by the subroutine must be satisfied by the program containing the subroutine. In many cases, two or more subroutines exist which perform very similar functions. They may differ in their time of execution or storage space requirement, in the precision with which they produce results, or in the number of parameters which can be varied. The choice of a particular

subroutine for use in a specific program can be made only after consideration of all these factors in terms of the overall program requirement.

Some difficulties are encountered when a program must use a particular subroutine several times but cannot employ indexing; i.e., when the subroutine is not used iteratively. If an open subroutine is to be used in this case, it is necessary to copy the open subroutine into the program at each point of use in the larger program. As in the case of straight-line programming, needless repetition of instruction sequences should be avoided.

The repeated copying of an open subroutine into a program at each point of use can be eliminated by making the subroutine a closed one. The subroutine is placed in memory apart from the program and provided with a return provision while a leave provision is inserted in the main program at each point of use of the subroutine. (See fig. 1–69.) The leave provision is simply a 0 BPX S instruction which causes an unconditional branch to the address of the first register containing the closed subroutine. The return provision within the closed routine consists of the instructions at addresses S and T. The instruction at location S stores, as the address half of the instruction at location T, the address of the instruction in

the main program from which the branch was performed and to which program control should return after completion of the subroutine. The instruction at location T performs the branch of program control to the main program. The 0 BPX S instruction which calls the closed subroutine into use can be considered as a pseudo-instruction code for the operation performed by the closed subroutine. For example, if the closed subroutine starting at location S extracts the square root of the number it finds in the accumulators and leaves the square root in the accumulators, the 0 BPX S instruction is a pseudo-instruction code for an extract-square-root operation. In effect, the 0 BPX S instruction is a new instruction which can be used just as are the other 48 instructions. Since any open subroutine can be converted into a closed subroutine by the addition of the return provision, the use of closed subroutines can greatly simplify the programming of a complex operation.

The only limitations on the use of subroutines are that the initial requirements of the subroutines must be satisfied before the subroutine is called into use and that the locations in core memory needed for storage of the subroutine must not be used for any other purpose while the subroutine is to be used or stored for later use. A section of core memory is usually reserved for storage of subroutines required by the program being executed.

5.4.1.4 Compiler Programs

When a number of standard subroutines exist and can be made accessible to the Central Computer System, programs can be written using pseudo-instruction codes calling for particular operations. A program of pseudo-instruction codes can be presented to the Central Computer System for preparation as a conventional program of instructions by a special program known as a compiler. The compiler recognizes the pseudo-instruction codes, selects the appropriate subroutines from storage, and prepares a program which accomplishes the desired final result. By using a compiler, the programmer is relieved of the job of mechanically assembling standard routines in punched card form into program decks or of rewriting these routines with addresses appropriate for the overall program. In effect, a compiler routine enables the Central Computer System to respond to more complex instructions than the 48 basic instructions built into it.

5.4.1.5 Utility Programs

Utility programs are essentially large closed subroutines which are handled as programs in their own right. The primary distinction between a utility program and a closed subroutine, other than size, is that a utility program performs some function for several other programs, whereas a closed subroutine performs a particular function for only one program. Closed subroutines are generally loaded into core memory with the program that

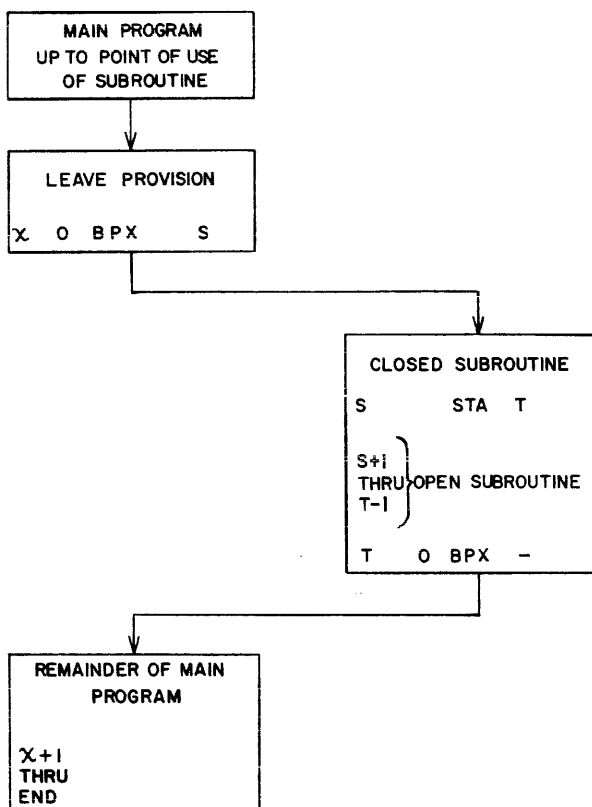


Figure 1–69. Use of Closed Subroutine
in Larger Program

will use them. Utility programs are usually loaded into core memory and left there while different programs are loaded and executed. Thus, the utility programs will usually occupy a particular block of registers in core memory throughout the execution of an entire series of programs which make use of the utility programs.

An example of a utility program is one which translates information from Hollerith code stored in a card image to binary form. A program of this type might be of use to more than one program and might therefore be placed in core memory as a utility program. Another utility program might be one which took binary information and converted it into Hollerith code in card image form for delivery to the line printer. A utility program of this type would be useful in the first running of any program to debug that program. The printout utility program might be called for at several points in the program under test to provide a printout of the contents of all computing registers for use in analysis of the program under test.

5.4.1.6 Parameters

If a subroutine or utility program is to be of wide possible use, it must be written with provisions for changing the parameters within the subroutine or utility program. Parameters may be those numerical constants used in processing other data or the locations of data to be processed. For example, a routine which can translate a block of binary information into octonary card image form and have this information printed out on the line printer may have as parameters the number of words in the block to be translated and the address of the first word in the block. If these parameters can be changed, the storage locations used for the subroutine can be changed, if necessary, to facilitate storage of the main program. Similarly, a main program can be written with unassigned values for parameters which may change from day to day or from problem to problem; the values for these parameters can then be set in manually before the program is to be run. Thus, any program, subroutine, or utility program can be made more versatile by writing it with parameters which may be reset for each run of the program.

Several means exist for setting the values of parameters in a program or routine. If the parameters can be set for the routine and left with these values, a parameter card (a punched card containing identification of the parameter and the value it is to have) can be loaded along with the cards containing the routine. A routine whose parameters are set in this way is said to have preset parameters, since they are preset when the routine is loaded. In some cases, program parameters may be set manually into test memory with the program written to refer to test memory when a parameter is called for. Finally, a subroutine can be written whose parameters are

set by the using program before it calls the subroutine into use. In effect, the using program would set the parameters in the subroutine by storing the proper values in appropriate locations in the subroutine and then calling the subroutine into use. Precautions are required within the using program to insure its setting the proper parameters into the subroutine before each use of the subroutine.

5.4.2 Program Coding

5.4.2.1 Absolute Programming

The programs and routines given as examples up to this point have been presented in a variety of forms. Instruction locations and address halves appear as octonary numbers or as symbolic addresses whose actual values depend upon other addresses which are program parameters. No difficulty is encountered in following these programs, since they are rather short. Similarly, no difficulty is encountered in writing a short program using absolute addresses; i.e., locations and address halves, perhaps abbreviated in octonary notation, describing actual memory addresses or values of numerical constants. In a short program, it is relatively simple to keep track of these addresses used in the program. Further, the insertion of an extra instruction which requires changing the storage locations of succeeding instructions and the address halves of branch class instructions is not difficult.

In the writing of a long program, manipulation of absolute addresses becomes difficult. Further, if an extra instruction must be inserted early in the program, the job of changing storage locations and instruction address halves becomes almost impossible. To avoid the tedious work required in absolute programming, a procedure known as symbolic programming can be used.

5.4.2.2 Symbolic Programming

When a long program is to be prepared, it is first divided into blocks of operations with each block of convenient size for manipulation. As each block is written, it can be tested, revised, and reviewed before its incorporation into the final program. Since the instructions within each block must not be assigned absolute addresses until after the blocks have been assembled into a complete program, an arbitrary or symbolic address can be used to facilitate both the writing of each block and the assembly of the entire program.

A symbolic address is written with up to six alphanumeric symbols; the first three symbols are usually alphabetical and the last three are numerical. A symbolic address notation thus would look like this: AB.C12.3. The first two letters generally identify the overall program which will contain the block being written. The third letter identifies the block itself, thus allowing the division of a program into 26 blocks. If more blocks are

necessary, the overall program may be assigned two or more combinations of letters for the program identification position. The first two digits allow the identification of up to 100 addresses within each program block, or 2,600 addresses per program. The last digit is generally used only for insertion of instructions after the block is initially written.

It is important to realize that a symbolic address has no inherent numerical significance. Although not strictly necessary, consecutive instructions are usually assigned consecutive symbolic locations to simplify sorting of the instruction steps into their proper order. Symbolic addresses need be assigned only to the location of the first instruction in a sequence and to those instructions whose locations are referred to in the address halves of other instructions. Further, not all addresses within the program need be indicated in symbolic form. If an actual address is known for a particular operand or operation, it can be written into the program and is usually indicated by enclosing the actual address in parentheses. Similarly, those instructions whose address halves should contain numerical constants (such as shift class instructions) can be written with the absolute value in octonary form and be indicated as absolute by their inclusion in parentheses. Table 1–61 shows the nonindexed number-sorting program given in figure 1–67 as it might be written in symbolic form.

The notations used in table 1–61 do not include any 6-digit symbolic addresses simply because no inserted instructions were necessary and because not all instructions are assigned specific symbolic addresses. If all of the instructions had been assigned symbolic locations, any added instructions would be assigned symbolic locations whose first five symbols match the symbols assigned to the instruction, after which the added instructions would be inserted, with the added instructions numbered sequentially in the sixth symbol position.

5.4.2.3 Assembly Program

Once a program is written in symbolic form, the task of translating it into absolute form can be performed by the Central Computer System under control of a utility program called an assembly program. The assembly program accepts a symbolic program presented on instruction cards. (Refer to 4.2.2.) An absolute location is assigned to each instruction with the locations assigned in the same order in which the instructions of the program are presented. The assembly program then makes another pass through the program being assembled to complete the assignment of absolute equivalents for all symbolic addresses and to translate all information in the program itself into binary form. (Since symbolic addresses are not translated, they can be written decimally without introducing any difficulties. The assembly program simply provides a one-to-one correspondence of

symbolic addresses to absolute addresses.) The assembly program will also punch a deck of binary cards from which the assembled program may be inserted and executed, together with a printout of the program, listing all comments, symbolic locations, constants, assigned storage locations and their contents in octonary form, and the initial and final drum storage locations, if any. The binary deck is prepared as described in 4.2.3.

In order to submit a program in symbolic form for assembly, not only must the instructions be prepared on punched cards but certain control information must also be prepared on punched cards.

5.4.2.4 Assembly Program Card Punching

Each instruction within a symbolically coded program is prepared on a separate instruction card. (See fig. 1–70.) The location field should be symbolically coded or may be blank if no symbolic tag is necessary for the location of that instruction. In the instruction field,

TABLE 1–61. NUMBER-SORTING PROGRAM,
SYMBOLIC FORM

LOCATION	INSTRUCTION	
	OPERATION	ADDRESS
AB A00	<i>CAD</i>	AB A30
	<i>DIM</i>	31
	<i>BLM</i>	AB A10
	<i>DCL</i>	(00020)
AB A01	<i>DIM</i>	AB A32
	<i>BLM</i>	AB A11
	<i>DCL</i>	(00020)
AB A02	<i>DIM</i>	AB A33
	<i>BLM</i>	AB A12
	<i>DCL</i>	(00020)
AB A03	<i>LST</i>	AB A40
	<i>HLT</i>	
AB A10	<i>CAD</i>	AB A31
	<i>BPX</i>	AB A01
AB A11	<i>CAD</i>	AB A32
	<i>BPX</i>	AB A02
AB A12	<i>CAD</i>	AB A33
	<i>BPX</i>	AB A03
AB A30	N ₁	
AB A31	N ₂	
AB A32	N ₃	
AB A33	N ₄	
AB A40	Result	

column 24 is reserved for the index indicator punched octonarily, columns 25 through 27 for the operation code punched alphabetically, and columns 28 and 29 for the index interval punched octonarily, if used. The address field may contain the address half of the instruction in symbolic or absolute form; an absolute address is identified by the lack of punching in column 30 and by the restriction to octonary punching in columns 31 through 35. If the address half is symbolic, columns 30 through 32 are alpha-numerically punched, and columns 33 through 35 are punched decimally. The comment field, columns 36 through 80, may be punched on an instruction card if a comment is necessary with the instruction.

A constant card is used to insert numerical constants in their address halves. Numerical constants may also be addresses, thus requiring the facility for inserting symbolic addresses on constant cards. The location field of a constant card need not contain an address if no symbolic tag has been used to refer to the location of the constant in the program. When both half-words inserted by the card are constants, they are punched in octonary form in the instruction and address fields, preceded by punches in the sign positions in columns 24 and 30; the control column, 17, is left blank. If the right half-word is symbolic, a 1 is punched in column 17. If the left half-word is symbolic, a 2 is punched in column 17. If both half-words are symbolic, a 3 is punched in column 17. For a constant card containing an instruction with a numerical constant as its address half, a 4 is punched in the control column. These control column punches direct the assembly program in its handling of the data on the constant card and allow the punching of both octonary and decimal information on a single card without confusing the assembly program.

Assignment cards perform the function of assigning absolute equivalents of symbolic addresses in symbolic programs. There are four types of assignment cards: location assignment cards, address assignment cards, drum location assignment cards, and temporary storage assignment cards. A location assignment card specifies the actual storage location for the word read from the next card read by the assembly program. Successive words read after this word are then assigned sequential storage locations; thus, the order in which instructions are submitted to the assembly program after the reading of a location assignment card determines their storage locations and eliminates the need for symbolic location codes for each word.

An address assignment card is used to supply an absolute address half for a word in the program being assembled whose symbolic location is given on the address assignment card. This type of assignment card is used to supply an address half for a word when the address half of that word cannot initially be assigned a sym-

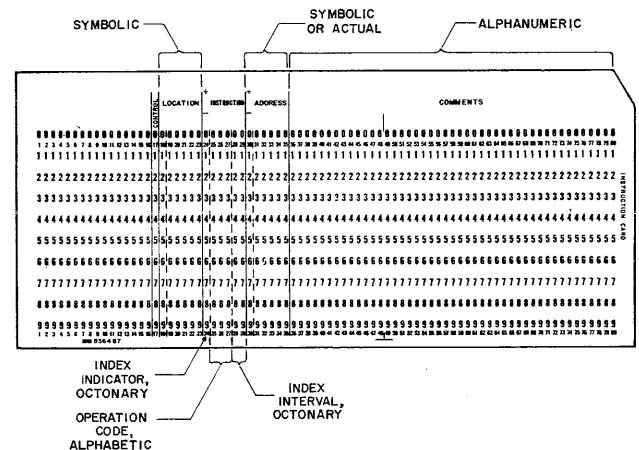


Figure 1-70. Instruction Card

bolic code; this condition may arise when instructions with symbolic addresses refer to data storage locations or to locations in sections of the program which are not being assembled at this time. Address assignment cards may be inserted at any point in a deck of program cards.

If a portion of a program is to be stored on a drum field after assembly, the starting drum storage address and drum interleave (if any) can be specified by a drum location assignment card. The drum location assignment card assigns a drum storage location to the word on the next card read by the assembly program. It should be noted that the drum storage location (if any) for the first word on each binary card is punched on that card to allow loading the appropriate portion of the assembled program into drum storage. The drum location assignment card indicates, to the assembly program, the starting drum storage location, allowing it to assign starting drum storage locations to each binary card containing the assembled program.

A temporary storage assignment card reserves blocks of actual addresses for storage purposes within the assembled program. The card itself contains two symbolic addresses, the first and last in the block of storage; it may reserve either core memory addresses or drum storage addresses. Use of a temporary storage address assignment card eliminates the need for an address assignment card for each register to be reserved for storage.

Control or assembly control cards direct the operation of the assembly program. A restore card, identified by a 5 punched in the control column, causes the assembly program to operate the line printer carriage feed, feeding the printout form up to the first line of the next page. By inserting restore cards at appropriate points in the symbolic program, the printout of the assembled program can be divided by routine or program block to

facilitate interpretation of the printed program. A drum termination card, identified by a 6 punched in the control column, stops the assignment of drum storage locations to words in the program begun by a drum location assignment card. A termination card, identified by a 9 punched in the control column, informs the assembly program that the entire symbolic deck has been read and is ready for processing. The termination card also contains the starting location of the program in its address field with the starting location given either in symbolic or absolute form. The assembly program uses this information to prepare a branch control card which is used in loading the assembled program from binary cards.

Comment cards, containing only information in the comment field, can be placed in a symbolic program deck

to insert long comments on the printed listing of the assembled program. In addition, an identification card may be inserted along with the symbolic program deck for use in the punch identification section of the assembly program. This type of card bears information in columns 33 through 48 which will be punched in the identification fields of all binary cards prepared by the assembly program. If the identification card has a 9 punched in column 49, the assembly program will convert the information in columns 33 through 40 into binary form and punch this binary information in the 12 row of each binary card prepared by the assembly program. The punching of all these special cards is summarized in table 1-62.

TABLE 1-62. ASSEMBLY DIRECTION CARD PUNCHING

CARD TYPE		FIELD OR COLUMN									
MAJOR VARIATIONS		CONTROL	LOCATION		INSTRUCTION				ADDRESS		
		17	18	19-20	21-23	24	25-27	28-29	30	31-32	33-35
Instruction	Absolute address	—	AN	AN	D	—O	A	—O	—	—O	—O
	Symbolic address	—	AN	AN	D	—O	A	—O	AN	AN	D
Constant	Both halves constant	—	AN	AN	D	S	O	O	S	O	O
	Right half symbolic	1	AN	AN	D	S	O	O	AN	AN	D
	Left half symbolic	2	AN	AN	D	AN	AN	D	S	O	O
	Both halves symbolic	3	AN	AN	D	AN	AN	D	AN	D	AN
	Operation code and constant	4	AN	AN	D	—O	A	—O	S	O	O
Assignment	Location	—	—	O	O	—	—	—	—	—	—
	Address	—	AN	AN	D	—	—	—	—	O	O
	Drum location	—	O	O	O	O	—	—	—	—	—
	Temporary storage	—	AN	AN	D	—	—	—	AN	AN	D
	Restore	5	—	—	—	—	—	—	—	—	—
	Drum termination	6	—	—	—	—	—	—	—	—	—
	Termination with symbolic address	9	—	—	—	—	—	—	AN	AN	D
	Termination with absolute address	9	—	—	—	—	—	—	—	—O	—O
Comment		Punched only in comment field columns 36-80									

Key: — no punching
O octonary
D decimal
A alphabetical
S octonary sign
AN alpha-numerical
—O empty or octonary

Digits indicate punch in corresponding row of card.

5.4.2.5 Printout of Assembled Program

An assembly program produces two forms of an assembled symbolic program submitted to it. The assembled program is prepared on binary cards for reinsertion and execution of the program. In addition, a printout of the assembled program is produced on the line printer. The printout of a portion of a program might resemble the example given in table 1-63.

This fragment of a printed out program can be compared with the symbolic program given in table 1-61, of which this is the printout. The left half of the printout is usually the absolute program, with one exception: the operation code for instructions is the alphanumeric representation. The absolute coding for the operation halves of the instructions is printed with the symbolic addresses. By transposing these two columns before printing, the absolute program can be followed more easily than if the instruction codes were also shown in octonary form. The positioning of information on the printout of an assembled program is determined by the wiring of the line printer control panel. (Refer to 4.5.2.1.)

5.4.3 Program Testing

5.4.3.1 General

When a program has been completed, it should theoretically be able to perform its intended function. However, the possibility of errors in the program exists until the program has been tested. In general, program testing (or debugging) involves running the program

through a properly operating computer and observing the results of the program at various points in its execution. If the program is used to calculate the outcome of a problem whose result is already known, the end result of the program execution can simply be compared with the known correct result. If the problem handled by the program is unsolved, other tests may be required. (Even after successful comparison with a known result, the program may still contain errors which must be located by other means.)

5.4.3.2 Program Monitoring

A program may be tested by monitoring its execution to insure that each sequence in its operation is correct. The program may be advanced a memory cycle or instruction cycle at a time, manually, while observing indicator lights showing the contents of all computing and control registers in the Central Computer System. This single-step procedure is quite time-consuming and is therefore impossible in a long program. An alternative to single-step operation is the temporary insertion of *HLT* instructions at critical points in the program. Again, analysis of the register indicators is necessary to decide whether or not the program has operated correctly up to that point. The disadvantage of this method is the lack of information about the contents of core memory. Selection of proper stopping points can minimize this disadvantage but the method is still quite time-consuming.

Most properly written programs contain checking provisions within themselves to determine the existence of incorrect results. In most cases, these checking provisions can be used to cause a program stop, at least during testing. When a program is stopped by the detection of an error, the error is located somewhere between the last error halt provision which did not stop the program and the error halt provision which did. That portion of the program can then be run on a step-by-step basis to discover the exact cause of the error. The physical facilities used in maintaining AN/FSQ-7 Combat Direction Central (discussed in Chapter 10) can also be used in debugging programs.

TABLE 1-63. PRINTOUT OF ASSEMBLED PROGRAM

INSTRUCTION			COMMENTS	INSTRUCTION		
LOCATION	OPERATION	ADDRESS		LOCATION	OPERATION	ADDRESS
00400	CAD	0.00430	OBTAIN FIRST NO.	AB A00	0.01000	AB A30
00401	DIM	0.00431	COMPARE SECOND NO.		0.01640	AB A31
00402	BLM	0.00415	TEST		0.05500	AB A10
00403	DCL	0.00020	RESTORE FIRST NO.		0.04600	00020
00404	DIM	0.00432	COMPARE THIRD NO.	AB A01	0.01640	AB A32

5.4.3.3 Test Programs

Various programmed facilities can be used in testing a new program. These include trace routines, post-mortem routines, and trap routines. A trace routine prints out the contents of selected registers after the completion of each step in the program under test. In effect, the trace routine is an executive program which alternately allows performance of one program step in the program being tested, then a printout routine, then another step of the program being tested, and so on.

A post-mortem routine, sometimes called a dump routine, causes the printout of all core memory registers or of selected core memory registers. The post-mortem routine thus provides a complete picture of the result of a program which cannot be obtained simply by examining the contents of the computing and control registers. Further, a post-mortem routine can be prepared to print out only those core memory registers which contain something other than zero.

A trap routine provides a printout of core memory contents for selected portions of a program being tested.

The portion of the program to be trapped is usually specified by control cards or by manual insertion.

5.4.3.4 Simulation Programs

It is possible to test programs written for one digital computer on another digital computer by using an executive routine which causes the digital computer being used to test the program to operate as the digital computer for which the program was written. The advantage of using a different computer for testing programs is dependent upon the state of the computer for which the program has been written. If construction of a new digital computer is nearing completion, it is desirable to have tested reliability programs available for use in testing the new computer. Since that computer may not be reliable immediately upon completion, the computer cannot be used to test programs which will later be used in testing the computer. In addition, simulation programs enable the use of a relatively inactive digital computer to test programs written for another computer whose operating time is taken up by other requirements. This is particularly advantageous in program testing, which is generally a very long process.

CHAPTER 4

DRUM SYSTEM

SECTION 1

SYSTEM DESCRIPTION

1.1 SYSTEM FUNCTION

AN/FSQ-7 Combat Direction Central stores information by means of the Central Computer System internal core memory, cards and tapes external to the Central Computer System, and a system of magnetic drums. This Chapter describes the functions and operations of the Drum System of the equipment as it applies to programming. Unless otherwise specified, reference to the direction central applies to both computers A and B of the duplexed central. (Refer to Chap. 1, 2, 3.)

The capacity of the Drum System enables it to fulfill storage requirements not satisfied by the Central Computer System internal core memory, while its speed enables it to fulfill requirements not satisfied by cards and tapes. The speed of a system is dependent upon its information access time; thus, the smaller the access time, the faster the system. The average access time of the Drum System is considerably greater than that of internal core memory (10.3 versus 3.0 microseconds, respectively); cards and tapes are much slower than either. However, successive transfers of words by the Drum System can occur at 10-microsecond intervals.

The Drum System performs two principal functions: as a time buffer, and as an auxiliary memory device. It acts as a time buffer between the Central Computer System and the Input, Output, and Display Systems. The Drum System also serves as an auxiliary memory device in that it stores information that must be reused from time to time. (See fig. 1-71.)

As a time buffer, the Drum System accepts information from the Input System at comparatively slow, intermittent rates. Periodically, at intervals prescribed by the Central Computer System program, this information is transferred to the Central Computer System at speeds up to the Drum System maximum speed (one word every 10.0 microseconds). Similarly, processed data for the Output and Display Systems is received by the Drum System from the Central Computer System at intervals prescribed by the Central Computer System program

and at speeds up to the Drum System maximum speed. The Drum System then transfers the processed information to the Output and Display Systems and to the Drum System of the other computer in AN/FSQ-7 Combat Direction Central. (See fig. 1-72.) The transfers from the Drum System to the Display and Output Systems are at speeds which are slower than Drum System maximum speed.

As an auxiliary memory device, the Drum System supplements the Central Computer System internal core memory. The storage capacity of the entire Drum System is about 19 times that of internal core memory (153,000 versus 8,192 words); the storage capacity of cards and tapes is larger than either.

About 64 percent of the Drum System capacity is reserved exclusively for the storage of information such as subprograms, mathematical values, and constants. This auxiliary memory portion of the Drum System alone, with a storage capacity of 98,304 words, has 12 times the capacity of the Central Computer System internal core memory. Additional storage space is obtained through the use of that section of the Drum Sys-

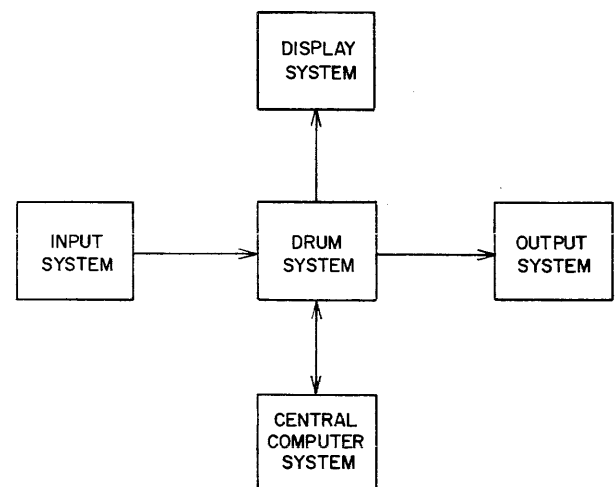


Figure 1-71. Relation of Drum System to Other Systems

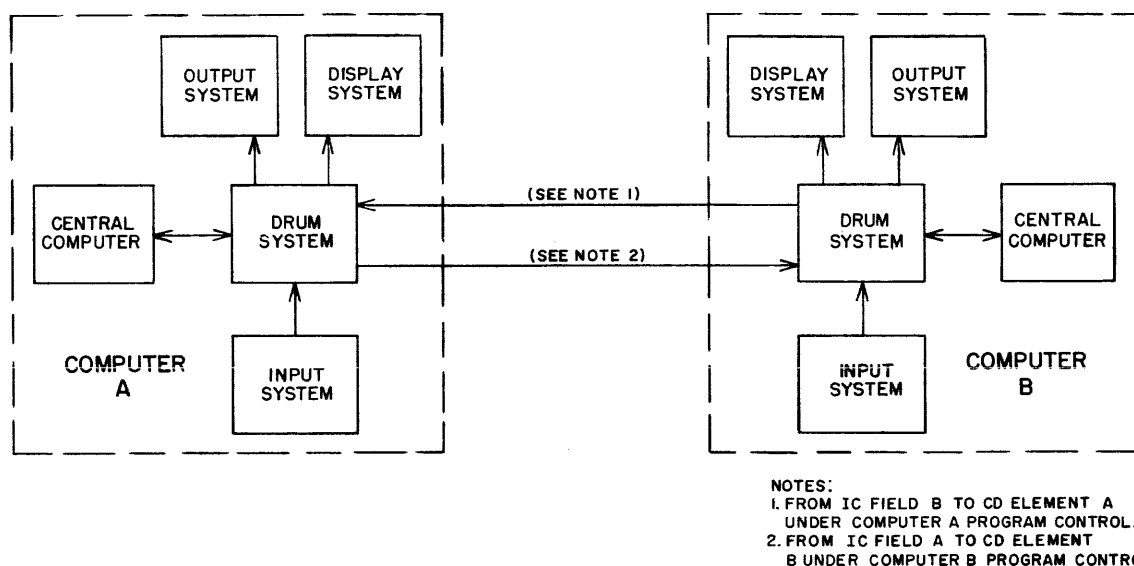


Figure 1-72. Intercommunication in AN/FSQ-7 Combat Direction Central

tem which is used as a time buffer between the Central Computer System and the Display System.

The transfer of information to and from the Drum System is logically divided into two categories: outside-to/from-drum (OD), and computer-to/from-drum (CD). (See fig. 1-73.) The OD and CD sides of the Drum System operate independently of each other.

1.2 SYSTEM DATA

1.2.1 Physical Description

There are 12 identical drums in each Drum System. A drum is a metal cylinder 12.6 inches in length with a diameter of 10.7 inches and a weight of 85 pounds.

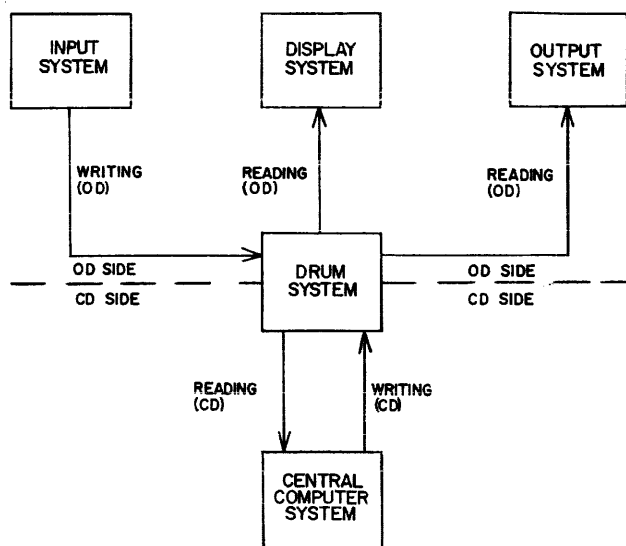


Figure 1-73. The OD and CD Sides of the Drum System

Each drum is individually mounted along its longitudinal axis on a fixed shaft, as shown in figure 1-74, and each is driven by a synchronous motor through a toothed belt at a speed of 2,914 revolutions per minute. The cylindrical surface of each drum is plated with a 0.0005-inch layer of a magnetic nickel-cobalt alloy.

As a drum rotates, fixed magnetic heads transfer information to and from its surface by recording (writing) binary information in the form of small electromagnetic flux patterns and later detecting (reading) these patterns. The fixed magnetic heads are mounted with a small air gap of critical tolerance between them and the drum surface. Since reading on one side of a drum (OD or CD) and writing on the other side often occur concurrently, two sets of heads are required, except in the case of the auxiliary memory drums where reading and writing occur only on the CD side.

Magnetic heads for reading or writing are held rigidly in place by bars which are parallel to the longitudinal axis of the drums. (See fig. 1-75.) Six pairs of these bars are arranged axially around each drum (each bar can support up to 40 magnetic heads). One bar of each pair supports the heads for OD operations, and the other bar supports the heads for CD operations.

1.2.2 Logical Description

Each drum in the Drum System is logically divided into sections, called fields. There are 75 fields in the Drum System. Eleven of the 12 drums contain six fields each; the 12th (the radar data drum) contains nine. Each of the six pairs of longitudinal bars arranged axially about a drum is associated with a field of the

drum. One bar of each pair supports the heads which perform OD operations. A unique arrangement is made to accommodate the nine fields of the radar data drum.

A drum field stores a particular type of information and is named to correspond with the type of information stored. Each drum derives its name from the names of the fields it contains; e.g., both the LOG and MIXD drums derive their names from the first letter of the abbreviated names of the fields each contains. Figure

1-76 is a pictorial representation of the logical divisions of each drum. In this figure and all subsequent figures, drum fields are shown as subdivisions of a drum. This convenient logical convention is used for illustrative purposes only and is not to be accepted as a true physical relationship.

1.2.3 Logical Definitions

The smallest unit of intelligence that can be written on or read from a drum is called a bit. (See fig. 1-76.) If

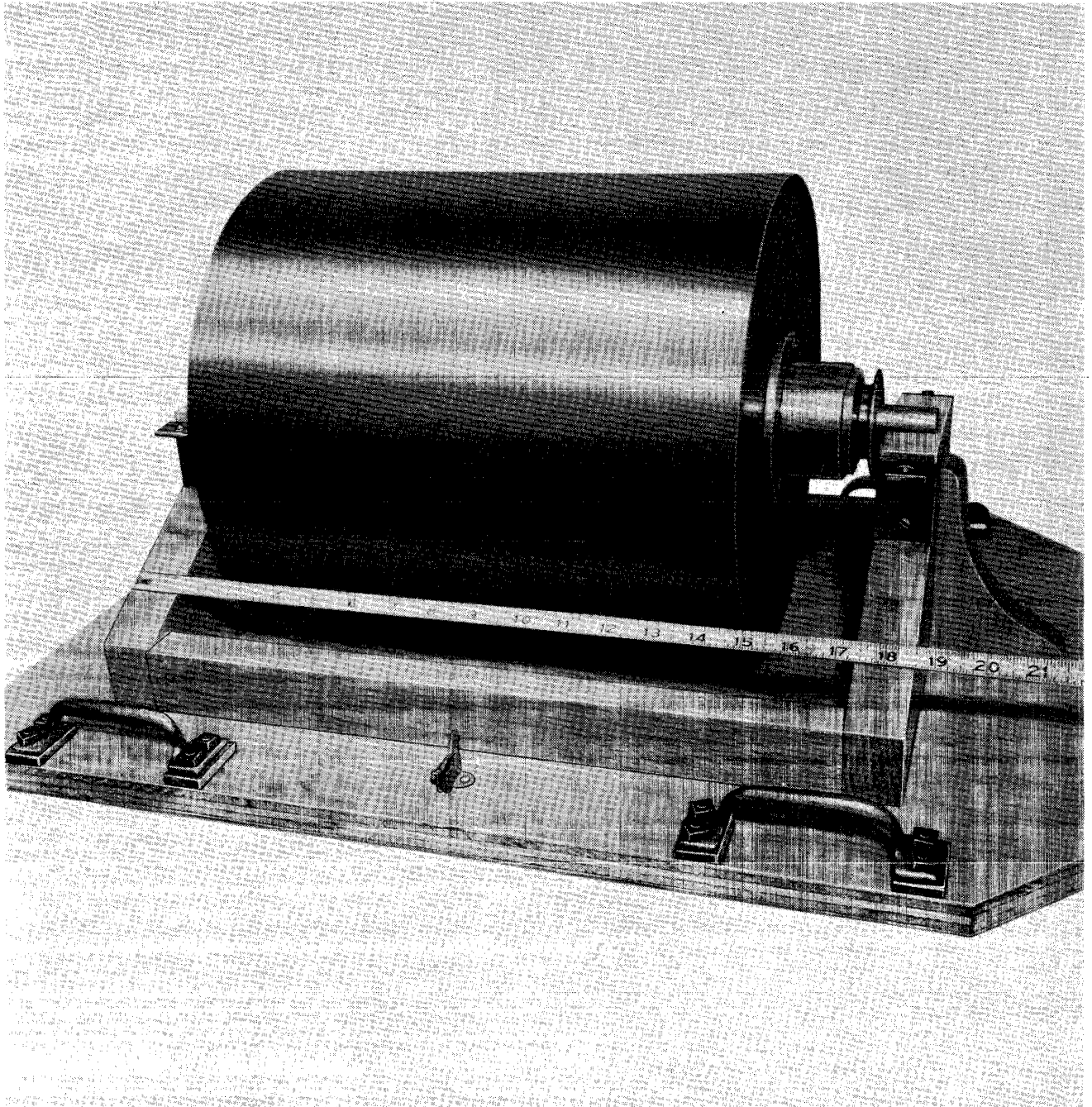


Figure 1-74. Magnetic Drum

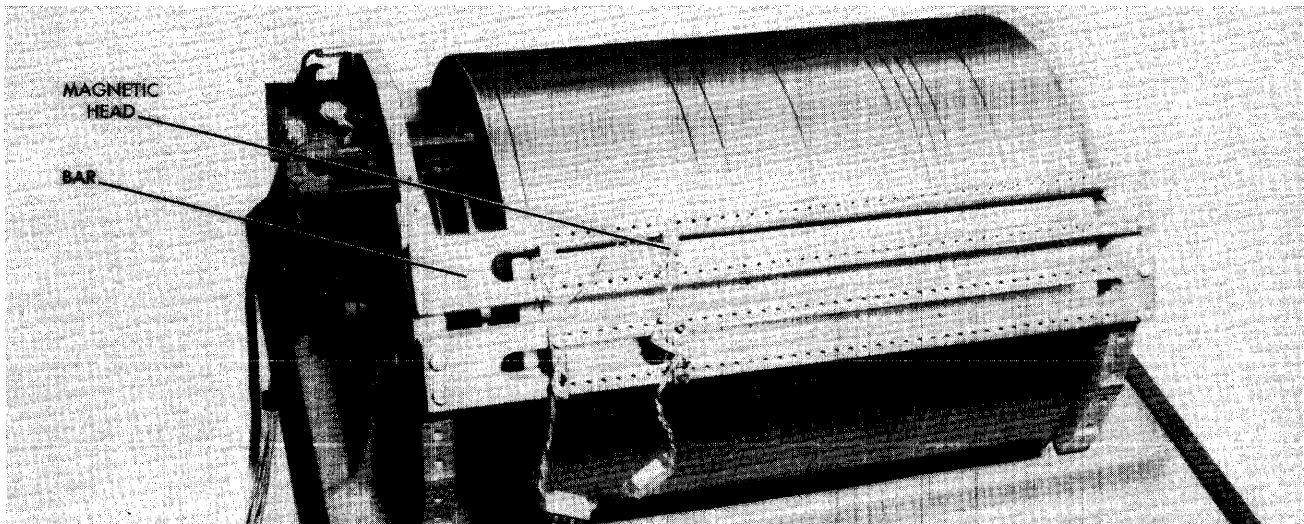


Figure 1-75. Magnetic Drum and Heads

the small electromagnetic flux pattern is positive, the bit is a binary 1; if the flux pattern is negative, it is a binary 0. Once written, a bit is stored on a drum until another bit is written over it or it is deliberately erased. Reading from a drum does not in any way alter the bits recorded on the drum surface.

As a drum rotates, closed circumferential bands of drum surface, called drum channels, pass under the magnetic heads (each channel is written or read by one head on the CD side or one head on the OD side). Since there can be as many as 40 magnetic heads on a bar, there can also be as many as 40 channels in a drum field. The rotational velocity of a drum and the timing of its reading and writing operations are such that each magnetic head can read or write 2,048 bits in each channel of a drum field.

Thirty-two drum channels of a field contain information bits, except on the radar data drum where only 24 of the channels in a field contain information bits. Each axial line of bits in a field is a drum register. Some fields have control channels, such as parity, marker status, and status channels. In addition, each drum has a timing channel and an index channel which are common to all fields on the drum.

During Drum System reading and writing operations, information is transferred to or from all bit positions in a drum register simultaneously by the 32 magnetic heads on one bar (which corresponds to the 32 information channels). The 32 bits which are transferred constitute a Drum System word. In addition to the 32-bit information word, a drum register also contains the control bits for the word. The times at which the control bits are transferred between the control channels and

their magnetic heads do not necessarily correspond to the times at which the words are transferred.

A number of words making up an organized piece of information are known as a message. One drum register or a group of consecutive drum registers containing the word or words of a complete message is called a slot. The term slot refers to the physical grouping of registers on a drum, while the term message refers to the logical grouping of words on the drum.

1.2.4 Timing

Each of the 12 drums of the Drum System is driven by its own synchronous motor at an individually regulated speed of 2,914 revolutions per minute. A drum therefore rotates completely once every 20.6 milliseconds. Drum System access time is the time required to locate and transfer information to and from a selected drum register. The length of time required for a selected drum register to come under the set of magnetic heads of its particular field can be anywhere from 0.0 to 20.6 milliseconds. Therefore, average access time is 10.3 milliseconds.

The radar data and track display drums are timed so that 2,060 registers are established in each field. The first 12 registers of each field cannot be used to store information; thus, the period between the midpoints of successive registers on these two drums is reduced to 10.0 microseconds.

With the exception of one circuit in the OD IC element, the circuits that develop timing information for use in Drum System operations are contained in the CD element. Circuits that develop timing information produce four standard pulses as each drum register passes

under the fixed magnetic heads. These pulses are called drum timing pulses; one series of four drum timing pulses constitutes a drum operation cycle. For simplicity, a drum operation cycle is considered as being of 10.0 microseconds duration for all drums, and the interval between successive drum timing pulses is therefore 2.5 microseconds.

When drum timing pulses are used to time the operations of the CD element, they are called CD 1, CD 2, CD 3, and CD 4 pulses. When drum timing pulses are used to time the operations of any of the OD elements, they are called OD 1, OD 2, OD 3, and OD 4 pulses.

Transfers to the drum are coincident with drum timing pulse 3, and transfers from the drum are coincident with drum timing pulse 1. Drum timing pulses 2 and 4 are used for control functions. In addition to their uses in the Drum System, the timing pulses are sent to all systems except the Central Computer System. In these other systems, timing pulses are used to control and time operations. The fact that other systems use the Drum System timing cycle automatically synchronizes their operations with the operations of the Drum System.

The relationship between the CD drum timing pulses of the Drum System and the Central Computer System is shown in figure 1-77; an understanding of this relationship is fundamental to comprehension of CD operations. It is to be noted particularly that the two timing sequences are independent of each other.

A Drum System CD operation cycle is of 10.0-microsecond nominal duration and contains four CD timing pulses at 2.5-microsecond intervals. A Central Computer System memory cycle is 6 microseconds long. Timing pulses occur at 0.5-microsecond intervals, and only 12 Central Computer System timing pulses, or 6 microseconds, are required to complete a Central Computer System machine cycle.

Normally, there are 2,048 drum operation cycles per drum revolution. There are also 2,048 drum operation cycles for the track display and radar data drums, since no reading or writing is performed during the first 12 of their 2,060 timing cycles.

A single 1 bit written in the index channel under the control of the Drum System manual control panel establishes the reference point for the addressing of all registers and for the counting of all revolutions of the drum during subsequent operations. For all but the auxiliary memory drums, the index channel is read by two magnetic heads, one on the OD side and one on the CD side. Since the auxiliary memory drums are written and read from the CD side only, the index channel on each of the auxiliary memory drums need be read by only one magnetic head. Therefore, there is an OD index pulse

and a CD index pulse for each drum except the auxiliary memory drums. On the CD side, the index channel is read at CD 1 time, and the drum register in which the 1 bit occurs is number 0000. The other drum registers are numbered consecutively to 2,047. On the OD side, the index channel is read at OD 3 time, and the drum register in which the 1 bit occurs is number 2,047.

1.3 SYSTEM LOGIC

The Drum System is logically divided into the OD

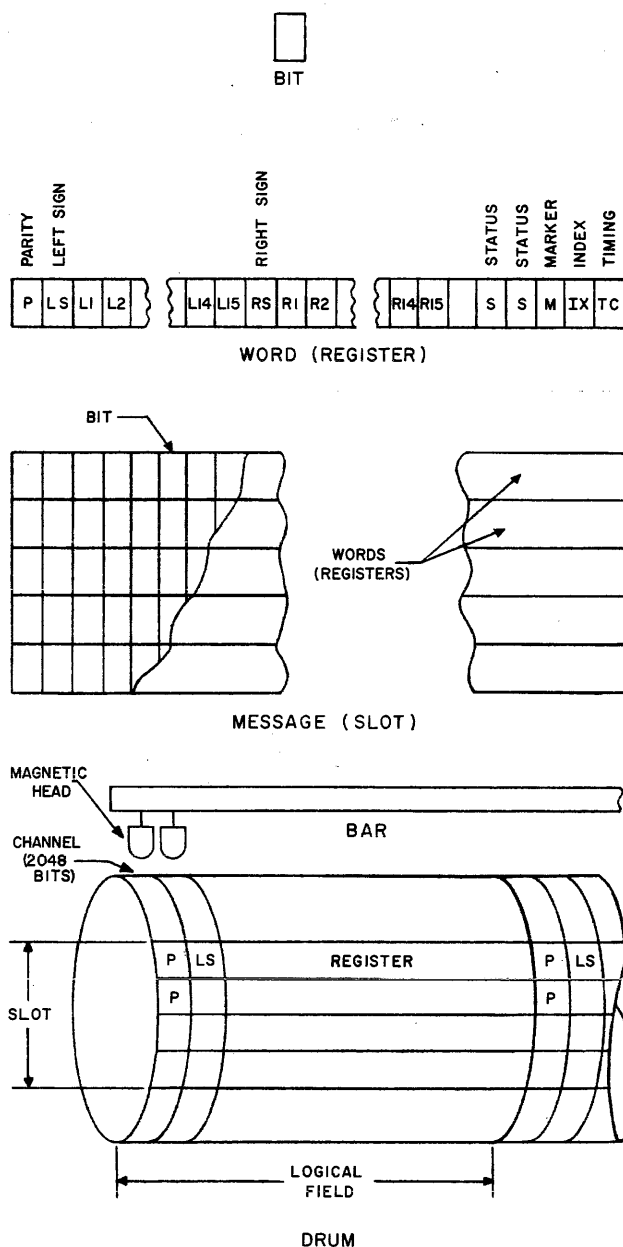


Figure 1-76. Drum System Information Units

input element, OD display element, OD output element, OD intercommunication element, CD element, auxiliary CD element, and the maintenance elements. (See fig. 1-78.)

The OD input element transfers input data from the Input System to the Drum System completely free of program control. This data is in relatively large quantities, at random time intervals, and at different rates.

The OD display element takes display data that has been stored on the display fields of the Drum System by the CD element and transfers this data to the Display System. The OD display element reduces the rate of transfer of display data so that the Display System can accommodate it.

The OD output element takes output data that has been stored on the output buffer fields of the LOG drum by the CD element and transfers the data to the Output

System. The OD output element transfers output data from the drum at a rate that the Output System can accommodate.

The OD intercommunication (IC) element provides a means of transferring information from one Central Computer System to the other. The information is written on the CD side of the IC field under program control of the transmitting Central Computer System. The information is read subsequently by the OD IC element under program control of the receiving Central Computer System. Since there is an IC field and an OD IC element in the Drum Systems of computer A and computer B, information transfer can take place in either direction. The CD element transfers information in both directions between the Central Computer System and the main drum group. The CD element also receives IC information directly from the OD IC element of the other computer. The auxiliary CD element is used during normal operation to

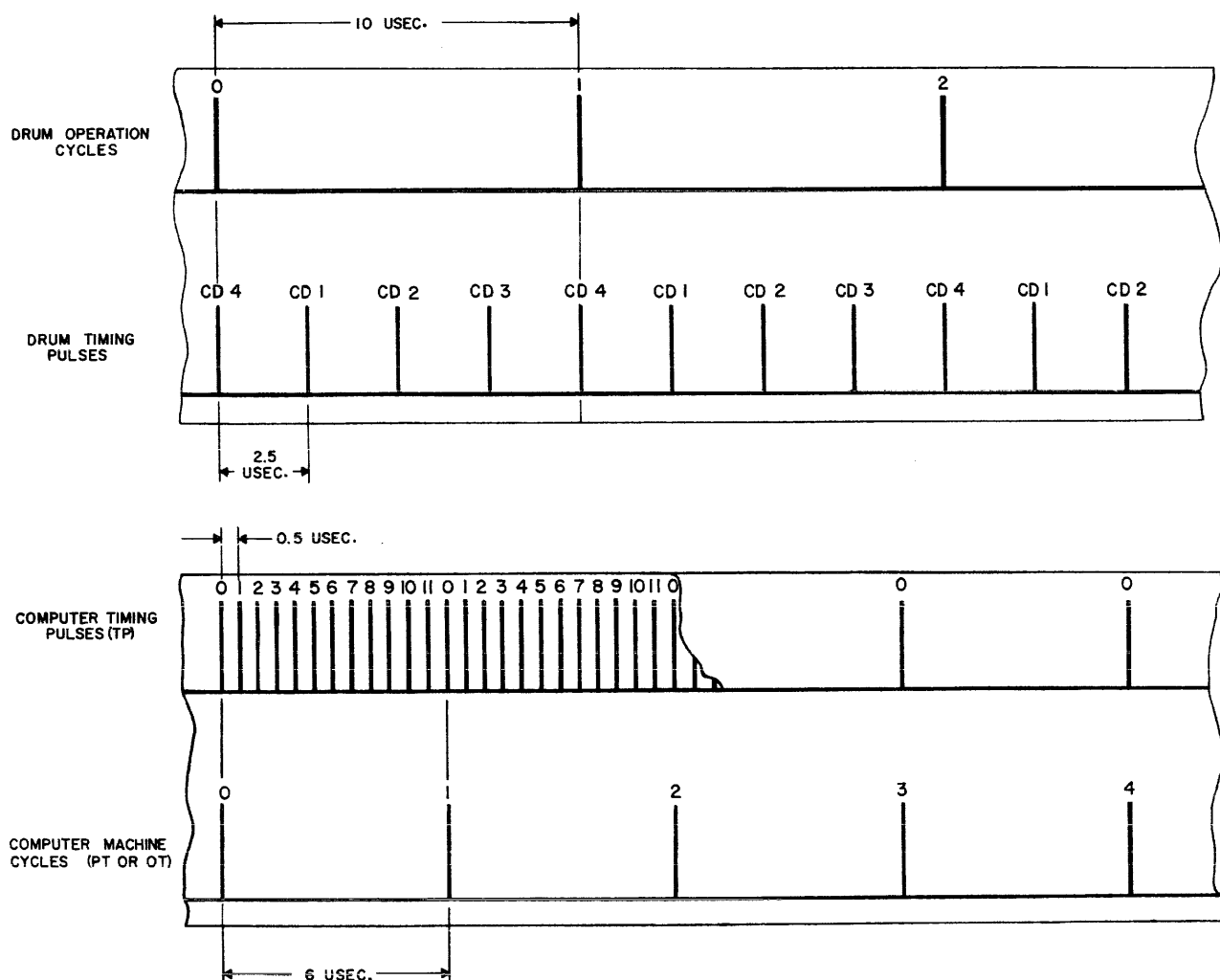


Figure 1-77. Timing Chart, Drum (CD Side) and Central Computer System

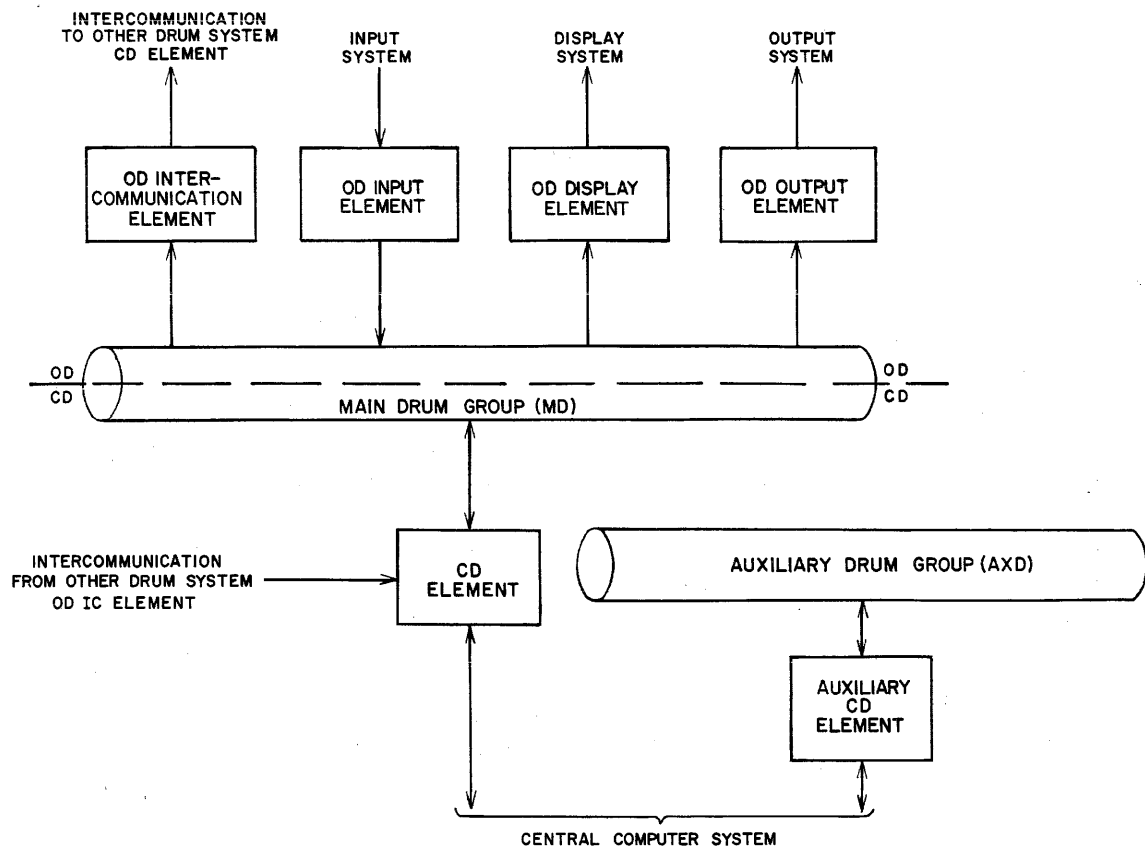


Figure 1-78. Drum System Logical Divisions

transfer information from the auxiliary drum group to the Central Computer System. The auxiliary drum group does not receive information during normal operation.

The computer test element provides the Central Computer System with a means of performing tests on other Drum System elements to detect any alteration of

data. The manual drum tester element enables maintenance personnel to exercise manual control over the main drum group operations for performance checking. The auxiliary manual drum tester provides a means for checking the operation of the auxiliary drum group and auxiliary CD element.

SECTION 2

SYSTEM OPERATION

2.1 GENERAL

The CD and OD operations of the Drum System are independent of each other. All CD operations are controlled by the program of the Central Computer System; OD operations are independent of program control and are predetermined by the design of the equipment. Those fields of the Drum System that are read or written on the OD side as well as on the CD side act as time buffers, whereas those fields that are read or written only on the CD side serve as auxiliary memory. Certain fields actually fulfill both functions.

As a time buffer, the Drum System serves to change rates of information transfer. For example, the Drum System collects and stores information arriving at random intervals from the Input System for rapid, periodic transfer to the Central Computer System under program control. Similarly, processed information is supplied to the Drum System by the Central Computer System at a rapid rate. The Drum System then transfers this processed information to the Output and Display Systems at a slower rate.

As an auxiliary memory device, the Drum System stores information such as processed data, subprograms, mathematical values, and constants to which the Central Computer System can refer. Information stored in the Drum System for transfer to the Display System must be compared at times to new information received by the Central Computer System. In such instances, that part of the Drum System devoted to display information serves as both a time buffer and auxiliary memory.

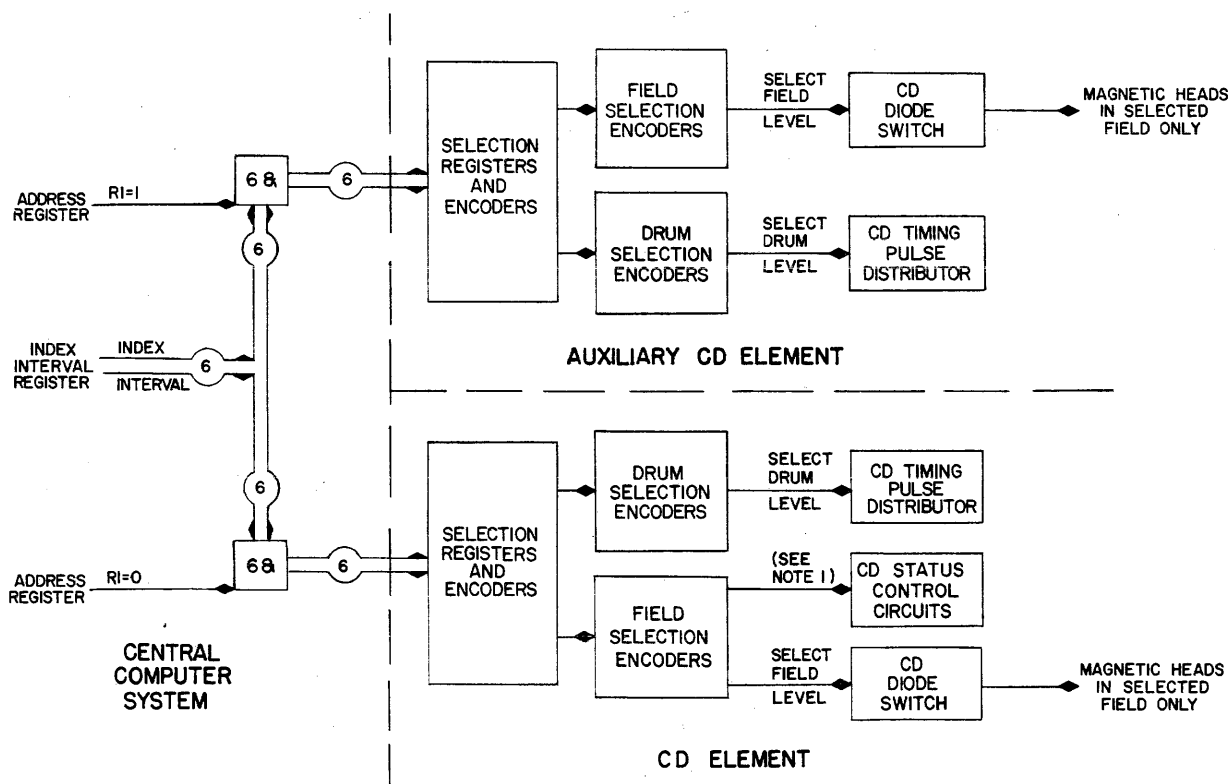
At any given time, the Central Computer System may require only a portion of the information stored in an auxiliary memory field. The required portion can be obtained by the Central Computer System if it has been written in a particular location in the field and that location in the field can be specifically selected for reading. To do this, each drum register in an auxiliary memory field is assigned an address which is used in selecting a specific drum register to be read or written. The mode of reading or writing on the CD side in a field whose drum registers have addresses is known as the address mode.

A field which serves solely as a time buffer need only store a word between the time of its writing on one side of the Drum System and the time of its reading on the other side. Since in this case a given word need be read only once, its location on the field is unimportant. Instead, provision must be made to prevent either writing a word over another word not yet read or reading a word twice. This is done by using two extra bits for each drum register on a field which is solely a time buffer. The two extra bits denote the status of the drum register. One bit indicates to the writing side of the field whether a word can be written into the drum register without destroying an unread word in that register. The other bit indicates to the reading side whether the drum register contains a word to be read. This mode of reading and writing is known as status mode.

Each drum field which receives or delivers information on the OD side is permanently connected to a given system. The mode of operation of each field is determined by the function it performs for the system to which it is connected. A field which is used solely as a time buffer uses the status mode, a field used solely as auxiliary memory uses the address mode, and a field which must be used as both a time buffer and auxiliary memory uses the address mode. A field written on the OD side cannot be read on the OD side. Conversely, a field read on the OD side cannot be written on that side. Thus, no selection of fields, mode of operation, or direction of transfer is necessary on the OD side.

In contrast to the OD side, only one drum field can be used in a CD operation at one time. Consequently, the Central Computer System program must select the drum group (main or auxiliary), the drum, and the field. No selection of address or status mode is necessary because the mode of operation of each field is determined by the function of that field. The selection of a field therefore implies selection of the mode of operation of that field.

The SDR (μ) instruction selects the drum group, the drum, and the field. Bit R1 determines whether (μ), the index interval which selects the drum and field, is supplied to the CD element or the auxiliary CD element.



NOTE:
LEVEL PRESENT ONLY
WHEN CHOOSING A FIELD
WRITTEN AND READ BY
STATUS

Figure 1-79. Drum and Field Selection

Thus, R1 effectively chooses the main or the auxiliary drum group. (See fig. 1-79.) When the index interval is received, a select-drum level and a select-field level are generated. The select-drum level enables the timing pulse distributor of the selected drum to supply the timing pulses which control the CD operation. The select-

field level enables reading or writing by the magnetic heads only in the selected field. When a field written and read by status is chosen (in the main drum group only), a level is generated by the field selection encoder which conditions the CD status control circuits. (Refer to table 1-64.)

TABLE 1-64. GROUP, DRUM, FIELD, AND MODE SELECTION CODES

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	02	Auxiliary memory 1	Address	Read, write
0	03	Auxiliary memory 2	Address	Read, write
0	04	Auxiliary memory 3	Address	Read, write
0	05	Auxiliary memory 4	Address	Read, write
0	06	Auxiliary memory 5	Address	Read, write
0	07	Auxiliary memory 6	Address	Read, write
0	10	Auxiliary memory 7	Address	Read, write

TABLE 1-64. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	11	Auxiliary memory 8	Address	Read, write
0	12	Auxiliary memory 9	Address	Read, write
0	13	Auxiliary memory 10	Address	Read, write
0	14	Auxiliary memory 11	Address	Read, write
0	15	Auxiliary memory 12	Address	Read, write
1	41	Auxiliary memory 13	Address	Read, write*
1	42	Auxiliary memory 14	Address	Read, write*
1	43	Auxiliary memory 15	Address	Read, write*
1	44	Auxiliary memory 16	Address	Read, write*
1	45	Auxiliary memory 17	Address	Read, write*
1	46	Auxiliary memory 18	Address	Read, write*
1	51	Auxiliary memory 19	Address	Read, write*
1	52	Auxiliary memory 20	Address	Read, write*
1	53	Auxiliary memory 21	Address	Read, write*
1	54	Auxiliary memory 22	Address	Read, write*
1	55	Auxiliary memory 23	Address	Read, write*
1	56	Auxiliary memory 24	Address	Read, write*
1	61	Auxiliary memory 25	Address	Read, write*
1	62	Auxiliary memory 26	Address	Read, write*
1	63	Auxiliary memory 27	Address	Read, write*
1	64	Auxiliary memory 28	Address	Read, write*
1	65	Auxiliary memory 29	Address	Read, write*
1	66	Auxiliary memory 30	Address	Read, write*
1	71	Auxiliary memory 31	Address	Read, write*
1	72	Auxiliary memory 32	Address	Read, write*
1	73	Auxiliary memory 33	Address	Read, write*
1	74	Auxiliary memory 34	Address	Read, write*
1	75	Auxiliary memory 35	Address	Read, write*
1	76	Auxiliary memory 36	Address	Read, write*
1	02	Auxiliary memory 37	Address	Read, write*
1	03	Auxiliary memory 38	Address	Read, write*
1	04	Auxiliary memory 39	Address	Read, write*
1	05	Auxiliary memory 40	Address	Read, write*

*Write only if write interlock switch on duplex maintenance console is unlocked.

TABLE 1-64. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
1	06	Auxiliary memory 41	Address	Read, write*
1	07	Auxiliary memory 42	Address	Read, write*
1	10	Auxiliary memory 43	Address	Read, write*
1	11	Auxiliary memory 44	Address	Read, write*
1	12	Auxiliary memory 45	Address	Read, write*
1	13	Auxiliary memory 46	Address	Read, write*
1	14	Auxiliary memory 47	Address	Read, write*
1	15	Auxiliary memory 48	Address	Read, write*
0	24	Crosstelling input	Status	Read, test write
0	25	Crosstelling input	Identity (R11-R15)	Read
0	40	Crosstelling marker (single channel)	Address	Write (contents of LS only)
0	27	Digital display	Address	Read, write
0	17	Digital display	Identity (R14-R15)	Test read
0	32	Gap-filler input	Status	Read, test write
0	33	Gap-filler input	Identity (R11-R15)	Read
0	16	Intercommunication (other)	Address	Read
0	26	Intercommunication (own)	Address	Read, write
0	76	Intercommunication (own)	Address	Test read
0	34	Long-range radar input 1	Status	Read, test write
0	35	Long-range radar input 1	Identity (R12-R15)	Read
0	50	Long-range radar input 1	Identity (R7-R15)	Read
0	36	Long-range radar input 2	Status	Read, test write
0	37	Long-range radar input 2	Identity (R12-R15)	Read
0	51	Long-range radar input 2	Identity (R7-R15)	Read
0	22	Manual input	Status	Read, test write
0	23	Manual input	Identity (R14-R15)	Read

*Write only if write interlock switch on duplex maintenance console is unlocked.

TABLE 1-64. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	30	Output buffer odd	Status	Write
0	31	Output buffer even	Status	Write
0	30	Output buffer	Identity (R14-R15)	Test read
0	31	Output buffer	Status	Test read
0	60	Radar data 1	Address	Read, write
0	61	Radar data 2	Address	Read, write
0	62	Radar data 3	Address	Read, write
0	63	Radar data 4	Address	Read, write
0	64	Radar data 5	Address	Read, write
0	65	Radar data 6	Address	Read, write
0	66	Radar data 7	Address	Read, write
0	67	Radar data 8	Address	Read, write
0	70	Radar data 9	Address	Read, write
0	47	Situation display	Identity (R5-R10)	Test read
0	20	Spare 1	Address	Read, write
0	21	Spare 2	Address	Read, write
0	41	Track display 1	Address	Read, write
0	42	Track display 2	Address	Read, write
0	43	Track display 3	Address	Read, write
0	44	Track display 4	Address	Read, write
0	45	Track display 5	Address	Read, write
0	46	Track display 6	Address	Read, write

*Write only if write interlock switch on duplex maintenance console is unlocked.

Execution of the *SDR (u)* instruction introduces a 120-microsecond delay before any transfer of information is started to allow the reading and writing circuits to settle. The delay is started by the deselect pulse generated upon execution of the *SDR (u)* instruction to deselect the IO unit involved in the previous IO operation. A subsequent *RDS n* or *WRT n* instruction is needed to start the CD operation and may introduce another delay if it changes the previously selected direction of transfer.

Since an *SDR* instruction always selects reading as the direction of transfer, a *WRT* instruction following

SDR introduces another delay to allow switching of the magnetic heads to a write condition. However, if an *RDS* instruction follows *SDR*, it is not necessary to introduce another delay because the magnetic heads are already conditioned to read. Table 1-65 presents the possible combinations of *SDR*, *RDS*, and *WRT* instructions and the minimum delay introduced in each case.

In cases 4 through 7, it must be assumed that an *SDR* instruction is given some time prior to the instruction sequence listed, and that no *SDR* instruction is inserted between the two instructions in each case. If an *SDR* instruction is given between the two instructions, then case 1 or 2 applies.

When the *SDR (u)* instruction chooses an addressable field, the following *RDS n* or *WRT n* instruction specifies the exact number of words to be transferred. After all of the words have been transferred, the IO word counter produces a computer-disconnect pulse which clears the IO interlock and ends the IO operation. However, the number of words to be read from a time buffer field using status mode cannot be determined prior to the execution of the *RDS n* instruction, nor can the number of drum registers which are available for writing be ascertained in a field using status mode prior to the execution of the *WRT n* instruction.

To prevent a continuous searching of the selected field until the required number of words has been transferred, the Drum System makes use of a disconnect counter in the CD element. The disconnect counter insures that each drum register in the selected field is examined only once for each IO operation. Execution of the *RDS n* or *WRT n* instruction starts the disconnect counter counting the CD 4 pulse associated with each drum register. If the IO word counter has not generated a disconnect pulse after all drum registers in the selected field have been examined, the disconnect counter produces a drum-disconnect pulse which clears the IO interlock and ends the IO operation. The IO word counter must be set to a number greater than 2,048 by the *RDS n* or *WRT n* instruction to prevent generating a computer-disconnect pulse before all drum registers have been searched.

2.2 TIME BUFFER OPERATION

2.2.1 Writing

2.2.1.1 Status

Writing by status mode enables a drum field to receive data from a number of input sources at random intervals and different repetition rates, with a maximum probability of storing all available information. On drums containing status controlled input fields, the OD and CD status channels are added to the word channels needed for storing information. The OD status channel is written on the CD side and read on the OD side. Conversely, the CD status channel is written on the OD side and read on the CD side. The contents of the OD status channel indicate whether drum registers are available for storing input words. The contents of the CD status channel denote whether individual drum registers contain words that can be read into the Central Computer System under program control when requested.

An OD status control circuit associated with each of the fields written by status reads the contents of the OD status channel. A 0 bit in the OD status channel indicates that the register with which it is associated is either empty or contains a word that has already been read by the Central Computer System. A 1 bit in the OD status

TABLE 1-65. DELAYS IN DRUM IO OPERATION

INSTRUCTION SEQUENCE	DELAY MEASURED FROM	MINIMUM AMOUNT OF DELAY (microseconds)
1. <i>SDR</i>	<i>SDR</i>	120
2. <i>SDR</i> <i>WRT</i>	<i>WRT</i>	120
3. <i>SDR</i> <i>RDS</i>	<i>SDR</i>	120
4. <i>RDS</i> <i>WRT</i>	<i>WRT</i>	120
5. <i>WRT</i> <i>RDS</i>	<i>RDS</i>	120
6. <i>RDS</i> <i>RDS</i>		0
7. <i>WRT</i> <i>WRT</i>		0

channel indicates that the register contains a word that has been written during a previous drum revolution and has not yet been read by the Central Computer System. When a 0 bit is detected in the OD status channel, the OD status control circuit allows a word to be written (if available) into the drum register associated with the bit. At the same time, a 1 bit is written in the CD status channel. If no information to be written on the drum is available at this time, the OD status control circuit writes a 0 bit in the CD status channel, notifying the CD side of the drum that the register contains no information.

The logical operation of all fields using the status mode of writing is nearly identical. (See fig. 1-80.) The magnetic head that reads the OD status channel of a field is physically offset ahead of the heads that write in the field by an amount equal to the distance traveled by a drum register in 10.0 microseconds. Thus, the OD input element knows the status of an input field register 10.0 microseconds before that register starts to pass under the heads which write in it.

As an illustration of input information writing, assume that a register of a particular field is empty. At OD₁ 1 time, the magnetic head on the OD side which reads the OD status channel detects a 0 bit.

At OD₁ 3 time, the OD status control circuit sends a demand pulse to the Input System, requesting information. If no information is available, there is no response. However, if information is available, the Input System

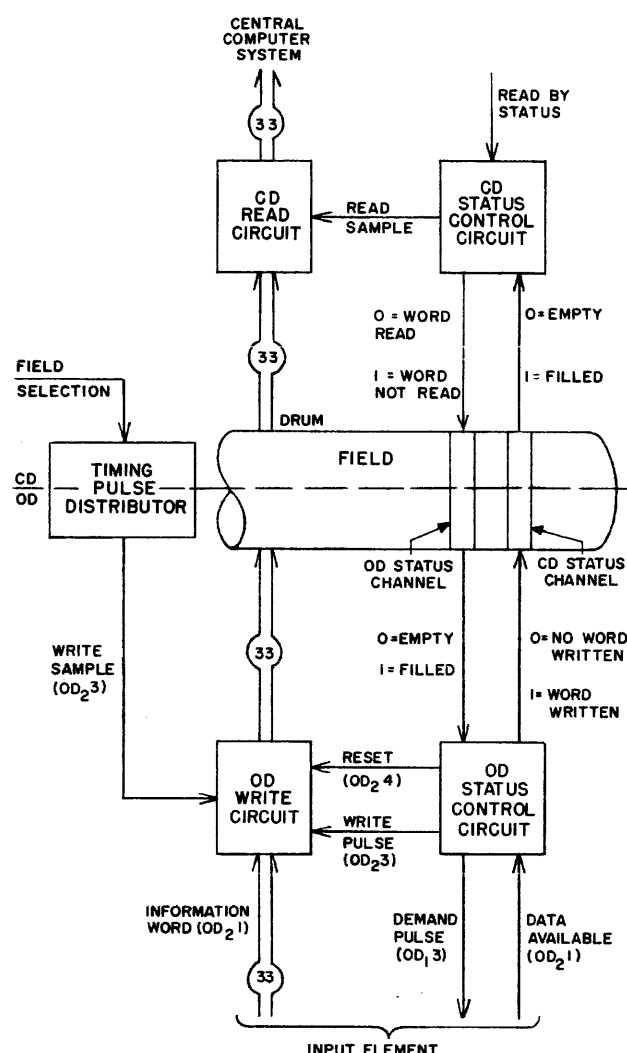


Figure 1-80. Writing by Status

delays until $OD_2 1$ time, and then sends a data-available pulse to the OD status control circuit. At the same time ($OD_2 1$), an information word is transferred to the OD write circuit.

At $OD_2 3$ time, after the 0 bit in the OD status channel is detected, the status control circuit sends a write pulse to the OD write circuit. This write pulse transfers the information word from the write circuit to the magnetic heads and into the selected drum register during the period of the write-sample pulse from the timing circuit. At the same time ($OD_2 3$), the status control circuit causes a 1 bit to be written in the CD status channel. At $OD_2 4$ time, a reset pulse prepares the write circuit for the receipt of the next information word.

The 1 bit written in the CD status channel provides the means of notifying the CD side of the drum that information is available in that particular register. When the 1 bit in the CD status channel is detected by

the CD status control circuit, the CD read circuit reads the word. If the word is accepted by the Central Computer System, a 0 bit is written in the OD status channel by the CD status control circuit. This notifies the OD status control circuit that the information in that particular register has been accepted by the Central Computer System, and new information can be written in the register.

If the word is not accepted by the Central Computer System, a 1 bit is written in the OD status channel. This notifies the OD status control circuit that the information in that particular drum register has not yet been accepted by the Central Computer System, and the drum register cannot be rewritten. All reading operations on the CD side are performed under program control by the Central Computer System. The transfer of information on the CD side is covered more fully in the discussion on status identification. (Refer to 2.2.2.1.)

2.2.1.2 Modified Status

A modification of the status mode is used when writing messages which contain two words. In this mode of writing, the OD status control circuit will inspect every other status bit in the OD status channel. This effectively divides the field into slots of two registers each.

When a 0 bit is detected in the OD status channel by the OD status control circuit, it signifies the first register of an empty slot. The OD status control circuit then sends a demand pulse to the Input System to determine whether a message is available for writing in the empty slot. If a message is available, the Input System returns two data-available pulses, at 10.0-microsecond intervals, each in conjunction with a word of the message. The data-available pulses cause the writing of each word in the drum registers and the writing of 1 bits in the associated CD status channel. The CD element reads a field by modified status in the same manner in which it is written; i.e., only the status bit associated with the first register of each slot is inspected.

2.2.1.3 Marker Status

Messages that are three words in length are written on a drum field by a modification of the status mode known as marker status. In addition to the OD and CD status channels, a third channel, called the marker status channel, is provided. The marker status channel is written before the start of normal drum operation by the CD element under program control of the Central Computer System. A 1 bit, called a marker, is written in the marker status channel adjacent to every third register. The markers in the channel indicate to the OD marker status control circuit the drum register in which the first word of a three-word message is to be written. The remaining two words are written in the two regis-

ters immediately following the register associated with the marker bit.

As the drum rotates, the marker and OD status channels are read continuously. When a 0 is read in the OD status channel (empty register) at the same time that a marker is read in the marker status channel (first register of a slot), the marker status control circuit produces a demand pulse. The demand pulse determines whether a three-word message is available to be written on the drum. If a message is available, the OD input element receives three consecutive data-available pulses at 10.0-microsecond intervals. Simultaneous with each data-available pulse, the OD input element transfers a word of the three-word message to the OD write circuit. Each data-available pulse produces a write pulse which causes the writing of an individual word into a drum register. At the same time, 1 bits are written in the CD status channel next to each of the three drum registers.

2.2.2 Reading

2.2.2.1 Status Identification

The source of type of information written by status on the OD side of the Drum System is indicated by identity bits contained within the message. By reading all messages which contain the same identity bits, the messages can be assembled by the Central Computer System into tables, each capable of being processed by an iterative program. In this way, the Drum System acts as a sorting device as well as a storage medium.

To read information from a particular source or of a particular type, a modification of reading by status is used; this is called reading by status identification. Reading by status identification compares the identity bits contained within a message with identity bits specified by the Central Computer System. The identity bits specified by the Central Computer System are contained in the right half-word of the SDR instruction. When the SDR instruction is executed, the identity bits are loaded into the drum control register and a 120-microsecond delay is introduced. (Refer to 2.1.) Messages will be accepted by the Central Computer System only if the identity bits within the message correspond to those in the drum control register. If they do not compare, the message is not accepted by the Central Computer System.

A 1 bit detected in the CD status channel indicates the presence of a word in the associated drum register that has not been read. This causes the CD status control circuit to send a read-sample pulse to the CD read-write control circuit (See fig. 1-81.) The read-sample pulse is supplied to the CD read circuit as a read-sample pulse and to the comparison circuit as a compare pulse. The CD read circuit then transfers the word in the drum register to the IO buffer register.

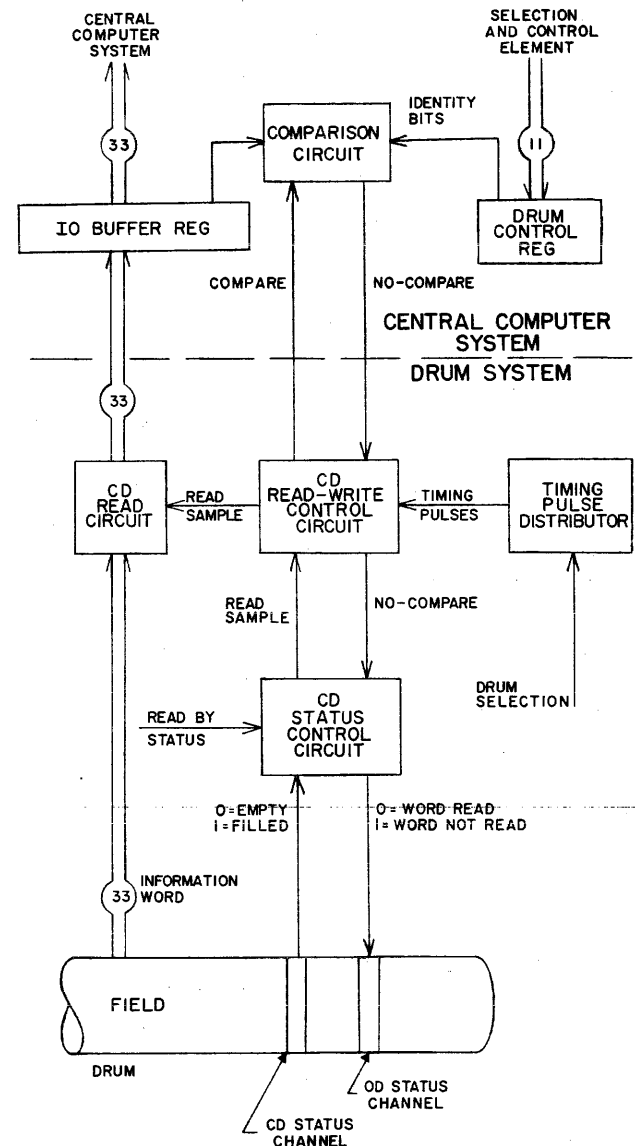


Figure 1-81. Reading by Status-Identification

The compare pulse requests the Central Computer System to compare the identity bits in the word now in the IO buffer register with the identification bits in the drum control register. If the comparison fails, the Central Computer System rejects the word by sending a no-compare pulse to the CD status control circuit via the CD read-write control circuit. Upon receipt of the no-compare pulse, the CD status control circuit causes a 1 bit to be written in the OD status channel to prevent a new word from being written in that drum register on the OD side.

If the comparison is successful, the word in the IO buffer register is accepted and transferred to the IO register for storage in core memory. The no-compare pulse is not sent to the Drum System. The CD status control

circuit then writes a 0 bit in the OD status channel to permit a new word to be written on the OD side in the drum register that has just been read. When reading messages which contain more than one word (written on the OD side by modified or marker status), only the first word contains identity bits. If the Central Computer System accepts the first word, the remaining words in the message are accepted automatically.

2.2.2.2 Status

When desired, it is possible to transfer unsorted blocks of information from the Drum System to the Central Computer System by reading the information by status. In this mode of reading, the Central Computer System accepts all information written in those registers associated with a 1 bit in the CD status channel. The CD element transfers each word to the IO buffer register in the same manner as that used in reading by status identification, but the no-compare pulse is suppressed by the Central Computer System. Thus, each word is accepted as it is read.

In addition to specifying the field to be read, the index interval in the *SDR* instruction also specifies the mode of reading; i.e., status or status identification. The *RDS* instruction requests more words to be read than expected to be in the field. In this way, the disconnect pulse will be generated by the disconnect counter in the Drum System after all the words in the field have been read.

2.3 AUXILIARY MEMORY OPERATION

2.3.1 Writing

2.3.1.1 Address

Status control techniques for the transfer of data between the Drum System and the Central Computer System are satisfactory for reading or writing randomly occurring data that is read only once. However, these methods are unsatisfactory when specific data must be referred to from time to time. Specific items of program data such as subprograms, mathematical tables, constants, or processed information which is temporarily stored outside of core memory must be available for reference whenever needed. To be accessible for reference this information must be stored in a specific order in specific locations in the Drum System. Writing by address control satisfies this requirement.

In writing by address control, the Central Computer System specifies the particular drum field in which a message is to be written, the particular register in which writing is to begin, and the number of words to be written. This is done by programming three IO class instructions: *LDC x*, *SDR (u) r*, and *WRT n*. The right half-word of the *LDC* instruction, *x*, specifies the first address in core memory from which information is to be transferred to the Drum System. The *SDR (u) r*

instruction designates the drum, the field, and the specific register in which writing is to begin. (Refer to 2.1.) Bits L10 through L15, designated as (*u*), indicate the drum and field, while bits R5 through R15, designated as *r*, indicate the specific register. Since 11 bits can represent any binary number whose decimal equivalent is 0 to 2,047, bits R5 through R15 (11 bits) can represent the address of any one of 2,048 registers.

A drum register address actually represents the angular displacement of the register from a reference location called the index point. The address is determined by a counter, called the angular position counter (APC), which is stepped by 1 as each register in the field passes the writing position. Since the counter is reset by the index pulse which occurs once per revolution, each drum register of an addressable field has an assigned address within the range of 0 to 2,047. For the same reasons that the status channel read heads are physically offset 10.0 microseconds ahead of the reading and writing heads, the APC is set 10.0 microseconds (one drum register) ahead of the drum register whose address it contains. Thus, the APC contains the address of a register 10.0 microseconds before the particular drum register passes under the read or write heads. By supplying the contents of the APC to the Central Computer System, the Central Computer System can compare the address of the register about to come into writing position with the drum register address specified by the *SDR (u) r* instruction. If the comparison is successful, the first word is written in the drum register and the remaining words are written into consecutive drum registers without further address comparison.

With the execution of the *WRT* instruction, the Central Computer System sends a start-write pulse to the Drum System and initiates two break cycles. During the first break cycle, the first word to be written is transferred from core memory to the IO register. During the second break cycle, the first word is transferred from the IO register to the write circuit, and the second word to be written is transferred from core memory to the IO register. When the first word is transferred to the write circuit, the Central Computer System sends an IO-register-to-write-register pulse to the read-write control circuit. (See fig. 1-82.) The IO-register-to-write-register pulse notifies the read-write control circuit that a word has been transferred to the write circuit.

The start-write pulse (generated during the execution of the *WRT* instruction) conditions the read-write control circuit to send a start-compare level to the timing pulse distributor after a 120-microsecond delay which allows the field and drum selection circuits to settle. (Refer to 2.1.) The timing pulse distributor receives the start-compare level and generates a read-out-angular-position-counter pulse. This pulse transfers the



A modification of writing by address is known as writing by address with interleave. In writing with interleave, a prescribed number of drum registers are skipped between the writing of specific drum registers. The interleave pattern is selected by bits L13 through L15 of the *WRT* instruction; words can be written in every 8th, 16th, or 64th register, as selected. The drum control register, which initially contains the address of the first word to be written, is stepped by the selected interleave pattern after the transfer of each word. Comparison of the angular position counter (APC) contents and the address in the drum control register is made for each word and must be successful for each word written.

The maximum number of words which can be written by interleave in a single IO operation is determined by the selected interleave pattern. If the interleave pattern specifies the skipping of 16 registers between each word written, the maximum number of words which can be written is 128 words (2,048 divided by 16). In order to completely fill the drum field being written with this

Figure 1–82. Writing by Address

contents of the angular position counter to the right IO buffer register. The same pulse is also sent to the read-write control circuit as a compare-address pulse. The compare-address pulse is then sent from the read-write control circuit to the comparison circuit to effect comparison of the contents of the right IO buffer register with the contents of the drum control register. The drum control register now contains the drum register address specified by the SDR (u) r instruction.

When comparison between the right IO buffer register and the drum control register is unsuccessful, a no-

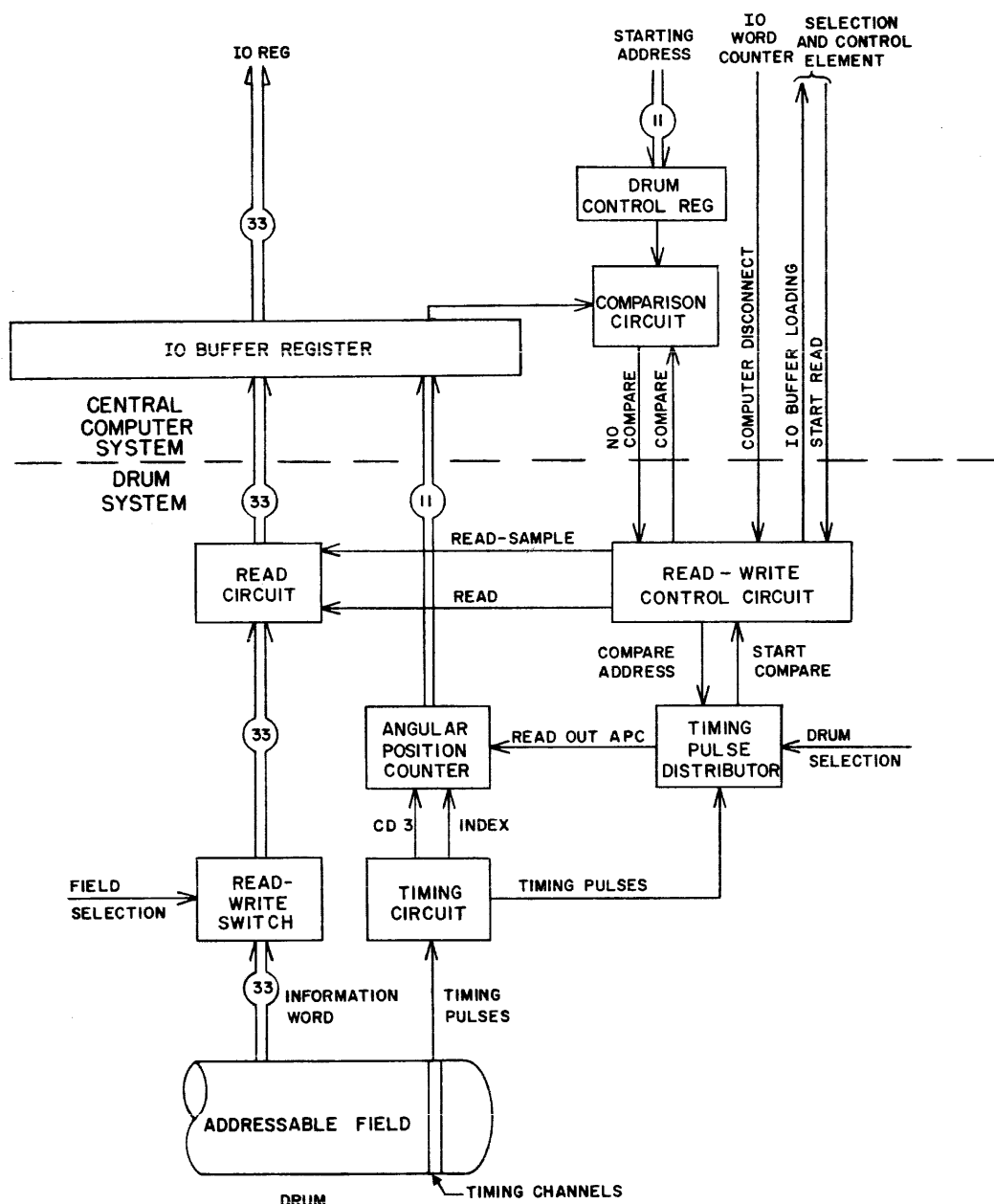


Figure 1-83. Reading by Address

particular interleave pattern, 16 IO operations must be scheduled. Writing by address with interleave can take place only on the CD side of the Drum System.

2.3.2 Reading

2.3.2.1 Address

The logical concept of full and empty drum registers employed in status mode is not applicable to reading by address where the same words can be read to the Central Computer System many times during normal operation.

Three IO class instructions are required in reading by address: *LDC x*, *SDR (u) r*, and *RDS n*. The

right half-word of the *LDC* instruction, designated as *x*, specifies the first address in core memory into which information is transferred from the Drum System. The *SDR (u) r* instruction designates the drum, the field, and the specific drum register in which reading is to begin. (Refer to 2.1.) As in writing by address, a successful comparison of the contents of the APC and the drum register address specified by the *SDR (u) r* instruction must be made before the first word is transferred from the Drum System to the Central Computer System. If the comparison is successful, the first word is read to the Central Computer System. The remaining words are then read from consecutive registers without further ad-

dress comparison until the number of words specified by RDS n have been read.

In reading by address, the 120-microsecond delay which allows the field and drum selection circuits to settle is initiated by the execution of the SDR (u) r instruction. With the execution of the RDS n instruction, the Central Computer System sends a start-read pulse to condition the read-write control circuit. (See fig. 1-83.) At the end of the 120-microsecond delay, the read-write control circuit produces a read level which conditions the read circuit and sends a start-compare level to the timing pulse distributor. The timing pulse distributor then produces a readout-APC pulse and a compare-address pulse. The read out-APC pulse transfers the contents of the APC to the right IO buffer register. The compare-address pulse is then sent to the comparison circuit via the read-write control circuit to effect comparison of the contents of the right IO buffer register with the contents of the drum control register. The drum control register now contains the drum register address specified by the SDR (u) r instruction and the right IO buffer register contains the address of the drum register coming under the read heads.

When comparison between these two addresses is unsuccessful, a no-compare pulse is generated by the comparison circuit. The no-compare pulse prevents the generation of the read-sample pulse in the read-write control circuit. Until address comparison is successful, the comparison circuit returns a no-compare pulse for each compare pulse received.

A successful address comparison between the right IO buffer register and the drum control register suppresses the no-compare pulse, permitting the generation in the read-write control circuit of the read-sample pulse. This pulse causes the word under the read heads to be transferred to the IO buffer register. The right IO buffer register is cleared after the address comparison is made, thereby allowing receipt of the first word. The read-sample pulse also generates an IO-buffer-loading pulse which notifies the Central Computer System that the first word is in the IO buffer register. The word in the IO buffer register is automatically transferred to the IO register and a break cycle is then initiated to transfer the word to core memory.

Each of the remaining number of words to be read is transferred to the IO buffer register as it comes under the read heads. When all of the words have been read, the Central Computer System sends a disconnect pulse which clears the read-write control circuit. No further reading can take place until a new start-read pulse conditions the read-write control circuit.

2.3.2.2 Interleave

Reading of an addressable field on the CD side can also be performed by interleave. The interleave pattern is

selected and the operation is carried out in the same manner as that employed in writing by interleave. (Refer to 2.3.1.2.)

2.3.2.3 Precession

Those fields used as a time buffer between the Central Computer System and the Display System are also used as auxiliary storage by the Central Computer System and are therefore written and read on the CD side by address. If these fields were read by address on the OD side, the maximum rate of transfer of information would exceed the capacity of the Display System to accept the information. Therefore, a method of reading is used which slows the rate of transfer of information on the OD side and reads all of the information in the field in one continuous operation. The modification of reading by address necessary to fulfill these requirements on the OD side is known as precession. Precession is the process of reading words from an addressable drum field whereby every n th register or slot is read until all of the words in the field have been read (necessitating a number of drum revolutions). For the following discussion of precession, consider an addressable field where, on the OD side, 63 registers are skipped between successive readings of single registers.

Reading by precession must start from the index point. At the start of a reading operation, a scale-of-64 counter in the Drum System is conditioned by the index pulse which, on the OD side, occurs at OD 3 time. The counter then counts each drum register (starting from the first register) as it passes under the read heads by counting the OD 2 pulses associated with each drum register. When the 63rd drum register is counted, the contents of the next drum register to pass under the read heads are transferred at OD 1 time, to the Display System and the register counter is returned to 0. The register counter again counts the drum registers as they pass under the read heads, and again the 64th register is read. In this manner, 63 drum registers are skipped, the 64th is read, 63 are skipped again, the 128th read, and so on in the same reading pattern. In one drum revolution, therefore, 32 words are read (2,048 divided by 64).

If the same pattern is used for the second drum revolution, the same registers would be read again. This is avoided by counting the index pulse as well as the OD 2 pulse during the 2,048 register. This means that the register counter will reach the count of 63 one count sooner than on the previous revolution, and that the 63rd drum register from the index point will be read on the second drum revolution.

The register counter is then stepped normally for the rest of the revolution, and each register read during this revolution immediately precedes each register read during the previous revolution. For each succeeding

revolution, the reading pattern is moved back one register. At the end of the 64th revolution, the index pulse coincides with the maximum count (63) in the register counter. This condition produces a pulse which ends the reading operation.

In some applications, reading by precession involves the skipping of a prescribed number of slots and then the reading of a specified slot which contains more

than one register. In order to avoid overlapping the previous reading pattern, the reading pattern is precessed by the number of registers in a slot at the start of the second drum revolution and at the start of each successive revolution. If more than one field is to be read, field switching is performed when one field is completely read. Reading of the next field by precession begins automatically.

SECTION 3

INFORMATION FLOW

3.1 OD INFORMATION TRANSFERS

3.1.1 Input System Information

Information supplied by the Input System is of four types, each from a separate element of the Input System:

- a. Manual input data from the manual input (MI) element
- b. Crosstelling data from the crosstell (XTL) input element
- c. Long-range radar data from the long-range radar input (LRI) element
- d. Gap-filler data from the gap-filler input (GFI) element

The information from each element is written onto a separate drum field by an independent writing circuit in the OD input element. The flow of input data through the Drum System is shown in figure 1-84.

Manual input data, which includes information prepared for insertion via punched cards and information gated out of the Display System, passes through the OD input element to be written on the manual input field by status mode. (Refer to 2.2.1.1.) It should be noted that no parity bit accompanies a manual input drum word. However, each manual input word contains a number of bits identifying the type of information it contains and the source of that information.

Crosstelling data, which comprises messages received from adjacent centrals, passes through the OD input element to be written onto the XTL input field. Since each XTL message requires three drum words, XTL information is written by the marker status mode. (Refer to 2.2.1.3.) The first word of each XTL message contains bits indicating the time at which the message is written onto the XTL field and bits identifying the source of the message.

Long-range radar data, which includes either range and azimuth reports on targets or height-above-ground reports on targets, passes through the OD input element to be written onto one of the two LRI fields. The choice of the LRI field is made by the LRI element of the Input System. Since each LRI message is two drum words in length, it is written by the modified status mode. (Refer to 2.2.1.2.) Both words contain bits indicating both the time at which they are written and the source of the message.

Gap-filler data, which comprises range and azimuth reports on targets, is written onto the GFI field by the OD input element by status mode. Each drum word containing a message also contains bits indicating the time of message writing onto the drum field and other bits identifying the source of the message. No parity bit is included with a GFI message on the drum fields.

Since all four types of Input System messages, when placed on drum fields, contain identity bits, they can be read on the CD side by status identification. This mode of reading allows the sorting of messages into tables, one table for each source, each of which can be processed by a single iterative program. A more complete description of each message is given in Part 4.

3.1.2 Output System Information

Information for the Output System is written onto the output buffer (OB) fields of the LOG drum by the Central Computer System by status mode. The Output System reads these fields by the status identification mode. Since the Central Computer System writes only in alternate registers of a field in one IO operation, consecutive acceptances of words by the Output System cannot occur more often than once every 20 microseconds; the division of the three OB fields into two fields (OB even and OB odd) on the CD side is the only requirement to make the OB fields act as a time buffer for the Output System.

The Output System searches each OB field in sequence, looking for acceptable words. (See fig. 1-85.) The switching of field reading is handled by the OD output element. The length of search is controlled by the Output System so that all registers are searched only once. A more complete description of output information transfers is given in Part 6.

3.1.3 Display System Information

Three types of information are supplied to the Display System via the Drum System: digital data, track display data, and radar data. All three types of information are written on the CD side by address. One field of the MIXD drum is used for transferring digital data to the Display System. Track display data requires a full drum of six fields. Radar data also requires a full drum; however, the radar data drum is divided into 9 fields of 24-bit registers rather than 6 fields of 32-bit registers. (See fig. 1-86.) All of these fields are read on the OD

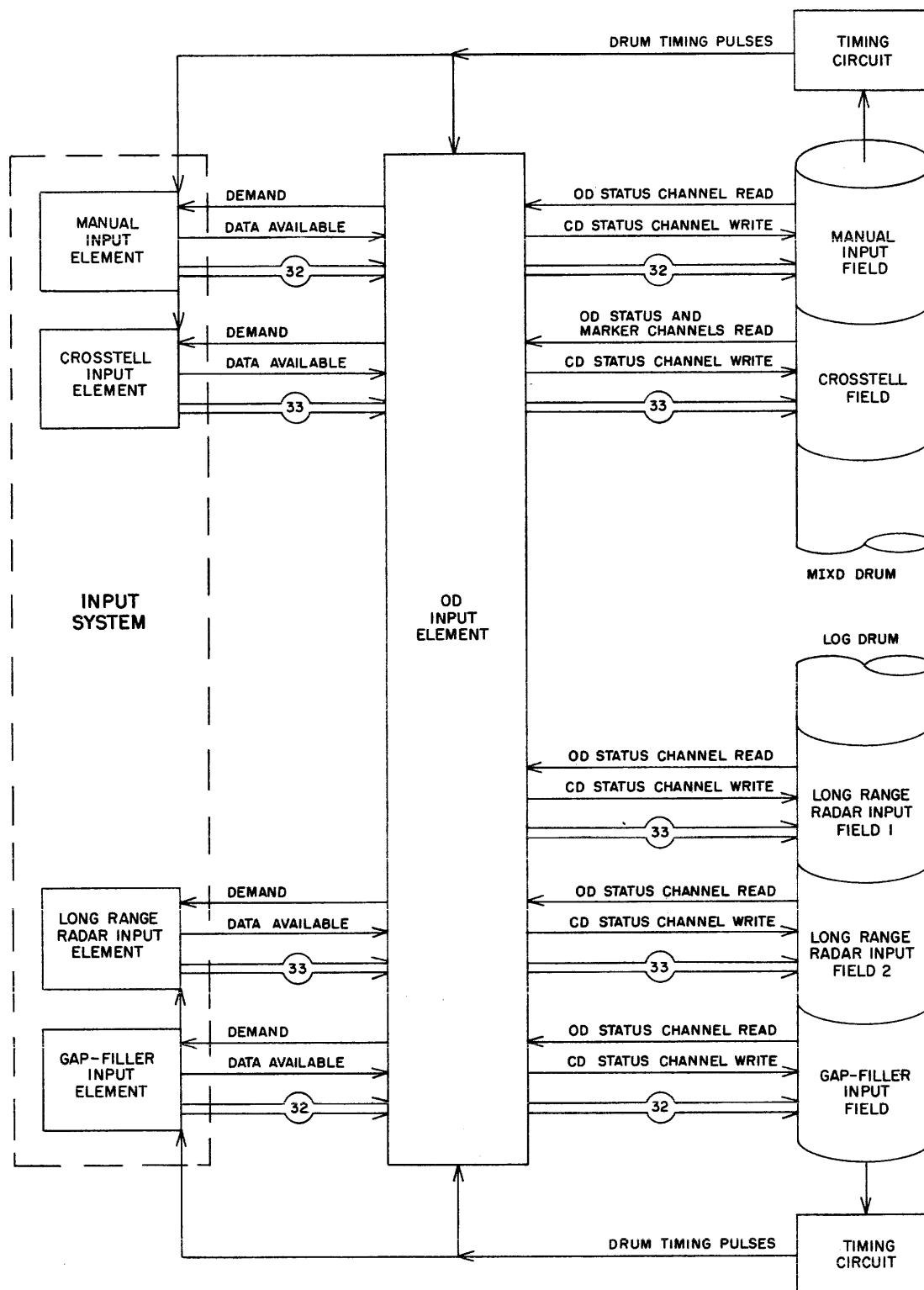


Figure 1-84. Input System Information Flow (OD Side)

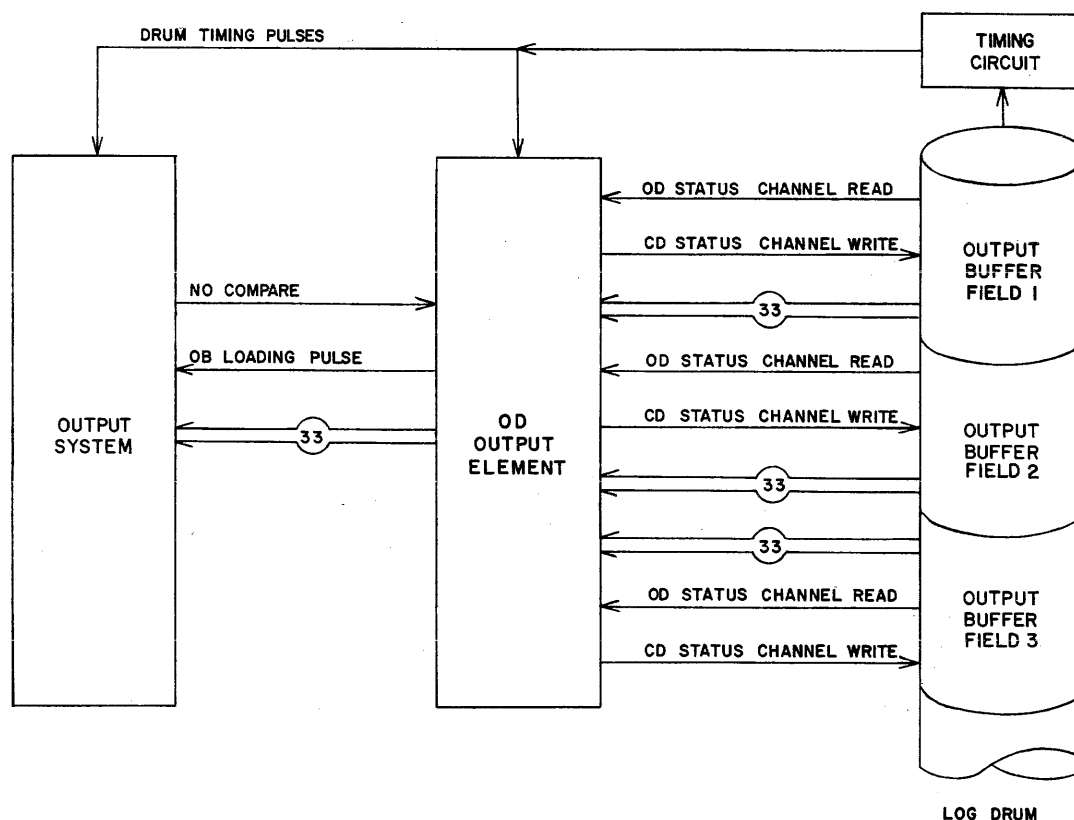


Figure 1-85. Output System Information Flow (OD Side)

side by precession, to allow the use of addressable fields as time buffers between the Central Computer System and the Display System. (Refer to 2.3.2.3.)

Reading of the digital display field on the OD side is initiated by a programmed instruction. The precession pattern used in reading this field involves the reading of every 64th register on the field. The drum must complete 32 revolutions in order to read every register on the digital display field.

Each field of the track display drum is read in sequence by precession in alternation with readings of eight of the fields on the radar data drum. Reading of these two drums is continuous; the only control exercised by the Central Computer System program is in the reservation of one radar data field for writing on the CD side. Switching between fields on each drum and between drums is accomplished by the OD display element. The precession patterns used in reading the track display and radar data drums differ because of the difference in the information forms on these two drums. The precession pattern used in reading a track display field involves the reading of eight consecutive registers (containing one track display message), then skipping eight groups of eight registers before reading another eight consecutive registers. The precession pattern used in reading a radar data field involves the reading of one

register (containing a radar data message), then skipping five registers before reading another register. Display System information transfers are discussed more fully in Part 5.

3.2 CD INFORMATION TRANSFERS

3.2.1 Auxiliary Memory Information

The Central Computer System may store and reobtain information on the auxiliary memory drum fields using address mode or address mode with interleave. Auxiliary memory drums A and B each contain six fields, allowing storage of up to 24,576 words. The two spare fields on the MIXD drum add another 4,096-word storage capacity. (See fig. 1-87.) These fields may be used for storage of processed information outside of core memory. In addition, the track display drum, the radar data drum, and the digital display field contain information which may be reobtained by the Central Computer System for use in further processing of data. The dual function of these fields (both as a time buffer and as auxiliary storage) is made possible by writing and reading them on the CD side by address while they are read on the OD side by precession. Only one drum field may be selected and used by the Central Computer System in an IO operation, since all CD information transfers must make use of the CD element of the Drum System.

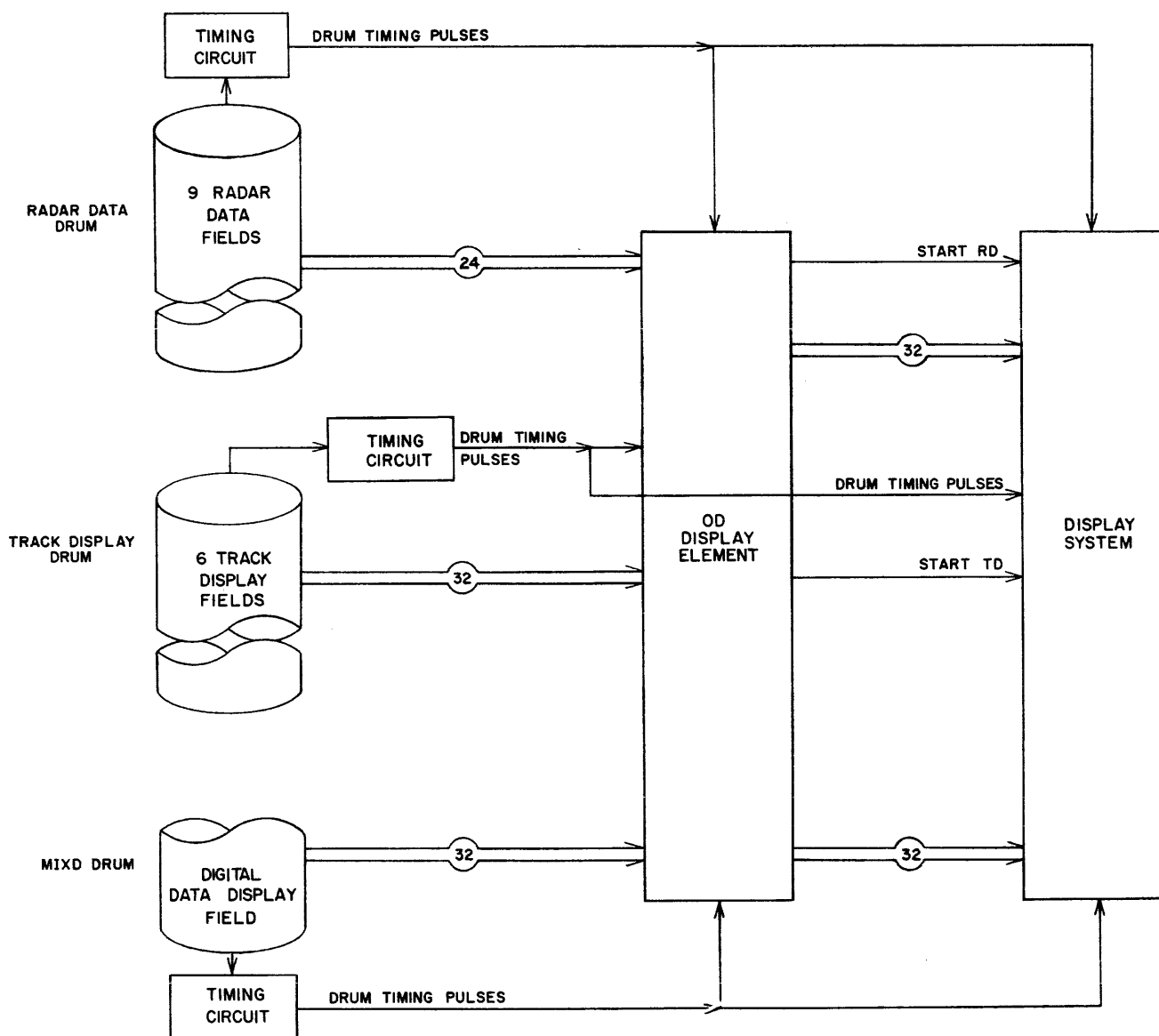


Figure 1-86. Display System Information Flow (OD Side)

The Drum System of AN/FSQ-7 Combat Direction Central includes an auxiliary drum group which allows additional storage space for 73,728 words. (See fig. 1-88.) All fields in the auxiliary drum group are addressable. However, an interlock is provided on the writing circuit in the auxiliary CD element to prevent writing on these fields during normal drum operation. The fields in the auxiliary drum group can thus be read but not written during normal operation and are, therefore, suitable only for storage of nonchanging data such as programs, mathematical tables, and tables of constants. The selection code for each drum field and its possible modes of operation is given in table 1-64.

3.2.2 Input Information

The five drum fields containing information written

on the OD side by the Input System are read by the Central Computer System by status identification. (See fig. 1-89.) A separate IO operation is required to read the information from each source on a given field. Since the number of words on a field with a given identification cannot be known before the reading operation, the IO operation must be terminated by the disconnect counter for that field in the CD element.

It is possible to read an input field on the CD side by the status mode. This mode is seldom used, since it does not sort the information on the field into tables of items from specific sources and therefore places the responsibility of sorting on the Central Computer System. The SDR (μ) instruction which selects the field also indicates the mode of operation (status or status identification). (Refer to table 1-64.)

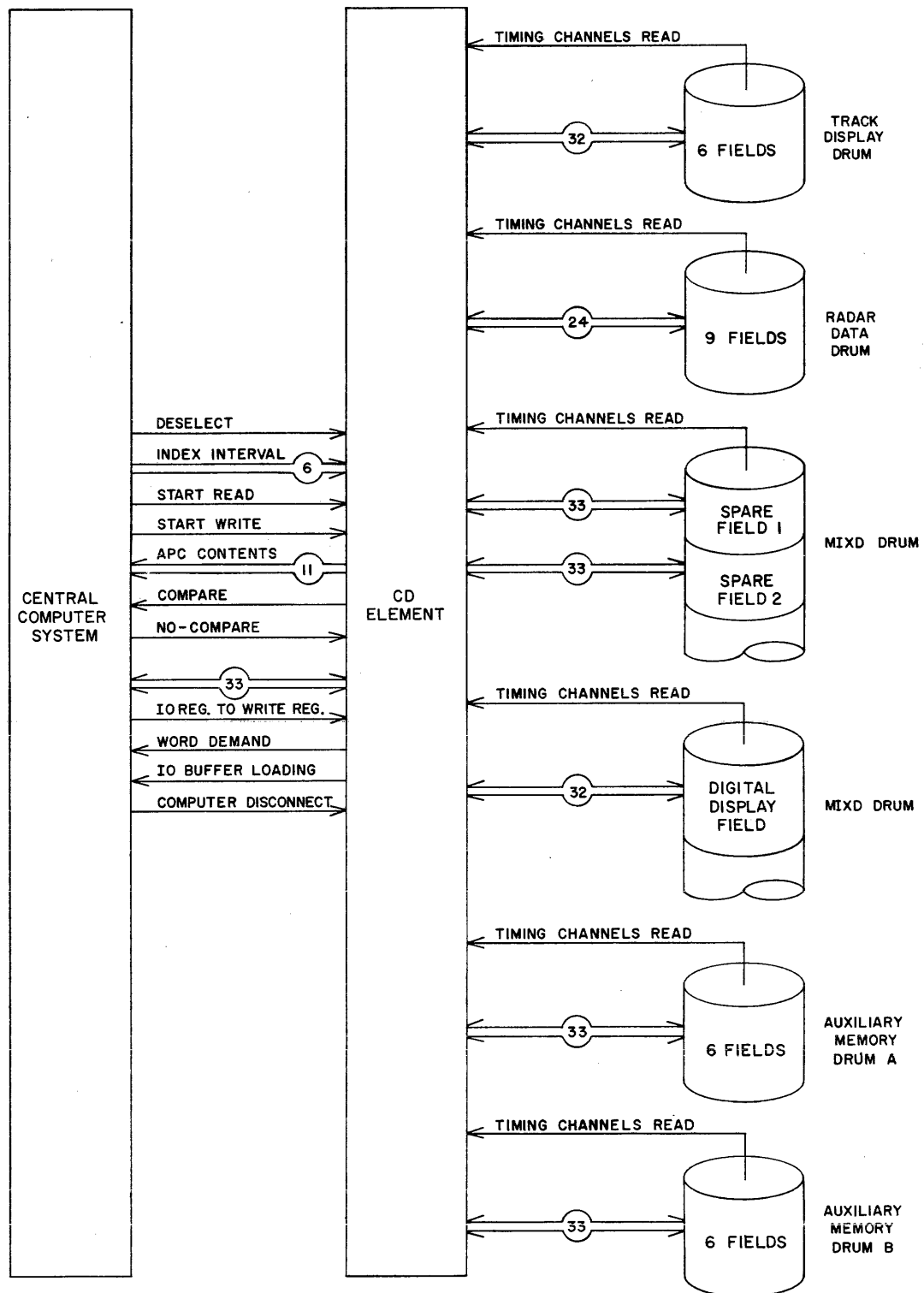
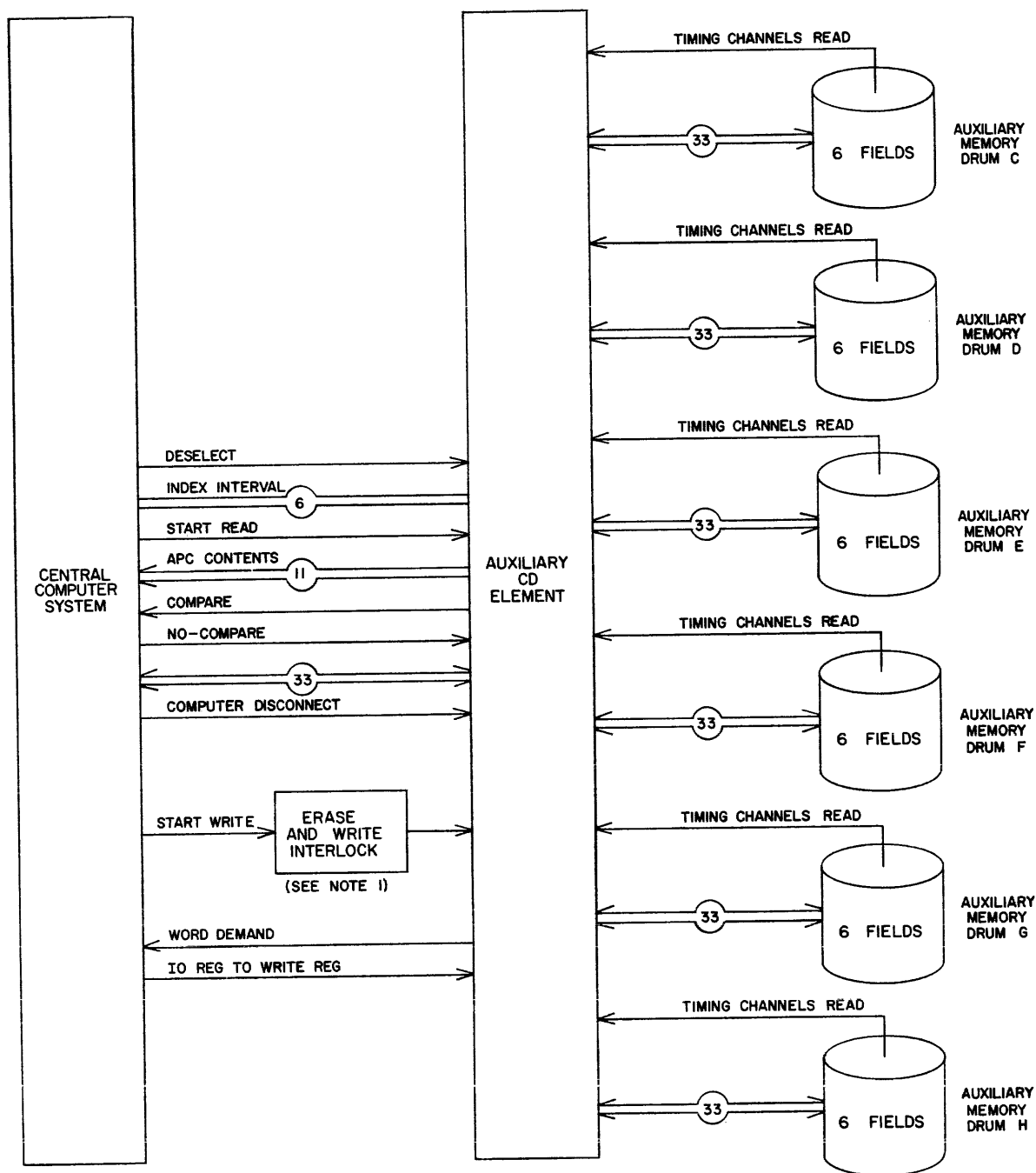


Figure 1-87. Auxiliary Information Flow

It should be noted that no parity bit is included with words from either the manual input (MI) field or from the gap-filler input (GFI) field. A more complete description of the information read from all five input fields is given in Part 4.

3.2.3 Output Information

Output information is prepared by the Central Computer System and delivered to the Output System via the output buffer (OB) fields of the LOG drum. (See fig. 1-90.) These three fields are treated by the



NOTE:

1. INTERLOCK LOCATED ON DRUM SELECTION SECTION OF AUXILIARY DRUM PANEL OF DUPLEX MAINTENANCE CONSOLE

Figure 1-88. Auxiliary Drum Group Information Flow

Central Computer System as two oversized fields of ^{3,054}~~3,072~~ registers each. The Central Computer System can select either OB even or OB odd. Once this selection is made, the CD element allows writing only in the even or odd registers of each field and switches writing from one field to the next as each field is completely searched. Since

writing on the OB fields is done by status mode, the output buffer fields are equipped with a disconnect counter to terminate a CD writing operation after all registers on the OB fields have been searched. Thus, if the Central Computer System program calls for the writing of more words onto the OB fields than can be placed

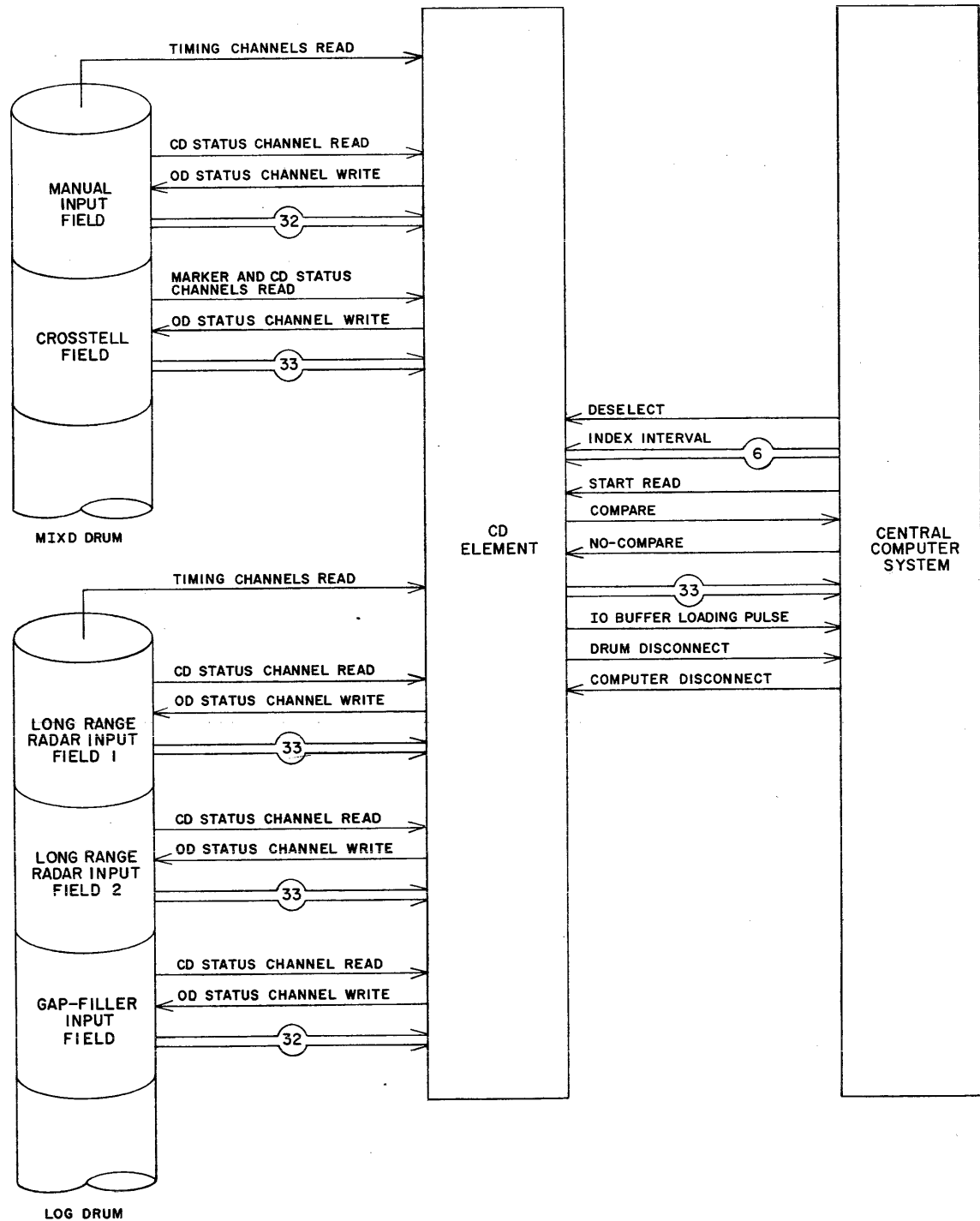


Figure 1-89. Input Information Flow (CD Side)

in empty registers on that field, the IO operation will not be prolonged after the OB fields have been completely searched for empty registers. A new IO operation is required to transfer the unwritten words onto the OB fields. However, the termination of the first IO operation by disconnect counter allows other IO operations to be performed without having to wait for the emptying of

registers on the OB fields. The division of the OB fields into odd and even reduces the rate at which information is delivered to ~~the Output System.~~ ^{EACH STORAGE SYSTEM.} (Refer to 3.1.2.) Table 1-66 summarizes Drum System information flow.

3.2.4 Intercommunication Information

Intercommunication is the exchange of information between the two computers of AN/FSQ-7 Combat Di-

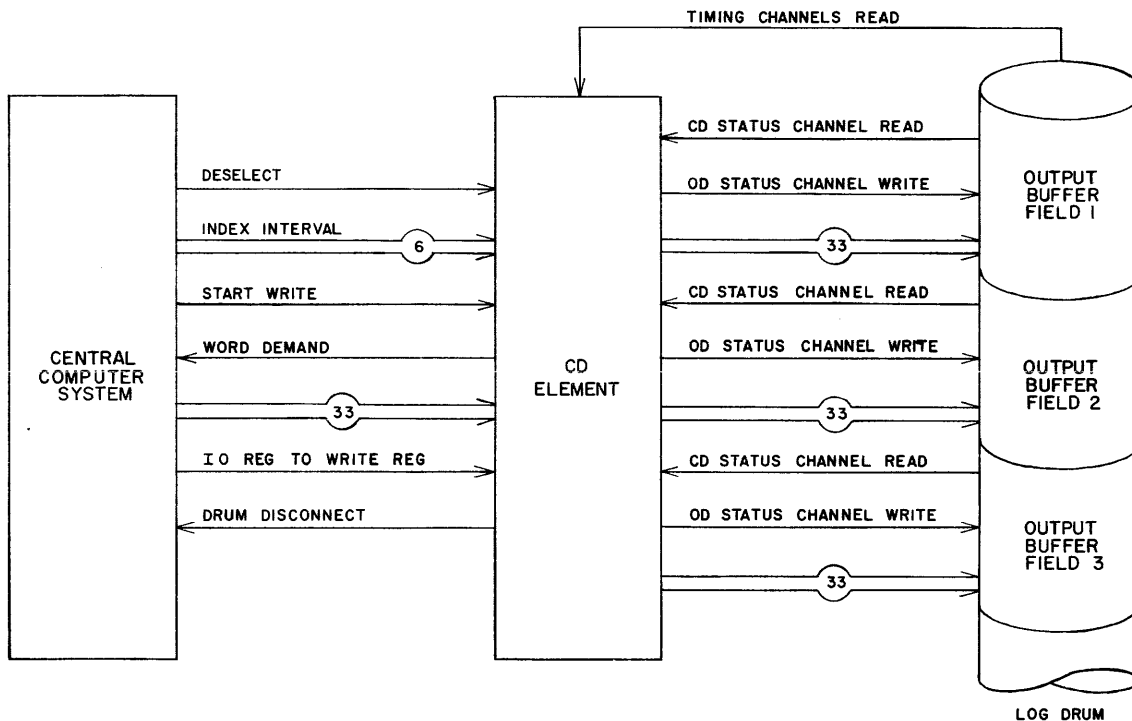


Figure 1-90. Output Information Flow (CD Side)

rection Central. The intercommunication (IC) field of the MIXD drum of each computer is the medium for the exchange of information. (See fig. 1-91.)

Information is written on the CD side of IC field A by address under program control of Central Computer System A. Subsequently, the information is read by the OD IC element A under program control of Central Computer System B. This information is transferred directly to CD element B. Since there is an IC field and an OD IC element in the Drum System of both computers A and B, information transfer can take place in both directions.

Four IC sense units are located in each computer. The sense units in computer A are operated by *PER* instructions (*PER* 10 through 13) from computer B and sensed and turned off by *BSN* instructions (*BSN* 43 through 46) from computer A. Similarly, the four IC sense units in computer B are operated by *PER* instructions from computer A and sensed and turned off by *BSN* instructions from computer B.

When IC information is placed on the IC field by computer A, a *PER* instruction will turn on a sense unit in computer B to indicate the presence of IC information available for transfer. The on condition of the sense unit is detected by a *BSN* instruction and computer B then reads the information on the IC field of computer A. When the transfer of information is completed, a *PER* instruction from computer B will turn on a sense unit in

computer A to indicate that the transfer is complete and that the IC field may be written again.

3.3 TEST OPERATIONS

3.3.1 General

Three classes of tests are used to verify the reliability of Drum System operations. One class is conducted automatically during normal operation and is cursory, since it checks only the number of 1 bits in an information word. This test is called the parity check. The second class of tests is conducted only during maintenance and preoperational periods. These tests, called computer tests, are more exacting, since every bit of a word can be checked.

In computer tests, the Central Computer System writes test words in drum field registers and then reads from the same registers. The identity of the words read is checked by identification bits within each word. Bit-by-bit comparisons are then made in the Central Computer System. These comparisons reveal any discrepancies in the storage or transfer of information in the Drum System.

A third class of tests enables maintenance personnel to exercise manual control over the Drum System. These tests make use of the manual drum tester element. In this way, the Drum System can be isolated from all other systems for maintenance procedures or preoperational checks. Facilities are provided for checking the

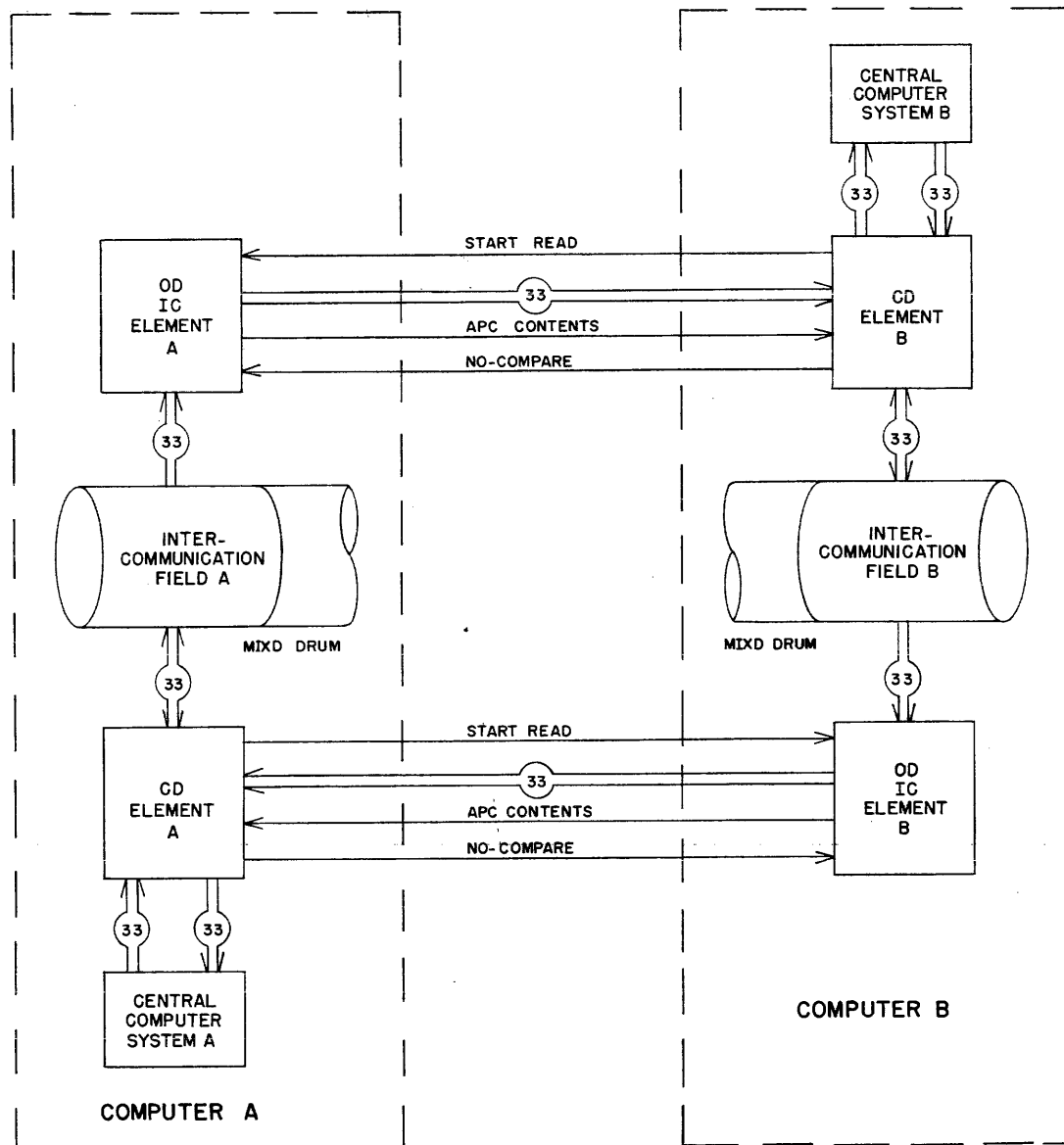


Figure 1-91. Intercommunication Information Flow

storage and control operations of the drums and the CD and OD elements.

3.3.2 Parity Check

All fields of the Drum System (except the radar data fields) have an additional channel called the parity channel. Each word written in these fields may include an extra bit which is written in the parity channel. The extra bit is called a parity bit and can be either a 1 or a 0. The determination of whether a 1 or a 0 is written is made in the originating system by circuits which count the number of 1 bits in an information word. On the CD side, these counting circuits are in the Central Computer System; on the OD side, they are in the Input System. If the total number of 1 bits in a word is an odd number, a 0 parity bit is added; if the total is even, a 1 bit is added.

This is governed by the principle requiring the total number of 1 bits in a word (including the parity bit) to be an odd number.

The parity check can be made only in the Central Computer System on the CD side and in the Output System on the OD side. Circuits in these systems check the contents of a word by counting the number of 1 bits in a word (including the parity bit) before accepting the word. Every word taken from core memory has a parity bit added before it is written in a field. Generally, words written by the Central Computer System in those fields which are used as auxiliary memory have their parity checked only when reread by the Central Computer System. The only exception is the intercommunication field, where parity checks are also performed by the other computer. Words written in the output buffer

TABLE 1-66. DRUM SYSTEM INFORMATION FLOW

DRUM NAME	ABBREV. NAME	NO. OF FIELDS	INFORMATION STORED	SOURCE, CD SIDE	SOURCE, OD SIDE	DESTINATION, CD SIDE	DESTINATION, OD SIDE
Auxiliary memory A	AM-A	6	Subprograms, math values, target data	Central Computer System	None	Central Computer System	None
Auxiliary memory B	AM-B	6	Subprograms, math values, target data	Central Computer System	None	Central Computer System	None
Auxiliary memory C thru H (Auxiliary drum group)	AM-C thru AM-H	36	Subprograms, math values, and constants	Central Computer System	None	Central Computer System	None
Radar data	RD	9	Radar returns, correlated or uncorrelated	Central Computer System	None	Central Computer System	Situation display generator element
Track display	TD	6	Tabular and vector target data	Central Computer System	None	Central Computer System	Situation display generator element
LOG		6					
Long-range radar input	LRI	2	Target position, identity friend-or-foe, altitude, radar source	None	Long-range radar input element	Central Computer System	None
Output buffer	OB	3	Ground-to-ground, ground-to-air, and teletype messages	Central Computer System	None	None	Output System
Gap-filler input	GFI	1	Target position, radar source	None	Gap-filler input element	Central Computer System	None
MIXD		6					
Manual input	MI	1	Meteorological and target data, own weapons status, action orders, target designation	None	Manual input element	Central Computer System	None
Intercommunication	IC	1	Intercommunication messages	Central Computer System	None	Central Computer System	Other computer
Crosstelling input	XTL	1	Miscellaneous information and handovers	None	Other Centrals	Central Computer System	None
Digital display	DD	1	Tactical and strategic ground and air information	Central Computer System	None	Central Computer System	Digital display generator element
Spare 1		1	Same as auxiliary memory	Central Computer System	None	Central Computer System	None
Spare 2		1	Same as auxiliary memory	Central Computer System	None	Central Computer System	None

(OB) fields have their parity checked in the Output System.

On the OD side, a parity bit is contained in input words from the long-range radar input (LRI) and crossteling (XTL) input elements. The parity of these words is checked in the Central Computer System. Although there is a parity channel in both the gap-filler and manual input fields, words written in these fields do not contain a parity bit. Parity checking in this case is done only for testing purposes.

If a drum parity error is detected, a sense unit is turned on and an alarm signal is generated. Three actions may be taken upon generation of an alarm, depending upon the settings of certain switches. The program may come to a halt or it may ignore the alarm and continue. The program may also branch to a predetermined location when the alarm signal is generated.

If the alarm signal causes the program to halt, the current instruction may not be completed. The IO transfer in progress is interrupted, core memory is effectively disconnected, and the IO address counter stops. However, the IO word counter continues to count, and the IO unit tries to complete the transfer. The words being read are discarded and, if being read by status, are permanently lost. If the transfer of information is to a drum, the words written after the alarm are all positive zero.

If the alarm signal causes the program to branch, the program is stopped immediately and the current instruction may not be completed. The contents of the program counter are transferred to the right. A register will be one greater than the address of the instruction being executed when the alarm occurred. The IO unit is deselected, and the IO interlock is cleared. All registers and controls are cleared (with the exception of the right. A register, the IO word counter, and alarm indicators and sense units). The computer then continues with the instruction in register 20,010₈, located in test memory. In the case of a drum parity error detected by the Output System, the action taken is determined by switches associated with test facilities for the Output System.

3.3.3 Computer Tests

Computer tests performed during maintenance and preoperational periods check the Drum System and its interaction with the Central Computer System. Each computer test is individually programmed and is performed in three steps by circuits of the Central Computer and Drum Systems previously described, functioning either with or without the Drum System computer test element. (See fig. 1-92.)

The Drum System computer test element enables the Central Computer System to perform some of the required tests without changing intersystem logic. This is

done by providing the Central Computer System access to the otherwise inaccessible OD side of the Drum System. In the cases of the auxiliary memory drums which have no OD side, the computer test element is not used.

In computer testing, the General Computer System first writes test words into a selected field or drum from the side of the Drum System from which the field or drum is normally written. Next, the Central Computer System reads these test words back from the selected field or drum, again from the side of the Drum System which is normally read. Finally, in the Central Computer System, the test words read are compared with those written. Table 1-67 lists the elements which can be used in conjunction with the computer test element and the type of tests available.

**TABLE 1-67. COMPUTER TEST OPERATIONS
WHICH UTILIZE COMPUTER TEST ELEMENT**

OD ELEMENT	SIDE	OPERATION	MODE	OCTONARY CODE
Manual	OD	Write	Status	22
Long-range radar No. 1				34
Long-range radar No. 2				36
Crosstell				24
Gap-filler				32
Intercommuni- cation (own)		Read	Address	76
Digital display			Identity	17
Situation display				47
Output buffer				30
Output buffer			Status	31

In the OD input element computer test, the Central Computer System utilizes the computer test element to write test words from the OD side of the Drum System. In the first step of the test, the three required IO class instructions are executed, and the writing of test input information proceeds. The OD input element performs the operation using the status or marker status modes. In the second step, the test words are read from the CD side by the CD element. In the third step, comparison is made between the test words written and those read.

In the OD display element computer test, the first step is the writing of test words by the CD element into the display fields. As the second step, the Central Com-

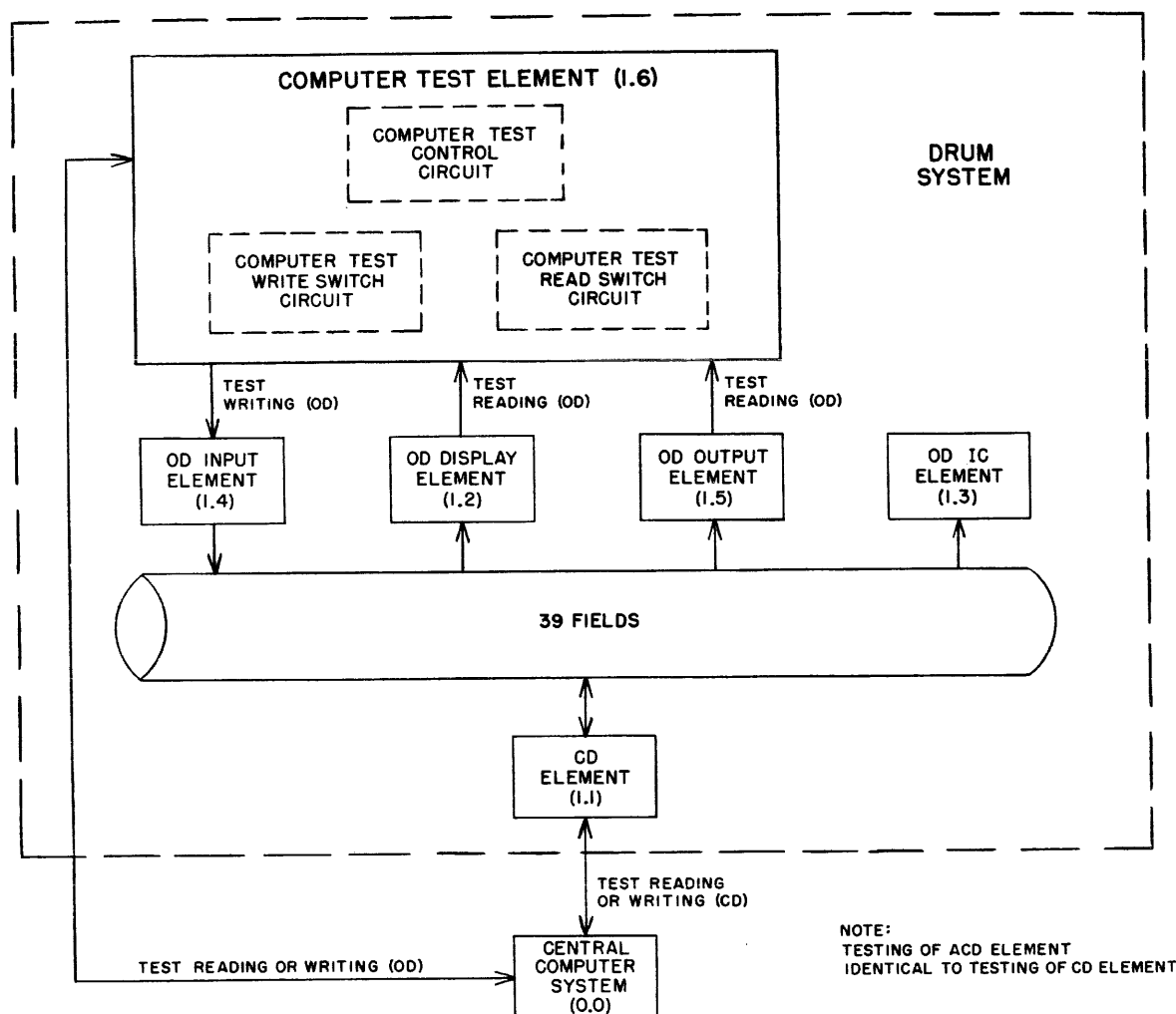


Figure 1-92. Computer Test

puter System utilizes the computer test element to read the test words from the OD side. After the three required IO class instructions are executed for test reading of either situation display or digital display fields, the OD display element performs the operation. In the third step, comparison is made between the test words written and those read.

The second step of the test, in which the display fields are read from the OD side, differs in two important ways from normal display field reading from the OD side. Since situation and digital display test reading are each individually programmed, both cannot be performed concurrently. Test reading is performed using an identity mode that limits the number of words read to the Central Computer System but does allow representative reading.

In the identity mode of reading, which is distinct from the status identification mode, the Central Computer System accepts only those words whose identity bits compare with those loaded into the drum control

register. The no-compare pulses sent from the Central Computer System are disregarded by the Drum System; field switching and processing, however, occur in the normal manner. There are six identity bits (R5 through R10) in every situation display test word. There are two (R14 and R15) in every digital display test word.

The first step of the OD output element computer test is similar to the OD display element test. Test words are written from the CD side by the CD element. In this case, the words are written into the output buffer (OB) fields. In the second step, the Central Computer System utilizes the computer test element to read the test words from the OD side. After the three required IO class instructions are executed for test reading either output buffer, even or odd, the OD output element performs the operation. In the third step, Comparisons are made in the Central Computer System.

3.3.4 Manual Drum Tests

The manual drum tester element enables maintenance personnel to exercise manual control over Drum

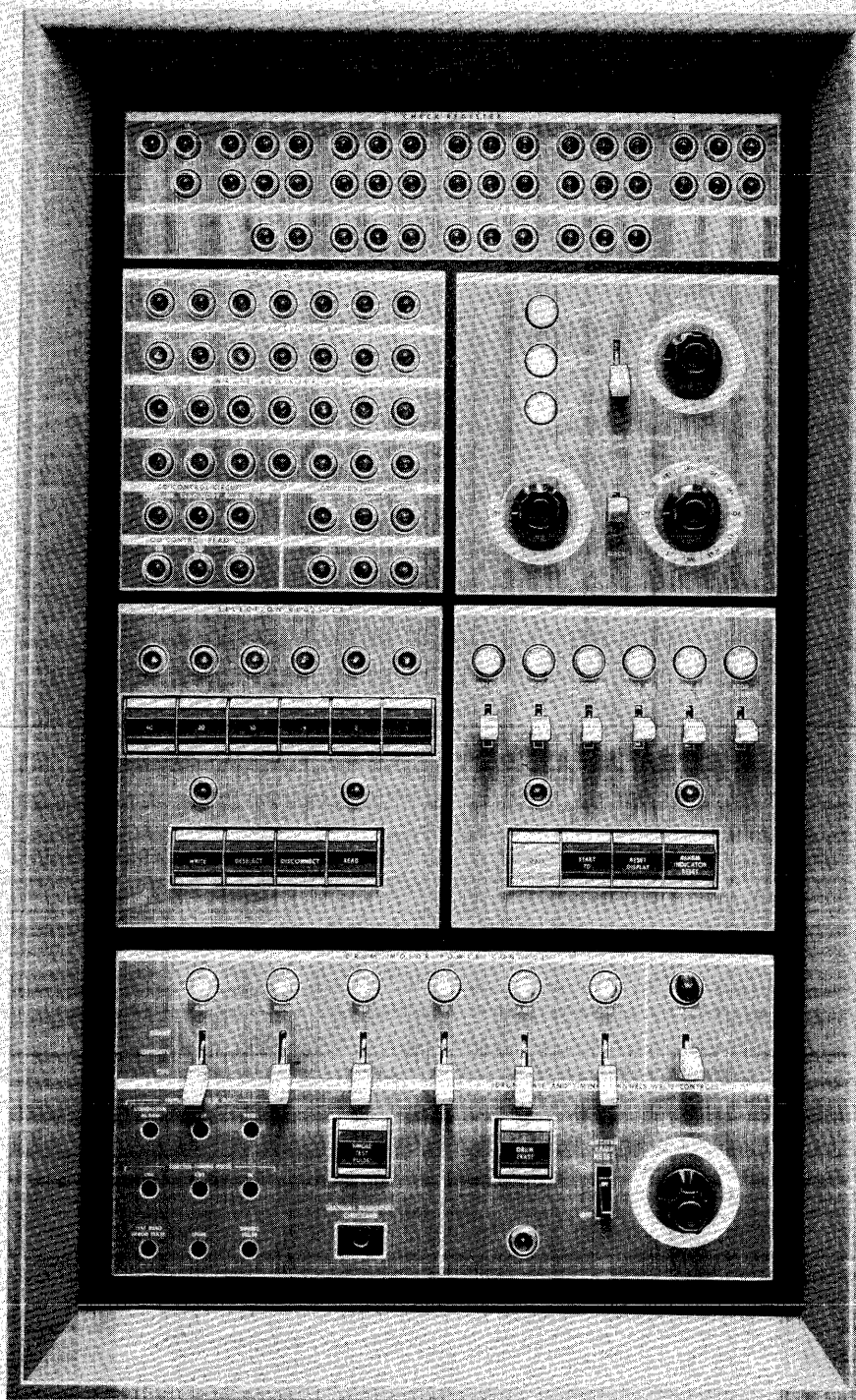


Figure 1-93. Manual Drum Tester Panel

System operations for performance checking. The manual drum tester element has facilities for checking the storage operations of the magnetic drums and the control operations of the CD and OD elements. In addition, the manual drum tester element contains manually operated controls that energize and de-energize the Drum System and lamps that indicate the condition of Drum System circuits.

The manual drum tester element contains five sections: the drum motor control unit, the drum timing and index channel write circuits, the drum erase circuits, the drum manual test circuits, and the indicating circuits. (See fig. 1-93.) The functions of the five sections follow.

The drum motor control unit in the manual drum tester element section controls the motors that drive the 12 drums of the Drum System. Manual starting and stopping of one, all, or any number of drums can be accomplished by using the circuits and controls of this section.

The drum timing and index channel write circuits write the timing and index channels on each of the 12 drums in the Drum System. It is necessary to write both channels whenever a drum is placed in operation after maintenance or during installation.

The drum erase circuits of the manual drum tester element erase all magnetic flux on the drums. Erasure is necessary whenever an error caused by unwanted flux on the drum surface is detected or at the conclusion of maintenance procedures.

The drum manual test circuits check all circuits in the Drum System as well as the operation of the drums themselves. These tests are performed without the use of any other system in AN/FSQ-7 Combat Direction

INFORMATION NOT AVAILABLE

Figure 1-94. Auxiliary Drum Tester Panel

Central. Manual tests are thus available during installation or maintenance procedures.

The indicating circuits in the manual drum tester element provide visual indications of the normal operating status of a number of Drum System circuit components. In addition, indicating circuits enable maintenance personnel to localize troubles detected during test procedures to the circuit or pluggable unit involved. Similar test facilities are available for the auxiliary drum group. (See fig. 1-94.)

CHAPTER 5

INPUT SYSTEM

SECTION 1

LONG-RANGE RADAR INPUT ELEMENT

1.1 INTRODUCTION

The Central Computer System of AN/FSQ-7 Combat Direction Central is capable of processing reports on aircraft flying within a given area in order to defend that area against air attack. Data on potential targets must be acquired, generally by radar, and introduced into the Central Computer System. This function is performed by the Input System.

The Input System is divided into four elements, each handling a different type of information. The long-range radar input (LRI) element receives data from radar sets with ranges of up to 200 miles as well as from height-finder radars located near the long-range radars. (See fig. 1-95.) The long-range radars are arranged to provide coverage of every point within the surveillance area by three different radars. Those portions of the surveil-

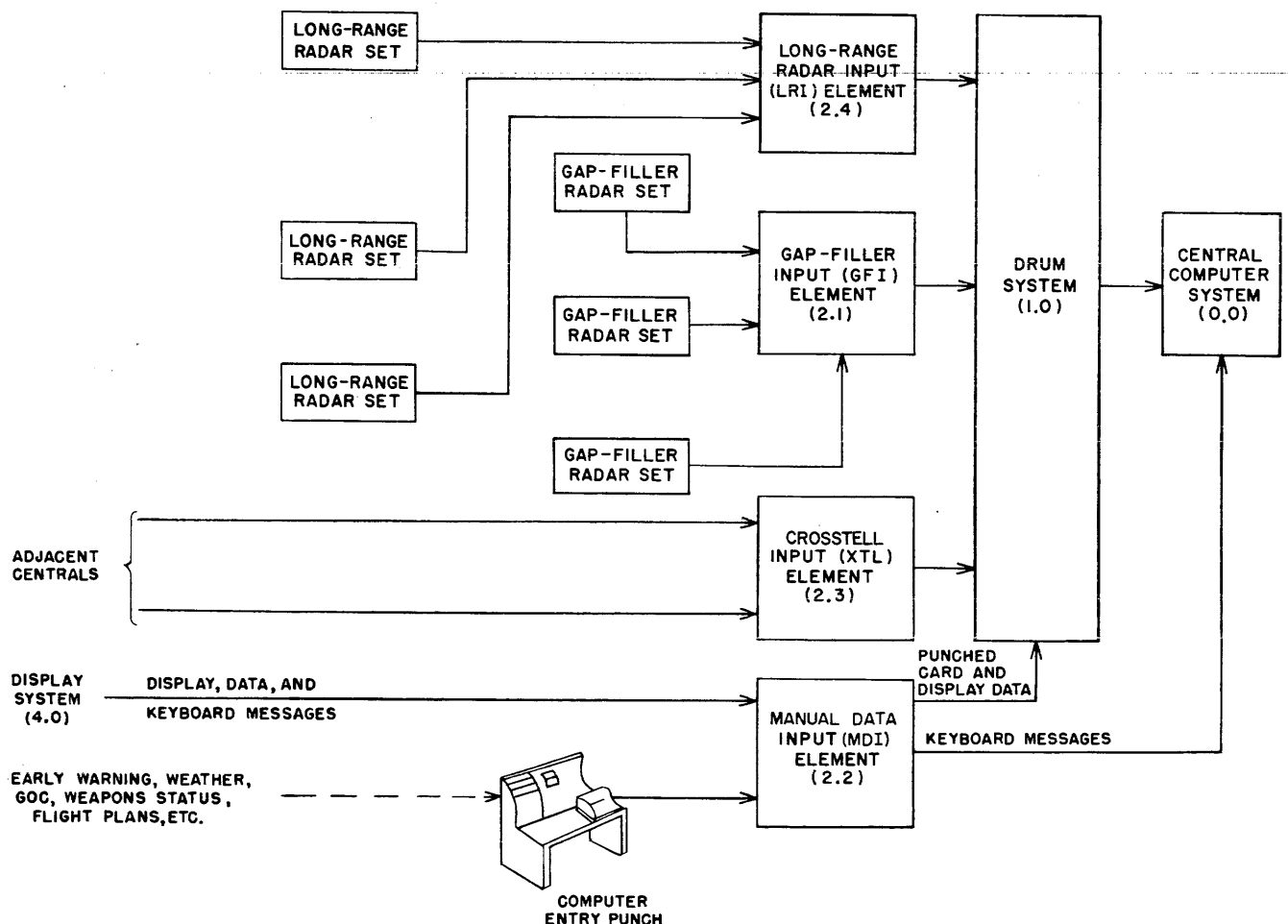


Figure 1-95. Input System Information Flow

lance area which are not covered by long-range radars are covered by radars with ranges of about 50 miles. These gap-filler radars are so called because they fill gaps in the long-range radar coverage. Since data from gap-filler radars is different in form from that supplied by long-range radars, this data is received and processed by the gap-filler input (GFI) element.

The crosstell (XTL) input element receives data already processed by adjacent centrals. Crosstell messages generally describe aircraft moving out of the surveillance area of the transmitting central into the area covered by the receiving central.

The manual data input (MDI) element handles miscellaneous data required by the Central Computer system in processing an air defense problem. Data such as early-warning radar or Ground Observer Corps reports, weather and weapons status reports, and flight plans of friendly aircraft are presented to the MDI element in punched card form via computer entry punches (key-punch machines modified to read as well as to punch cards) for temporary storage in the Drum System. Display data which must be referred back to the Central Computer System for further action is processed by the MDI element as is punched card data. In addition, directions are conveyed from air defense personnel to the Central Computer System in the form of keyboard messages made up by these personnel.

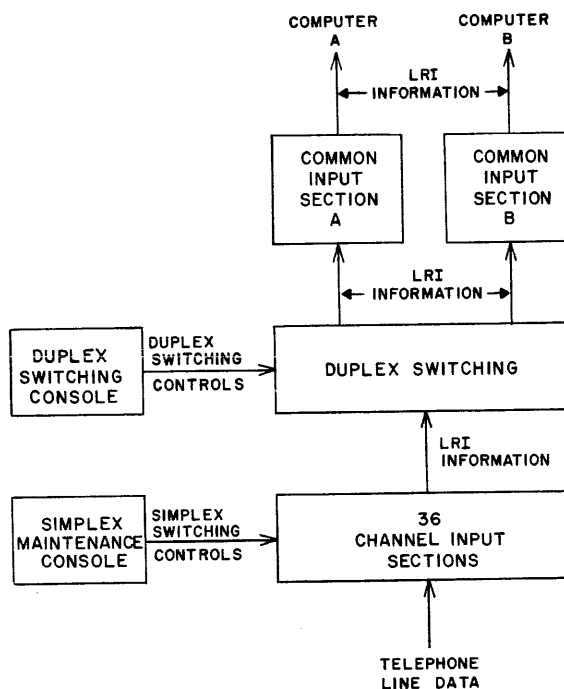


Figure 1-97. LRI Duplexing

Each element places its data on a separate drum field by status mode. (The keyboard messages handled by the MDI element are the only exception; keyboard messages are read directly by the Central Computer System.) Each input message on a given field contains identity bits which allow the Central Computer System to read and sort input messages into tables by type and by identity within each type.

1.2 LRI ELEMENT DESCRIPTION

1.2.1 Element Function

The LRI element receives data transmitted from a long-range radar site via telephone lines, processes this data, and stores it on the LRI fields of the LOG drum for subsequent program-controlled transfer to the Central Computer System. (See fig. 1-96.) Data is transmitted over telephone lines in serial form. The LRI element must, therefore, assemble this serial data for transfer in parallel onto the LRI drum fields. In the process, the LRI element checks the accuracy of information transmission and adds to each message the time of transfer to the drum fields and identification of its source.

1.2.2 Element Logic

The LRI element is divided into two sections: channel input (36 simplex units) and common input (duplex). Two channel input sections process the data from one radar site. The channel input sections feed through the common input section onto one of the two LRI drum fields. When a channel input section receives a mes-

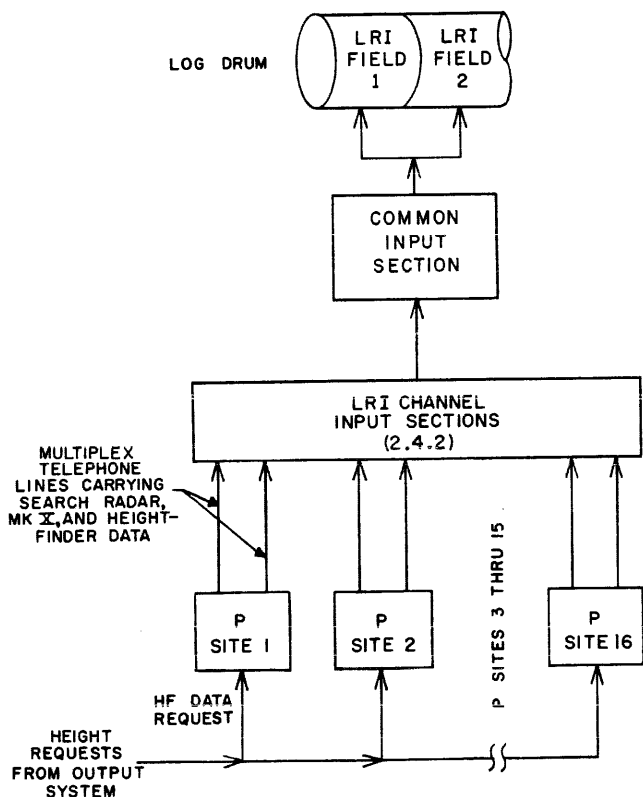


Figure 1-96. LRI Element Information Flow

sage, it accepts the message in serial form, checks for errors in transmission, and then holds the message until the LRI field onto which that channel can write is ready to receive the message. When requested by the Drum System, the message is sent through the common input section in parallel form. As it is written onto the LRI field which accepts it, the common input section adds several bits indicating the time at which the message is written and several other bits identifying its source. In addition, the common input section generates parity bits for each word placed on the LRI fields.

The channel input sections of the LRI element are simplex since the failure of one section would not seriously impair the operation of the central. The common input section, whose failure would disable the entire LRI element, is duplex; a common input section for the LRI element is included in computer A and in computer B. (See fig. 1-97.) Those channel input sections receiving data from radar set supply their outputs through the active common input section, whether A or B, as determined by the duplex switching console

which designates one computer as active. The status of the channel input sections is controlled at the simplex maintenance console. (Refer to Ch. 10.)

1.2.3 LRI Data

1.2.3.1 Data Sources

The equipment at a long-range radar site, known as a P site, comprises the data source for the LRI element. When fully equipped, a P site contains a long-range search radar set with IFF and two semiautomatic height-finder radar sets. (See fig. 1-98.)

The search radar set sweeps its coverage area with a beam narrow in azimuth and broad in elevation. The broadness in elevation permits coverage of a large volume of space in one sweep; the narrowness in azimuth allows precision of azimuth determination which is limited only by the width in azimuth of the radar beam. In most long-range radar sets, this width is approximately 1-1/2 degrees. Precision of range measurement is generally measured to an accuracy of 1/8 mile. A disparity thus exists between the precision of azimuth and of

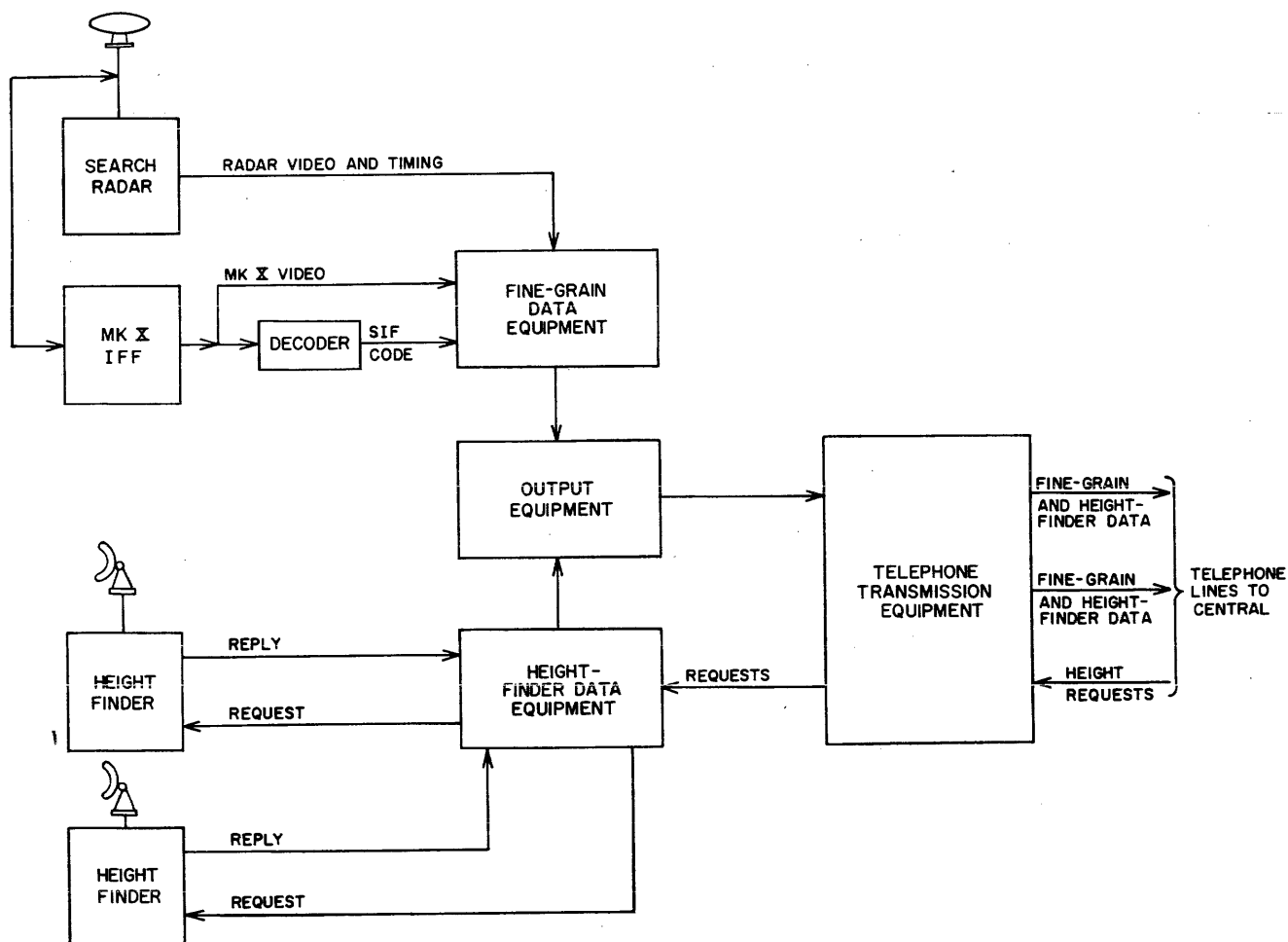


Figure 1-98. P Site Information Flow

range determination, since $1\frac{1}{2}$ degrees at 200 miles represents a distance of about 4 miles. The fine-grain data equipment which the search radar set feeds improves the azimuth accuracy by a process known as beam splitting.

The IFF set associated with the search radar set utilizes the antenna of the search radar in transmitting interrogation pulses and in receiving coded replies from aircraft carrying IFF transponders. These IFF returns are thus co-ordinated with the search radar returns from specific aircraft. The IFF returns can be used to generate target position information in exactly the same manner as that used in extracting this information from search radar returns. The IFF returns are superior to search radar returns in their freedom from spurious target returns due to reflections from ground obstructions or from clouds, from noise which is not likely to generate the patterned response required for an IFF response, and from poor propagation conditions, since IFF receives a transmitted response rather than the reflections of its own transmission. In addition, IFF may be used to identify a specific aircraft from the pattern of responses made by its transponder. If each transponder in a given area is assigned a unique response code, this code can be detected and indicated to the central as a selective identification feature (SIF) code.

The height-finder radar sets utilize radar beams broad in azimuth but narrow in elevation (effectively rotated 90 degrees from a search radar beam). Upon request from the central to a specific height-finder, the antenna of that height-finder is pointed in the azimuth direction of a selected target and moved in a nodding motion, scanning vertically for the selected target. The height above ground of the target can be generated by determining the range and elevation angle of the target. In addition, the height-finder operator can observe, more accurately than can a search radar operator, the relative number of aircraft making up the target and their grouping (whether dispersed predominantly in range, azimuth, or elevation). This data can then be transmitted to the central for use in identifying and locating targets more accurately than is possible with search radar data only.

1.2.3.2 Data Handling

Both search radar and IFF data are handled by the fine-grain data equipment at a P site. The search radar video information is first passed through a moving target indicator (MTI) filter which suppresses reports of nonmoving objects, thus preventing reflections of ground obstructions from being reported as targets. The search video is then presented on a radar mapper (a manually controlled filter) to remove reports of clouds or clutter. (See fig. 1-99, part A.) The radar-timing-and-azimuth-sector pulses applied to the mapper are used to synchronize the plan position indicator (PPI)

display of the mapper. A photomultiplier tube mounted over the display tube detects all targets which are not opaque by the mapper operator and reports these targets to the beam splitter.

The beam splitter generates more accurate azimuth measurements on each target by finding the middle of each target reported by the search radar set. As the search radar beam moves across an aircraft, more than one search pulse senses the aircraft. Therefore, more than one return pulse is received by the search radar set. By noting the antenna azimuth headings for the first and last returns from a given target, the center azimuth heading can be found to an accuracy determined by the ratio of degrees of antenna motion in azimuth per unit time to the number of search pulses within that same unit time. The actual number of return pulses for a given target is also supplied to the central as an indication of the size of the target (run length). In addition, since the beam-splitting process introduces some variation in the time between detecting the target and reporting it to the central (a delay that varies with the number of return pulses), this time delay is also reported to the central. Range information is sent to the central with the precision obtained by the search radar set.

IFF reports are supplied directly to the beam splitter; no mapping is required because IFF returns are inherently free of noise and clutter removable by mapping. If both search radar and IFF reports on a particular aircraft are received by the beam splitter, only the IFF reports are used. In this case, run length information is not generated since, in general, the number of IFF responses is independent of the number of aircraft at a given location. If SIF codes are being used, the IFF response pattern is presented to the central along with the normal position and time delay information.

Each height-finder radar operates in response to specific height requests addressed to it. A height request specifies the location of the aircraft with respect to the height-finder, its former height (if available), the address of the height-finder (both site and which height-finder at the site), a request number (to allow identification of the reply), and any special requests for the height-finder operator to determine the predominant dispersion of the aircraft observed. The request is passed through an address sensing unit to the portion of the height-finder data equipment handling the addressed height-finder. (See fig. 1-99, part B.) The target coordinates (in rectangular form) are presented to a converter which generates azimuth information (to position the height-finder antenna) and range information (to position a cursor on the height-range indicator of the set). The old height information is applied to the height coder for initial positioning of the height cursor on the height-range indicator. The request number is

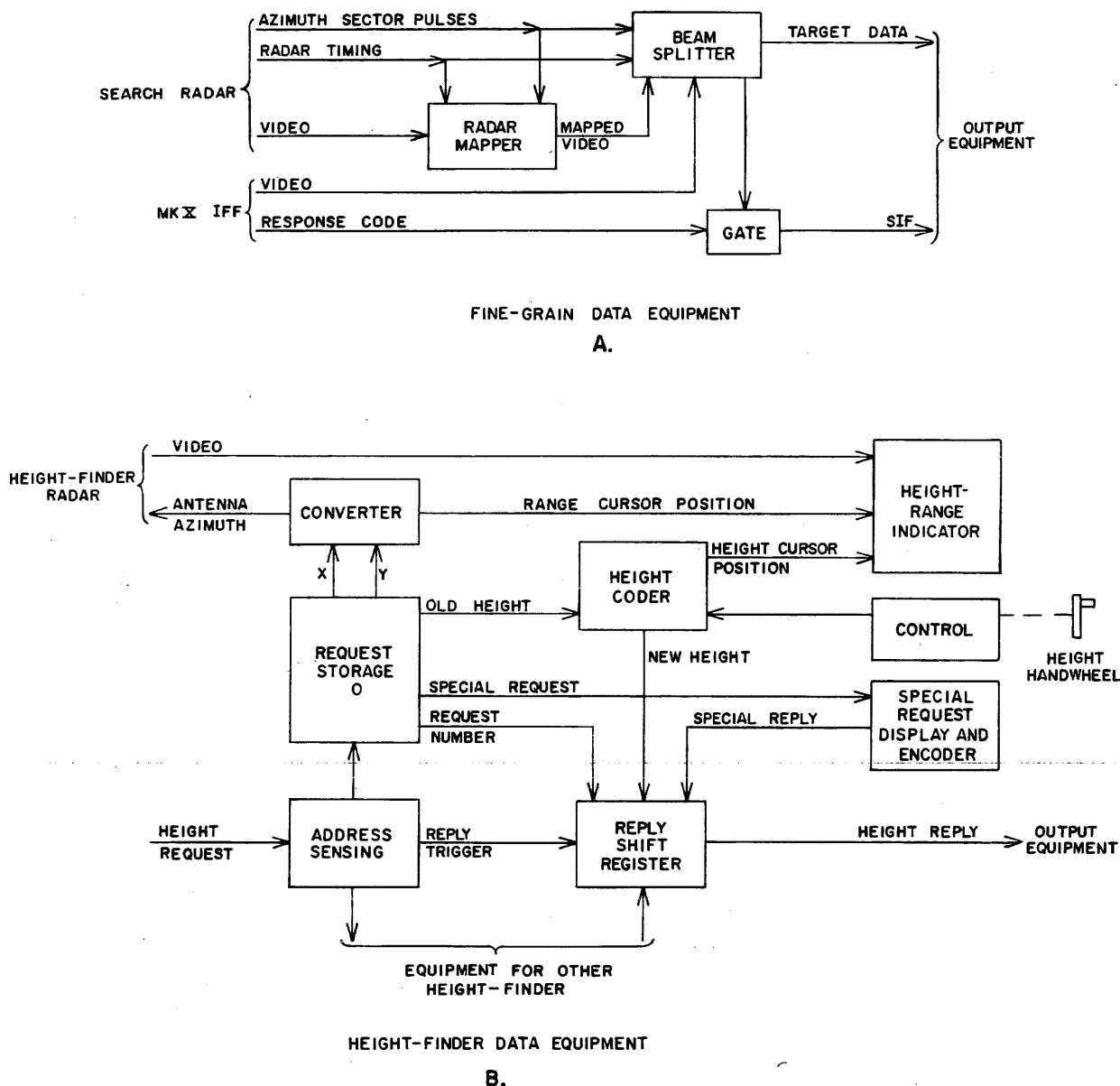


Figure 1-99. P Site Data-Handling Equipment

transferred to the reply shift register to be transmitted to the central within the height reply. The special request, if any, is indicated to the operator by one of a set of indicator lights whose meanings are preassigned.

The height-finder operator must find the indicated target on the height-range indicator and move the height cursor onto the target. (See fig. 1-100.) As the height-finder antenna moves vertically, the sweep displayed on the indicator tilts to match the elevation angle of the antenna. The new height is coded and supplied to the reply shift register. If a special request has been made, the operator replies by depressing the push-button of a set with preassigned meanings which represents the indicated reply. The operator is allowed 15 seconds to position the height cursor and up to 30 sec-

onds if a special reply is required. The height reply is not sent back to the central until it is triggered out by the next height request to that height-finder.

1.2.3.3 Data Forms

Information from a P site is transmitted to the central in the forms shown in figure 1-101. Each message is transmitted as a serial train of 52 pulses at a rate of 1,300 pps. Each message is therefore transmitted in 40 milliseconds. The sync pulse preceding the data in each message is used for control purposes. The busy bit transmitted at the sixth timing pulse contains a 1 if information is contained in the first message word; similarly, the busy bit transmitted at timing pulse 30 contains a 1 if information is contained in the second message word.

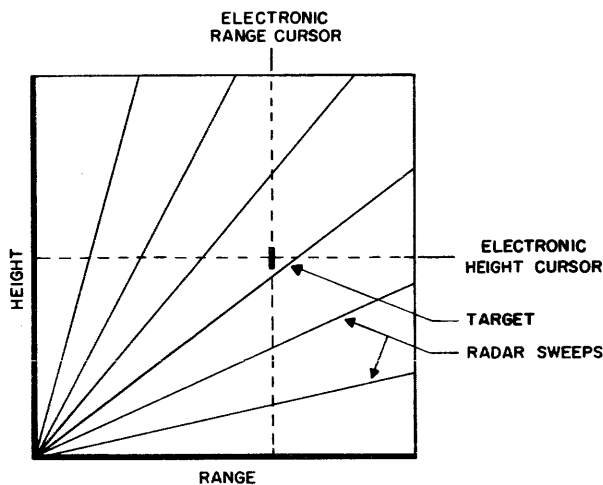


Figure 1-100. Height-Range Indicator, Semiautomatic Height-Finder

The parity bits at timing pulse 29 and timing pulse 52 indicate the parity of the message words preceding them. The parity bit at timing pulse 29 indicates the parity of the first message word; the parity bit at timing pulse 52 indicates the parity of the second message word. An even parity count is used to prevent generation of a parity alarm upon receipt of an empty message word (containing all 0's).

The message words within a search radar message contain the following information:

- a. Message word 1:
 1. Range, 10 bits, with a precision of $\frac{1}{4}$ mile
 2. Azimuth, 12 bits, with a precision of 0.088 degree ($\frac{1}{4,096}$ circle)
- b. Message word 2:
 1. Message label, 4 bits, identifying message type as search radar, IFF, or height-finder
 2. Time delay, 5 bits, with a precision of $\frac{1}{4}$ second
 3. Run length, 3 bits, indicating approximate size of target (number of returns).

If the message is IFF originated, its contents are identical with a search radar message except for the following:

- a. The message label identifies the message as IFF originated.
- b. No run length is indicated.
- c. The SIF, 12 bits, may be sent to identify a specific aircraft from its IFF response pattern.

A height reply contains the following information:

- a. Message word 1:
 1. Number of aircraft, 3 bits (approximate)

2. Separation, 2 bits, indicating degree of dispersion of a multiple target
 3. Formation, 2 bits, indicating type of formation, if any
 4. Special reply, 3 bits, if any
 5. Request number, 3 bits
 6. Address, 1 bit, indicating which of the two height-finders is replying
 7. New height, 8 bits
- b. Message word 2: Message label, 4 bits, indicating type of message.

All three types of LRI messages are transmitted to the central from a fully equipped P site intermixed on two telephone transmission channels. Thus, two LRI channel input sections are required to process the incoming data from a single P site.

1.3 ELEMENT OPERATION

1.3.1 General

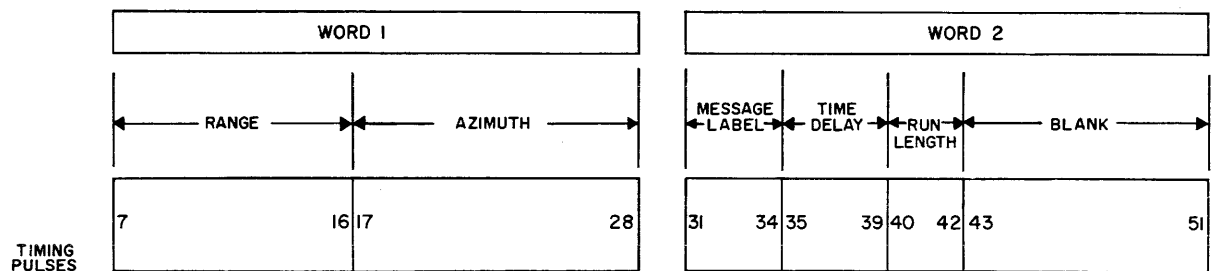
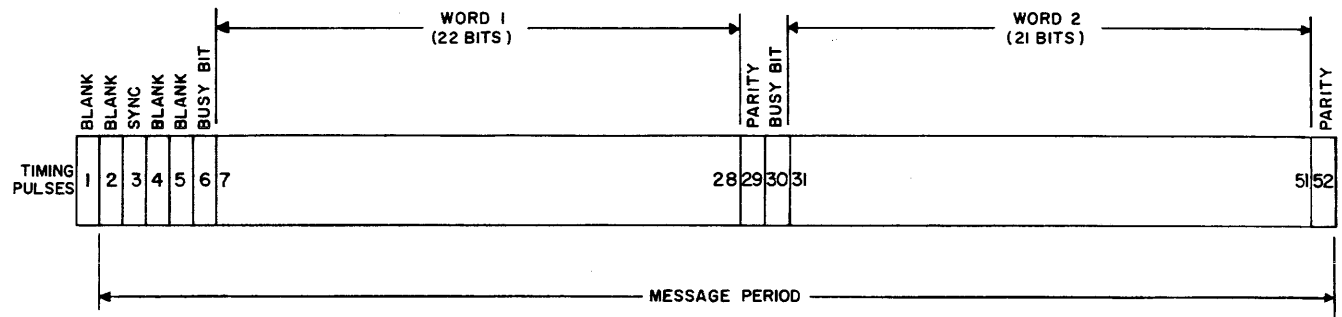
Each LRI channel input section receives and processes data from the telephone recovery equipment associated with one telephone channel. Each message supplied to an LRI channel input section in serial form is read in, checked for errors in transmission, and then stored until it can be transferred through the common input section to an LRI drum field. The operation of the LRI element can therefore be considered in three phases: read-in, checks, and transfer to drum.

1.3.2 Read-In

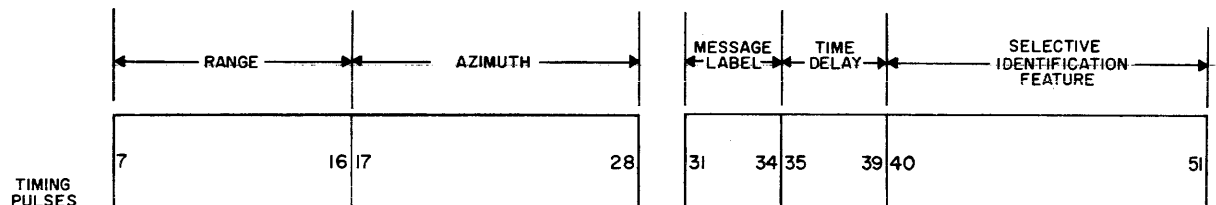
The pulses constituting a message period come from the telephone recovery equipment of a particular telephone channel at a rate of 1,300 cps. To make them compatible with the timing of the LRI element, the pulses are lengthened and synchronized with drum timing pulses by the data converter and synchronizer of the channel input section. (See fig. 1-102.) The messages supplied to the LRI channel input sections are, at present, of fixed 52-bit length. However, provisions are incorporated within the LRI element to handle a variable message length. The extent to which this message length may be varied, from sync pulse to sync pulse, is not less than 32 bits but not more than 64 bits.

Read-in begins when the sync pulse of a message period passes through the data converter and synchronizer. The sync pulse clears the core shift register, senses the parity busy bit for a parity error or a no busy bit condition detected in the preceding message, and resets the parity busy bit.

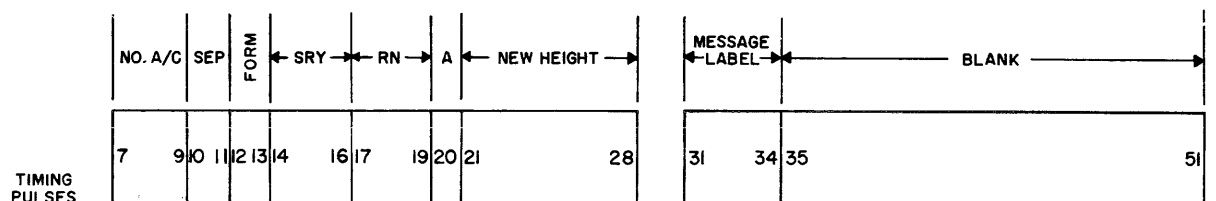
The timing pulses from the telephone recovery equipment are synchronized with drum timing and applied to the core shift register to initiate shift pulses for the core shift register operation. A shift pulse is applied to the core shift register directly after each data



SEARCH RADAR



MK X IFF



HEIGHT REPLY

NOTE:

NO. A/C NUMBER OF AIRCRAFT
SEP SEPARATION OF AIRCRAFT
FORM FORMATION OF AIRCRAFT
SRY SPECIAL REPLIES
RN REQUEST NUMBER
A ADDRESS

Figure 1-101. LRI Telephone Line Message Layouts

pulse is written into the first core, with a consequent effect similar to the shifting of a typewriter carriage one space after each letter is typed. The writing and shifting of the core shift register continues at a 1,300-pps rate until the busy bit short word or busy bit long word

is shifted out of the register, indicating that the last bit of the message period has been written into the register.

With the entire message in the core shift register, shifting does not have to be restricted to the input pulse rate but can be increased to 50,000 pps, the optimum

shifting rate of the core shift register. The busy bit short word or the busy bit long word from the core shift register is applied to the parity busy bit. If the parity count of the incoming message is even, the sync pulse of the following message causes a good-message pulse to be transmitted to the read-out core buffers, which in turn applies a start-fast pulse to the fast shift beginning the transfer of information at a fast rate to the word 1 and word 2 core buffers. The entire message is transferred from the core shift register to the word 1 and word 2 core buffers in 1.04 milliseconds at a 50,000-pps rate, which is an interval shorter than the time between arrival of the sync pulse of the following message and the arrival of the first data bit of that message. The fast shift is ended by the busy-bit pulse from the word 1 and word 2 core buffers to the fast shift. The core shift register is then cleared for the subsequent incoming message, and the message in the word 1 and word 2 core buffers is permitted to shift once more. The LR 8-delayed-

end-fast-shift pulse is applied to the read-out core buffer to enable the read-out alarm should the fast shift of the following message from the core shift register occur with a message in the word 1 and word 2 core buffers. When the drum demand pulse arrives at the read-out core buffer, word 1 stored in the word 1 and word 2 buffers is transferred to the common input section 2.5 microseconds later by the word 1 read-out-and-clear-drive pulse, and word 2 is transferred 10 microseconds after that by the word 2 read-out-and-clear pulse. The word 1 and word 2 core buffers are cleared preparatory to receiving a new message as each word is transferred to the common input section. When word 1 is transferred from the word 1 and word 2 core buffers, a write signal is also generated and sent to the common input section.

When the word 2 read-out-and-clear signal and the busy-bit-long-word signal occur in synchronism, a long-word signal is transmitted to the common input section, informing it that a long word is en route.

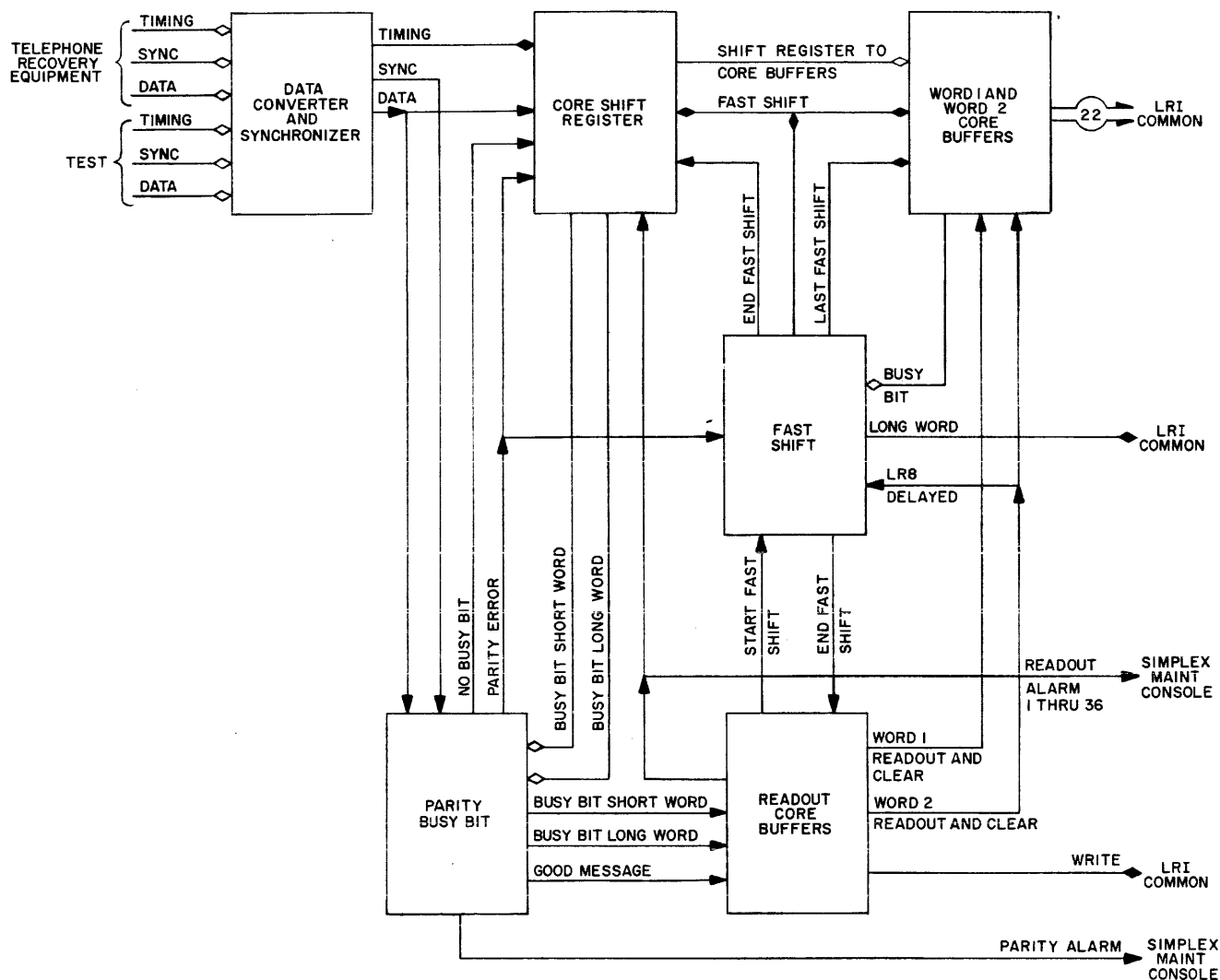


Figure 1-102. LRI Channel Input Section

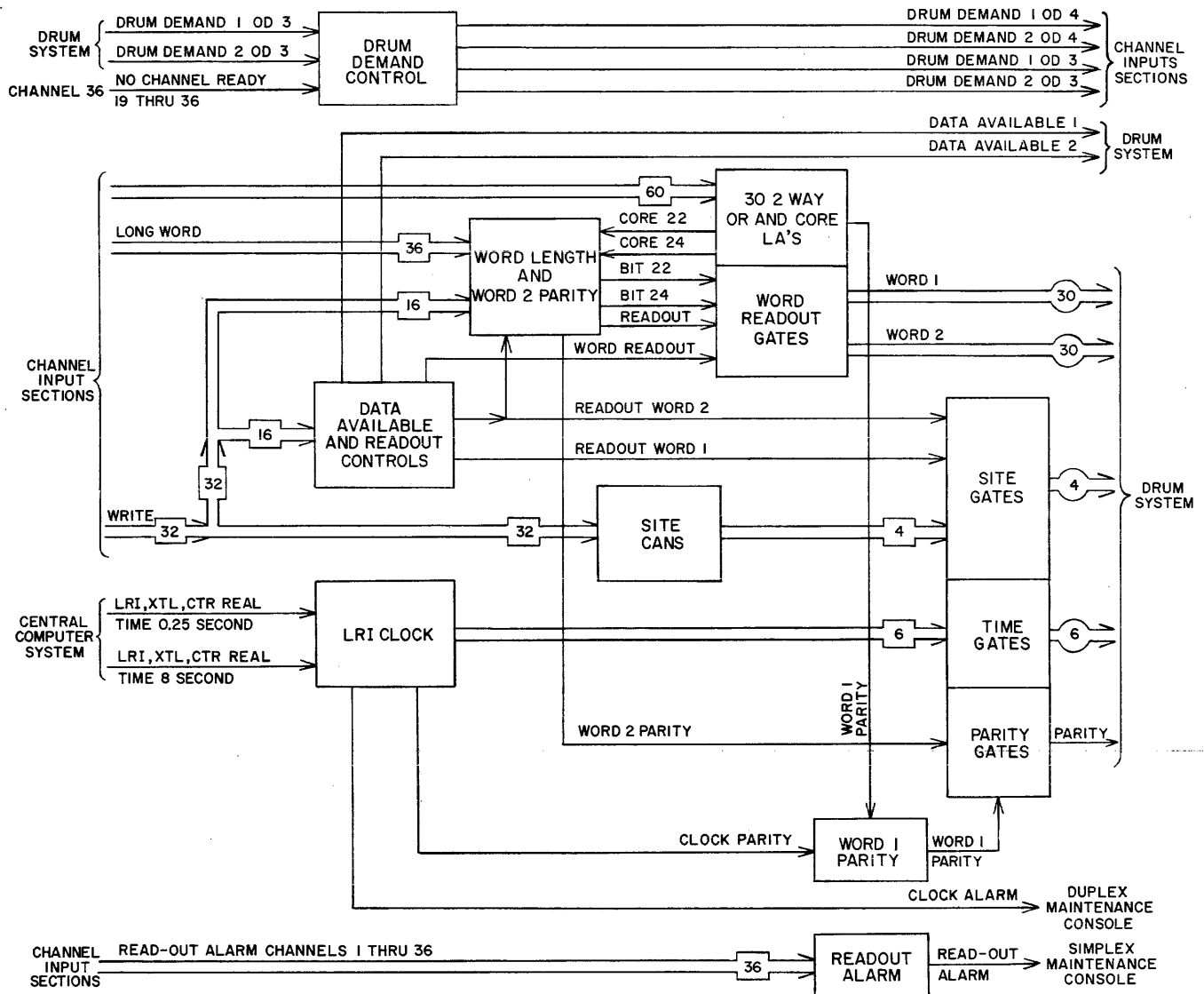


Figure 1-103. LRI Common Input Section

1.3.3 Checks

During loading of the core shift register, the data is also applied to the parity busy bit circuit where a parity count is taken on the LRI message. If the count is even as the last message bit is loaded, the parity-even level is up when the start-fast-shift pulse is applied to fast shift control. If the parity count is incorrect, no fast shift occurs, and the message (still in the core shift register) is lost when the core shift register is cleared by the parity-error pulse generated by the sync pulse of the next message. In addition, a parity-alarm signal is generated at the simplex maintenance console by the parity busy bit circuit if the parity-even level is down when the next sync pulse arrives.

If a busy-bit-short-word or busy-bit-long-word pulse from the core shift register is not received by the parity

busy bit circuit, the core shift register is cleared and any information contained therein is lost.

1.3.4 Transfer to Drum Field

Since the number of bits contained in an incoming LRI message exceeds 32, a slot of two drum registers on the LRI drum fields must be used for storage of each LRI message. The LRI drum fields are written on the OD side by a modification of the status mode which accommodates 2-word messages.

As an empty slot on one of the LRI drum fields comes into writing position, a demand (OD 3) pulse is applied to the drum demand control of the active common input section. (See fig. 1-103.) The drum demand control governs the transfer of data by status from all LRI channels onto the two LRI fields of the LOG drum. The 36 LRI channels are divided into two groups: LRI

channels 1 through 18 transfer data onto LRI field 2; LRI channels 19 through 36 transfer data onto LRI field 2. Since the data from both groups must pass through the same common input section to gain access to the Drum System, only one channel in one group can transfer data to a drum field at any one time.

The group of channels which is allowed to transfer data at one time is determined by drum demand control. This circuit receives the drum-demand (OD 3) pulses from the two LRI fields and supplies gated drum-demand (OD 3) and following drum-demand (OD 4) pulses to one group of channels at a time. If only one drum-demand pulse is received by the drum demand control, only the gated drum-demand pulses (at OD 3 and OD 4 time) interrogating those channels which can write on the ready field are generated. Thus, a drum-demand-field-1 pulse generates the gated drum-demand-1 pulses for channels 19 through 36. Similarly, a drum-demand-field-2 pulse generates the gated drum-demand-2 pulses for channels 1 through 18. However, if both drum-demand pulses from both fields are applied simultaneously to the drum demand control, only the gated drum-demand-1 pulses are generated. The gated drum-demand-2 pulses are not generated in this case unless none of the channels in the first group is ready to transfer data. The no-channel-ready-19-through-36 (unused drum-demand) pulse returned to the drum demand control allows the generation of the gated drum-demand-2 pulses from the drum-demand-field-2 pulse.

When the gated drum-demand (OD 3 and OD 4) pulses are applied to a ready channel, word 2 in the core buffers of that channel is transferred to the common input section. A write signal is also transmitted with word 2 and applied simultaneously to the site cans, word length and word 2 parity, and the data-available and read-out controls. After 10 microseconds (at which time word 2 has been transmitted to the Drum System), word 1 is transferred to the common input section.

The data bits of word 2 are applied to the common input section while the write signal causes four bits to be applied to the site gates and generates an odd-parity bit to the parity gates for drum word 1. The write signal also initiates readout of word 2 and transmits two data-available pulses to the Drum System, enabling each word to be written onto the drum field. The LRI clock receives pulses at 0.25-second intervals from the real time clock in the Central Computer System. A 6-bit count of these pulses is generated by the LRI clock and is added to LRI message word 1 (drum word 2) as it is written onto its LRI field. This clock time measures the 0.25-second intervals over a period of 8 seconds and is synchronized with the real time clock in the Central Computer System, allowing calculation of the time between the writing of a word on the OD side of the LRI

field and the reading of a word on the CD side by the Central Computer System. The LRI clock also supplies a clock-parity signal to the word 1 parity, indicating the parity of the 6-bit clock time as odd or even.

In summary, the common input section receives two words at different times from a channel input section and adds site identification and an odd-parity bit to word 2. Ten microseconds later, word 1 is transmitted to the Drum System with clock time and an odd-parity bit. Each word transferred by the common input section to the Drum System is accompanied by a data-available pulse which actuates the write circuit of the Drum System.

1.3.5 Alarm Conditions

Any alarm conditions detected in the LRI element are indicated to the Central Computer System by the LRI alarm signals which turn on alarm lights on the simplex and duplex maintenance consoles. The alarm conditions noted include incoming parity errors (discussed in 1.3.3) and read-out and clock alarms.

A read-out alarm is generated at the duplex maintenance console by a channel input section if that section has failed to transfer an LRI message from its word 1 and word 2 core buffers to the LRI field before another LRI message is fast-shifted into the word 1 and word 2 core buffers. (See fig. 1-101.) The channel-ready signal applied to the read-out core buffers indicates that the channel input section has an LRI message in the word 1 and word 2 core buffers ready for transfer to its LRI field. The message can be held in buffer storage until the succeeding message has been read into the core shift register. If the first message has not been transferred to the LRI field by the time the read-out core buffers receive the busy-bit-short-word or busy-bit-long-word signal for the second message, the first message is cleared from buffer storage, and a read-out alarm is generated.

The interval for which a message can be held in buffer storage without interfering with the processing of a succeeding message is the interval between generation of the channel-ready signal for the first message and generation of a busy-bit-short-word or busy-bit-long-word signal for the succeeding message, or approximately 39 milliseconds. The time required to search the entire LRI field once for an empty slot is only 20.5 milliseconds. Therefore, a read-out alarm can be generated only if no empty slot is found on an LRI field after almost two complete searches of that field; i.e., the read-out core buffers of a channel input section receive a channel-ready signal followed by busy-bit-short-word and busy-bit-long-word signals without any intervening gated drum-demand pulses only if the LRI field is completely full for more than one revolution of the LOG drum or if some equipment malfunction prevents ap-

INFORMATION NOT AVAILABLE

Figure 1-104. LRI Monitor Console

plication of the gated drum-demand pulses to the channel input section.

A clock alarm is generated by a common input section at the duplex maintenance console when the LRI clock undergoes a test for 1's and 0's. At the end of an 8-second counting sequence, the LRI clock is checked once for an all 0 content and once for an all 1 content. If an error should occur during these tests, a clock alarm neon at the duplex maintenance console is illuminated.

1.3.6 LRI Monitor

The LRI monitor can display information from the LRI common input sections prior to the delivery of this information to the Drum System. The LRI monitor enables operating personnel to check the quality of LRI information by visual means and to make a rough evaluation of LRI element operation independently of the Central Computer System. Figure 1-104 shows one such LRI monitor console.

The LRI monitor receives range and azimuth information in binary form from one channel input section through the LRI common section and converts this information to X and Y co-ordinate analog information for plan position indication (PPI) type display on the cathode-ray tube of the LRI monitor console. Four LRI monitor consoles are located at various advantageous points in the central, each having a separate function in the monitoring procedure. The LRI monitor consoles

are simplex while the associated information processing equipment is duplex to receive data from common input section A or B. The information-processing equipment is not duplex-switched, each being permanently connected to one common input section. Each of the four consoles contains provisions for selecting messages to be displayed. The LRI monitor consoles may also aid maintenance personnel in checking the operation of the channel input sections without interfering with element operation.

1.4 PROCESSED-DATA FORMS

Information processed by the LRI element is obtained from the LRI fields by the Central Computer System in the forms shown in figure 1-105. A search radar report contains the following information:

a. Drum word 1:

1. Run length, L11 through L13
2. Time delay, R2 through R6, with a precision of $\frac{1}{4}$ second
3. Message label, R7 through R11, identifying message as search radar, IFF, IFF with SIF, or height reply
4. Site identity, R12 through R15, identifying location of data source
5. Odd-parity bit

b. Drum word 2:

1. Range, L1 through L10 with a precision of $\frac{1}{4}$ mile
2. Azimuth, L11 through L15 (least significant) and R9 through R15 (most significant), with a precision of 0.088 degree ($\frac{1}{4,096}$ revolution)
3. Clock time, R1 through R6, with a precision of $\frac{1}{4}$ second
4. Odd-parity bit.

Those bits indicated as blanks or unused are written as 0's on the drum field.

The only difference between a search radar report and an IFF report is in the first drum word. The message label indicates that the report is an IFF report with or without SIF. If an SIF response is included, it occupies bits L1 to L13. An IFF report without SIF contains all 0's in these bit positions.

A height reply contains the following information:

a. Drum word 1:

1. Message label, R7 through R11, identifying report as a height reply
2. Site identity, R12 through R15
3. Odd-parity bit

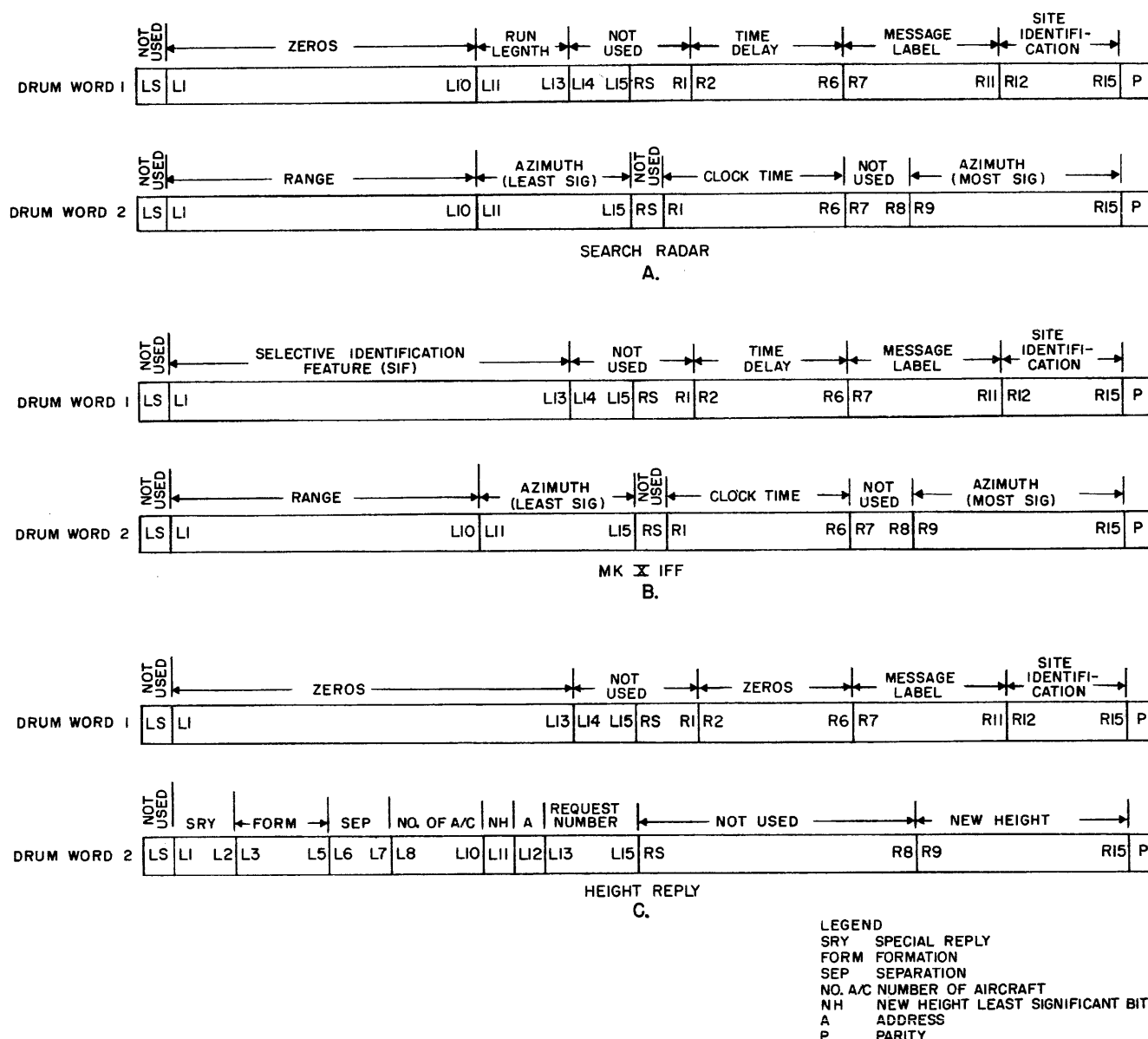


Figure 1-105. LRI Message Drum Field Layouts

b. Drum word 2:

1. Special reply, L1 through L2, containing a pre-arranged coded message
2. Formation, L3 through L5, a coded indication of the predominant arrangement of the aircraft reported as one target
3. Separation, L6 through L7, a coded indication of the distance between the aircraft making up the target
4. Number of aircraft, L8 through L10, an approximation of the number of aircraft
5. New height, R9 through R15 and L11 (least significant bit)
6. Address, L12, indicating which of the two height-finders at the P site is replying
7. Request number, L13 through L15, allowing identification of the reply with the target on which a height report was requested
8. Odd-parity bit.

Those bits labeled as blank or not used contain 0 when written onto the LRI fields. No clock time is included in height replies.

1.5 DUPLEXING

Duplexing in the LRI element is restricted to the common input section, which is duplicated in computer A and computer B. The channel input sections are sim-

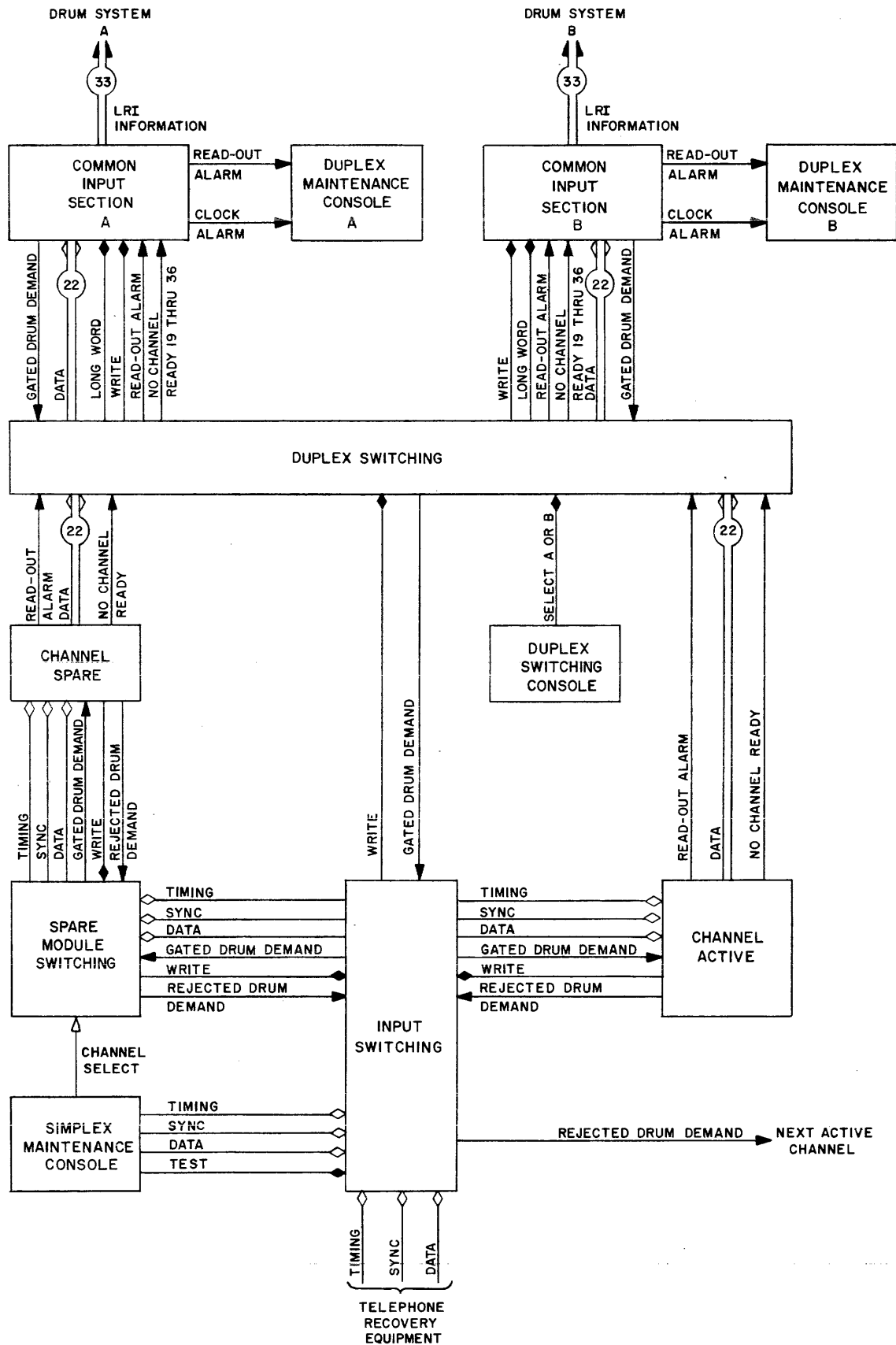


Figure 1-106. LRI Element, Simplex and Duplex Switching

plex. Within each group of channel input sections there are two spare channel input sections which can be switched to substitute for any other pair of channel input sections in that group. This substitution allows replacement or removal of a defective channel input section for testing and maintenance operations.

The basic switching scheme for the LRI element is shown in figure 1-106. Simplex switching, controlled at the simplex maintenance console, provides status control of all channels (making channels active or standby) and substitution of spare channels for normal channels. Duplex switching, controlled at the duplex switching console, directs the outputs of active channels to the active common input section and the outputs of standby channels to the standby common input section. Since the standby common input section is connected in the standby computer, a channel in standby can be tested, using the Central Computer System in programmed automatic maintenance operations.

If a channel input section is active and is to be switched to standby for test purposes or because that channel is giving rise to too many alarms, two switching operations are required to substitute the spare channel input section of that group for the channel being replaced. The spare channel input section must be switched to receive the data intended for the channel input section being replaced. (Spare switching actually replaces two channel input sections, since they are physically located in one module and switching of the entire module is simpler than switching one channel.) The data input switch for the spare channel must be turned to select telephone data rather than test input, and the spare channel must be switched to active status. Simi-

larly, the channel being replaced must be switched to standby or to off and its data input switch set to select test signals or turned off.

The channel-select signal from the simplex maintenance console places the spare channel input section electrically in the line of information and control signal flow normally occupied by the channel being replaced. Thus, the spare channel is interrogated by gated drum-demand pulses from the channel preceding the replaced channel and passes on unused drum-demand pulses to the channel numerically following the replaced channel. The spare channel can be used to replace a channel in standby status for comparison of results during tests, thus allowing localization of troubles during a test to either the channel input section or the common input section used in the test. The standby common input section supplies gated drum-demand pulses for interrogation of standby channel input sections just as the active common input section supplies gated drum-demand pulses for interrogation of active channel input sections. The site identity code added to messages passed through the spare channel when it is switched in to replace another channel is identical with the code which would be generated if the replaced channel input section were actually being used. The write signal from the spare channel is applied through spare module and input switching to go to duplex switching on the same line on which the write signal from the replaced channel would arrive. Since the line on which this signal arrives determines the site identity code added to the message passing through the common input section, the site identity code is that which would be added if the message were supplied by the replaced channel.

SECTION 2

GAP-FILLER INPUT ELEMENT

2.1 ELEMENT DESCRIPTION

2.1.1 Element Function

The gap-filler input (GFI) element processes radar data generated by short-range radars which are arranged to fill gaps in the radar coverage provided by the long-range search radars supplying data to the long-range radar input (LRI) element. These short-range radar sets, called gap-filler radars, operate unattended. Therefore, the filtering of data from these sets must be performed by the GFI element, unlike the LRI element, which receives filtered data from the P sites.

The GFI element receives radar data from each gap-filler radar. This data must be displayed to a radar mapper operator to allow manual filtering, then used to generate range and azimuth information on each target detected by the gap-filler radar set and passed by the mapper operator. As is done for LRI messages, the source of each target report must be included with the report, as well as the time at which it is received.

2.1.2 Element Logic

The GFI element is divided into four sections: data conversion, mapper, counter, and the common section. Each simplex channel of the GFI element contains a data conversion section, mapper section, and counter section. The common section (which is duplex) can receive data from all channels for transfer to the Drum System. (See fig. 1-107.)

The mapper section for each GFI channel receives data from one gap-filler radar through the data conversion receiver associated with the telephone channel from that radar. The mapper section generates a plan position

indicator (PPI) display from the gap-filler data. Filtered targets are detected by a photoelectric tube mounted over the mapper display and indicated to the counter section.

The counter section receives azimuth and range data as well as filtered target signals in order to generate range and azimuth measurements on each filtered target. The counter section holds the generated target report until the Drum System is ready to receive it. Upon request of the GFI field, the target report is transferred to the common section and is then placed in a drum register. The common section adds site identity bits to each report written on the GFI field; the Drum System adds the time of writing to each report.

The channel sections of the GFI element are simplex because failure of one channel section would not greatly hinder the overall operation of the central. However, if a common section should fail, it would disable the entire GFI element. The GFI common section is therefore duplex; a common section is included in both computer A and computer B. (See fig. 1-108.) The GFI channel sections receiving information from the GFI radars transfer this information to the active common section, whether A or B. One of the common sections is designated active and the other standby by the duplex switching console. The substitution of a spare channel for one that is faulty or that requires maintenance is controlled at the simplex maintenance console.

2.1.3 Data Source

The data source for each GFI channel is an unattended short-range, moving target indicator (MTI)

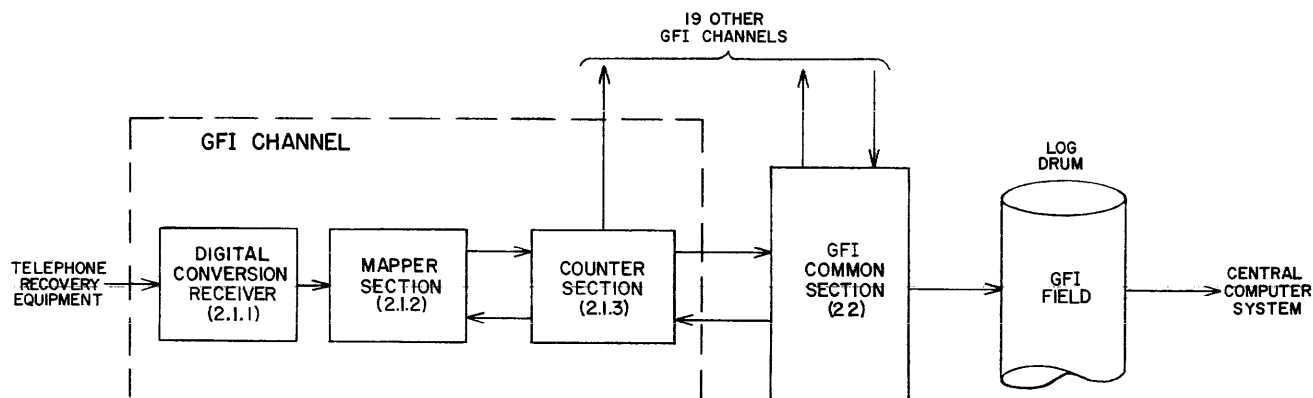


Figure 1-107. Gap-Filler Input Element

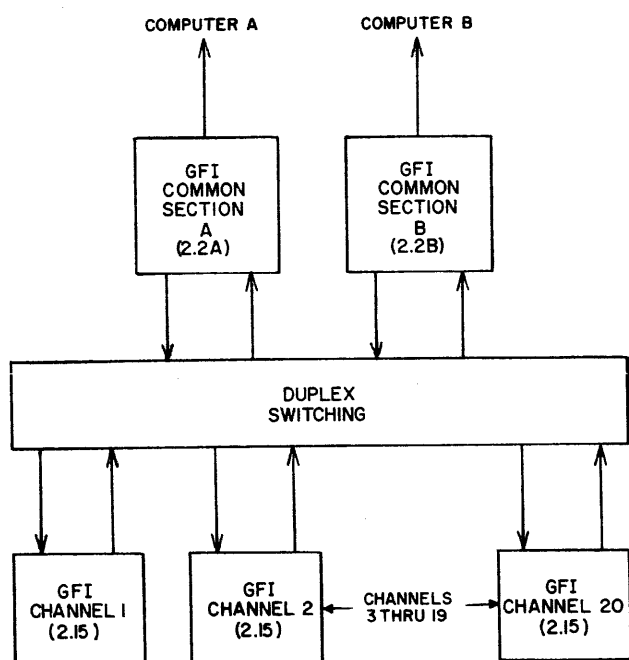


Figure 1-108. GFI Element, Duplexing

equipped radar. The data produced by this radar must be modified for transmission over an audio-frequency telephone channel. The modification is performed by quantizing of the coverage area.

The process of quantization involves the division of a continuously variable quantity into units of fixed dimension. As applied to the coverage area of a gap-filler radar set, that area is divided into azimuth sectors of fixed included angle with each azimuth sector divided into range segments of fixed radial length. (See fig. 1-109.) Thus, the coverage area of the gap-filler radar set is divided into units of fixed dimensions called quanta, whose angular width and range interval are predetermined.

The angular dimension of a quantum is approximately equal to the width of the gap-filler radar beam and is therefore roughly equal to its azimuth resolution. Measurement of azimuth angle allows the generation of position data with a maximum error of $\pm 1/3$ mile at a range of 30 miles. Measurement of range to a precision of $3/4$ mile is readily attainable by any radar. Thus, the problem becomes one of reducing the amount of data obtained by the gap-filler radar to an amount which can be transmitted over an audio-frequency channel with a precision matching the quantization of the coverage area.

The transmission of radar video information requires a communication channel with a bandpass of several megacycles. If the echoes received after a single radar pulse are accepted with a range resolution of $3/4$ mile, echo pulses that are spaced as little as 6 microsec-

onds apart must be recognized separately, thus requiring a fundamental frequency response as high as 160 kilocycles per second. If it is assumed for this discussion that the gap-filler radar sweeps its coverage area at the rate of 6 revolutions per minute, each azimuth sector is swept in 40 milliseconds $\left(\frac{10 \text{ seconds per revolution}}{256 \text{ sectors per revolution}} \right)$. If the gap-filler radar uses a pulse repetition frequency (prf) of 1,600 pulses per second, it can search each azimuth sector to a maximum range of 50 miles, producing 64 pulses (1,600 pulses per second \times 0.04 second) per sector. The 64 echo trains produced by these pulses describe the contents of a single azimuth sector. (See fig. 1-110, part A.) These echo trains can be integrated (summed over a period of time on an electrostatic storage tube) so that those target echoes which fall within a corresponding $3/4$ -mile interval of successive echo trains are added together to produce a composite report indicating the presence or absence of targets within the quantum of coverage area defined by the azimuth sector and range segment. (See fig. 1-110, part B.) The process of integration improves the resultant signal-to-noise ratio of the information, since randomly occurring noise pulses will not build up as large a sum in any one quantum as do regularly occurring target pulses. The number of targets within a given quantum is lost, since the echoes from a single target add up to a value sufficient to saturate the quantum storage box.

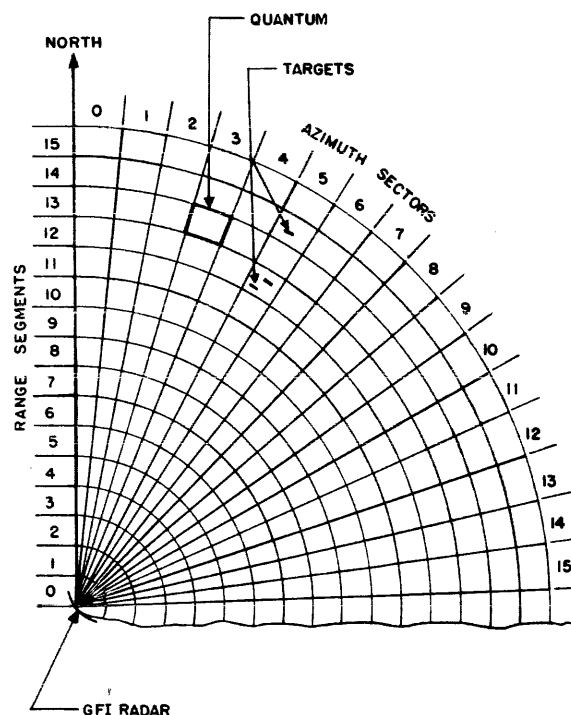


Figure 1-109. Quantization of GFI Radar Coverage Area, Simplified

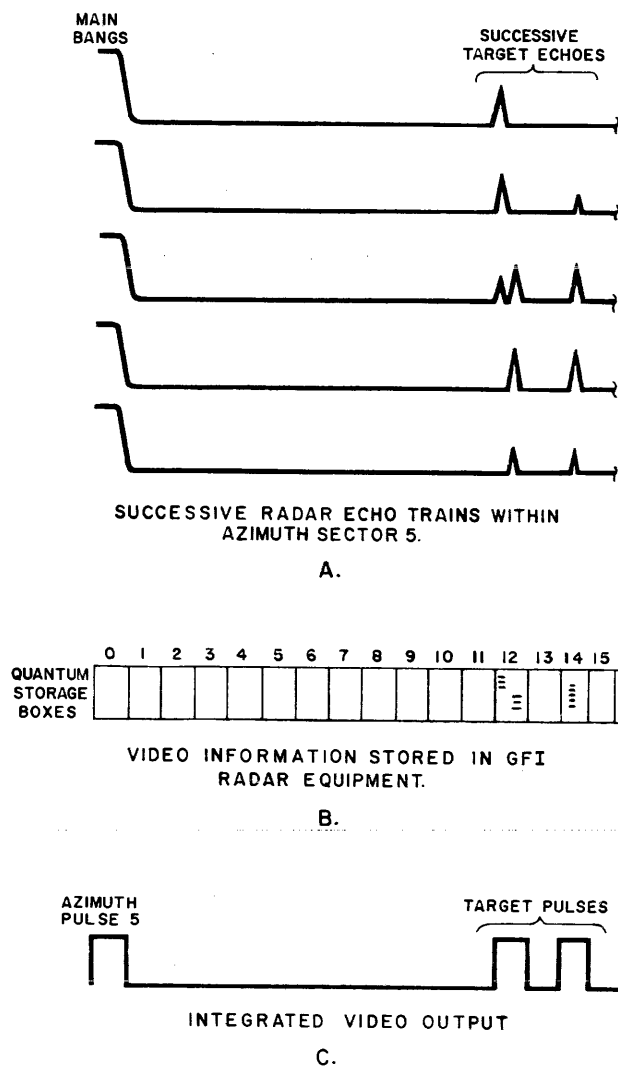


Figure 1-110. Video Integration of GFI Radar Data

The rate of GFI data transmission is set at around 1,600 quanta per second. With this figure fixed, the number of quanta and the maximum range reported within a given azimuth sector depend upon antenna rotation rate. The number of reportable quanta per sector must vary with the speed of antenna rotation, since the rate of transmission of quantum reports is constant, whereas the rate of acquisition of information on an azimuth sector varies with the antenna speed. For example, the number of reportable quanta per sector is determined as follows:

$$\frac{1,600 \text{ quanta reported per second} \times 60 \text{ seconds}}{\text{antenna speed in rpm} \times 256 \text{ sectors}} = \text{reportable quanta per sector.}$$

If the radar antenna rotates at 6 rpm, 64 quanta per sector are reportable, covering a range of 48 miles. The slowest allowable antenna speed, 2 rpm, makes it pos-

sible to report on 188 quanta per sector, covering a range of 141 miles. The fastest antenna speed, 10 rpm, makes it possible to report on only 38 quanta, covering a range of 28.5 miles. Variations in antenna speed only change the reportable number of quanta per sector without changing the maximum range of the radar set itself. The radar set cannot examine those quanta beyond its maximum range. Therefore, although time exists to send the reports, those quanta cannot be meaningfully reported to the central since no information on those quanta can be acquired by the radar set.

The integrated radar data transmitted to the central thus appears as a serial train of pulses indicating the start of an azimuth sector, followed by pulses positioned in time between successive azimuth pulses to represent those quanta within the sector being reported which contain targets. (See fig. 1-110, part C.) The azimuth pulse occupies the time which would normally be used to report those targets within the first quantum (the first $\frac{3}{4}$ mile of range), thus effectively removing some ground clutter which might otherwise appear as targets. The geographical reference for target reports is established by transmitting a double azimuth pulse, called a north pulse, as the gap-filler radar sweeps through true north.

2.2 ELEMENT OPERATION

2.2.1 General

The GFI element processes data from gap-filler radar sets. Each radar supplies integrated radar data over a telephone channel to a digital conversion receiver associated with the channel. Each channel contains a digital conversion receiver section, a mapper section, and a counter section. The operation of only one channel is discussed, since all GFI channels are identical in operation.

2.2.2 Digital Conversion Receiver

The digital conversion receiver receives and converts GFI radar data from the telephone recovery equipment. This data is then transferred, in serial form, to the GFI mapper section. Each GFI channel contains one digital conversion receiver section which converts sinusoidal telephone line signals to standard signals. The GFI mapper section can accept and process the GFI radar data represented by these signals.

2.2.3 Mapper Section

The mapper section of each GFI channel generates a display in PPI form of the quantized data furnished by the gap-filler radar and the digital conversion receiver section feeding that channel. The display is produced on the cathode-ray tube housed in a mapper console. (See fig. 1-111.) The mapper operator can eliminate undesirable target indications by covering their display on the cathode-ray tube face with a fluid that is opaque to

INFORMATION NOT AVAILABLE

Figure 1-111. GFI Mapper Console

blue light. This fluid prevents detection of the unwanted target display by the photomultiplier tube mounted on the arm overhanging the display tube.

The logic of the mapper section is shown in figure 1-112. The section receives azimuth, north, and target pulses from the digital conversion receiver section associated with that GFI channel. The azimuth pulses are filtered and used to generate the PPI display. The north pulse is used to keep the display synchronized with the actual sweep of the gap-filler radar. Targets appear on the mapper display tube as an initial blue flash followed by a long-persistence yellow glow. Since the photomultiplier tube is covered by a blue filter, only the initial blue flash is detectable if not masked by the mapper operator. Filtered target pulses are thus passed on to the counter section in synchronism with the application of the target pulses from the gap-filler radar. Those target reports accepted by the counter section cause that section to return a reintensify signal to the mapper section. The reintensify signal causes brighter display of accepted targets to the mapper operator. The target-data-inhibit level is sent to the counter section to prevent transfer of target reports to the Drum System if excessive noise is detected on the GFI telephone channel or if the mapper display is not synchronized with the gap-filler radar sweep. Certain incoming data measurements are made by the Central Computer System and are indicated to the mapper operator by the warning lights located above and to the right of the display tube on the console. (See fig. 1-111.)

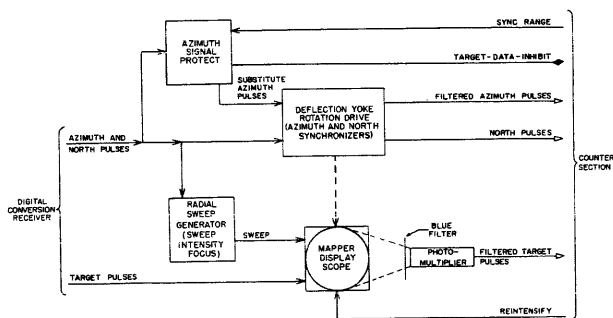


Figure 1-112. GFI Mapper Section,
Simplified Diagram

2.2.4 Counter Section

The counter section generates range and azimuth counts on each filtered target and supplies these counts to the GFI common input section for transfer upon demand to the GFI field of the LOG drum. The counter section receives range (or timing) pulses from the digital conversion receiver associated with the channel. This range information is synchronized with drum timing pulses and applied to the range counter. (See fig. 1-113.) The range counter keeps a running count of range in $\frac{3}{4}$ -mile increments between azimuth pulses, since the range counter is cleared by each azimuth pulse. Thus, at any given instant, the range counter contains the range co-ordinate of the quantum being reported to the GFI channel at that instant. Similarly, the azimuth counter maintains a running count of azimuth in increments of 1.4 degrees ($1/256$ circle) between north pulses. At any given instant, the azimuth counter contains the azimuth co-ordinate of the quantum being reported to the GFI channel at that instant. Therefore, the azimuth and range counters together contain the designation of the one quantum being reported to the GFI channel.

When a filtered target pulse is received by the counter section, it returns a reintensify signal to the mapper section and applies a counter-shift pulse to the

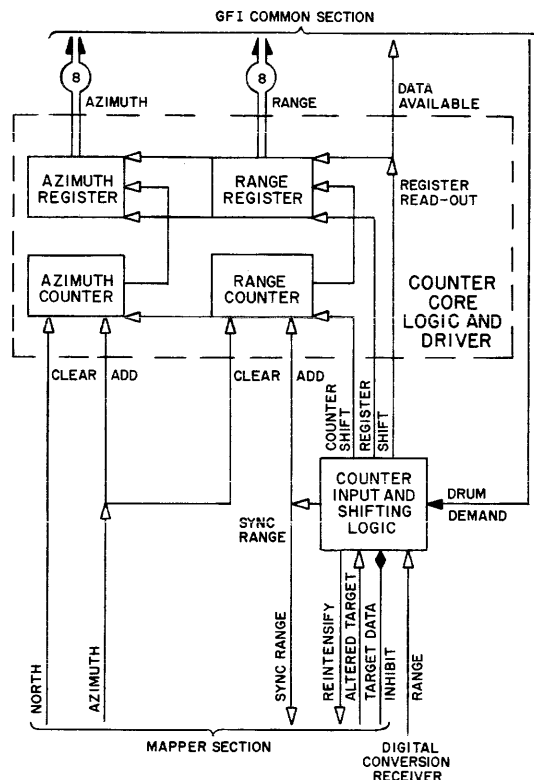


Figure 1-113. GFI Counter Section,
Simplified Diagram

azimuth and range counters. (The target-data-inhibit level will prevent these actions if it is up.) The counter-shift pulse transfers the azimuth and range counter contents to their respective read-out registers. The azimuth and range registers now contain the co-ordinates of the filtered target and are ready to transfer this data to a GFI field. A drum-demand pulse from the GFI common input section generates a register read-out signal which transfers the azimuth and range information to the GFI common input section. If a drum-demand pulse were applied to the counter section when no target data was ready for transfer, the drum-demand pulse would be passed on to the next GFI channel.

2.2.5 GFI Common Section

The GFI common section controls the transfer of data by status mode from the GFI channels onto the GFI field of the LOG drum. Since the data from all the active GFI channels must pass through the same transfer circuits in getting to the active Drum System, only one channel can transfer data to a drum field at any one time.

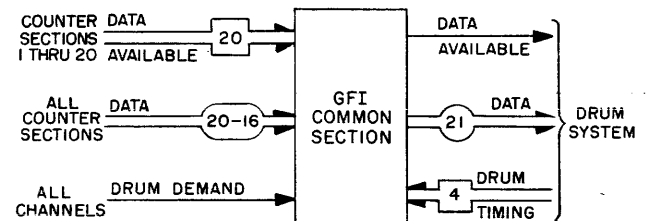


Figure 1-114. GFI Common Section,
Simplified Diagram

The GFI common section transfers data from one channel at a time to the Drum System. (See fig. 1-114.) The GFI drum field supplies a drum-demand pulse when an empty drum register is coming into writing position. The drum-demand pulse interrogates all the active channels in a predetermined sequence. When a channel is found with information available for transfer, that information is transferred to the GFI common section together with a data-available pulse. The common section adds site identification to the information as it is transferred to the Drum System. Clock time is added to the drum word by the Drum System as that word is placed on the GFI drum field. The data-available pulse actuates the write circuits within the Drum System, enabling the GFI drum word to be written on the drum field.

2.2.6 GFI Monitor

The GFI monitor allows operating personnel to visually check the quality of GFI data and mapping or filtering operations. The GFI monitor can also be used as an aid in the detection of malfunctions within the GFI

INFORMATION NOT AVAILABLE

Figure 1-115. GFI Monitor Console

channels and to provide photographic records of GFI data for program evaluation and training purposes. The GFI monitor consoles, as shown in figure 1-115, display the same type of data as do the gap-filler mapper consoles. However, the GFI monitor consoles have data selection features that are not included in the gap-filler mapper consoles. For example, each of the GFI monitor consoles is able to select raw data (unfiltered data displayed at a gap-filler mapper console), filtered data (data passed by a mapper console), or raw data plus intensified filtered data (reintensified target reports). Further, the GFI monitor consoles may select oriented or nonoriented displays. Oriented displays consist of those displays which are immediately available to the GFI monitors because the PPI scan at the GFI monitor console is rotating in synchronism with the antenna of the selected GFI radar set (hence, one GFI channel).

A nonoriented display is that which may not be immediately available to the GFI monitor console because the PPI scan rotation of the monitor console is operating asynchronously with the rotation of the GFI radar set when a channel is selected for examination. Pushbutton modules are provided at the GFI monitor consoles to enable the operator to select any one of the GFI channels supplying data to the GFI common section. When a pushbutton is depressed, the selected data handled by that channel is displayed at the monitor consoles. The camera-equipped monitor console is provided with other pushbutton modules that control the operation of the camera.

The GFI monitor consoles are each connected to special data-handling units. The GFI monitor camera console receives data and certain control signals from the camera console module, while the other GFI monitor console receives data signals from the monitor console module. The auxiliary console of the supervisor also provides camera control signals to the GFI monitor camera console and to the camera console module to control the operation of the GFI monitor camera console.

2.3 INFORMATION FORM

The form in which a GFI message appears on a GFI field is shown in figure 1-116. Each drum word contains the following information:

- a. Range, LS through L7, with a precision of 3/4 mile
- b. Clock time, L10 through L14, equivalent to clock time in an LRI message but added to a GFI message by the Drum System
- c. Azimuth, RS through R7, with a precision of 1.4 degrees (1/256 circle)
- d. Site identity, R11 through R15, identifying the origin of the target co-ordinates within the message.

It should be noted that no parity is added to a GFI message.

2.4 DUPLEXING

Duplexing of the GFI element is restricted to the GFI common section which is duplicated in both com-

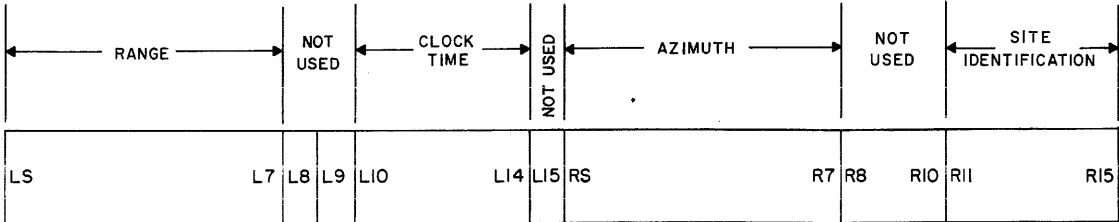


Figure 1-116. GFI Message Drum Field Layout

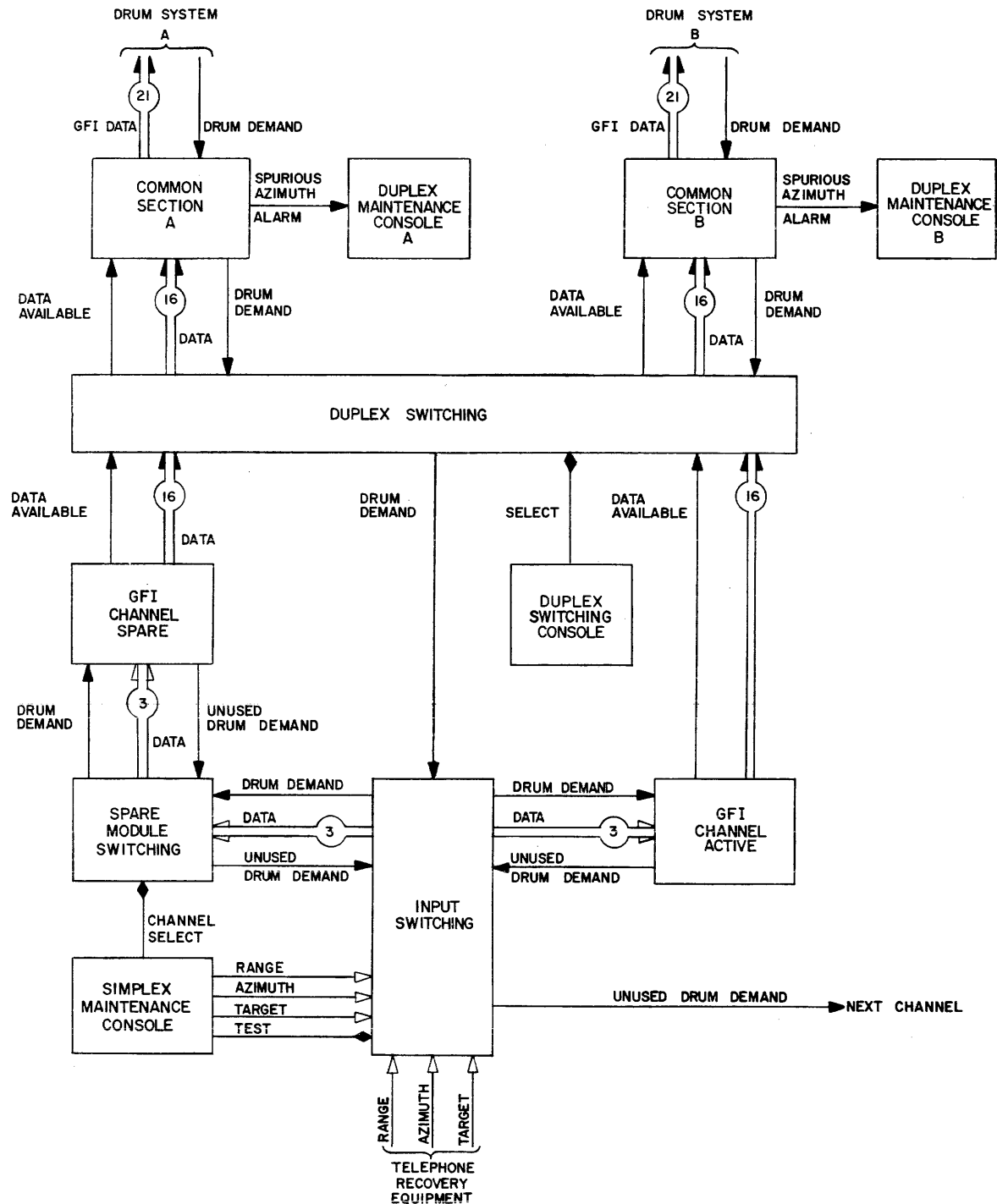


Figure 1-117. GFI Element, Simplex and Duplex Switching

puter A and computer B. The GFI channel sections are simplex. There are four spare GFI channels which can be switched to substitute for any of the active GFI channels. This feature permits replacement of a defective channel or the isolation of a GFI channel for test and maintenance operations.

The switching scheme for the GFI element is shown in figure 1-117. Simplex switching, controlled at the

simplex maintenance console, provides status control of all channels (making channels active or standby) and substitution of spare GFI channels for normally active channels. Duplex switching, controlled at the duplex switching console, directs the outputs of the active channels to the active common section and the outputs of the standby channels to the standby common section. Since the standby GFI common section is connected in

the standby computer, a channel in standby status can be tested using the standby Central Computer System in programmed maintenance operations.

If a GFI channel is active and is to be switched to standby status either for test purposes or because that channel is defective, two switching operations are required to substitute the spare channel for the channel being replaced. The spare GFI channel section must be switched to receive the data intended for the channel being replaced. (Spare channel switching actually replaces two GFI channels, since they are physically located in one module and switching of one module is simpler than switching one channel.) The data input switch for the spare GFI channel must be turned to select telephone line data rather than test input signals, and the spare GFI channel must be switched to active status. Similarly, the channel being replaced must be switched to standby or off and its data input switch set to select test signals or turned off.

The channel-select signal from the simplex maintenance console places the spare GFI channel section electrically in the line of information and control signal flow normally occupied by the GFI channel section being

replaced. Thus, the spare GFI channel section is interrogated by drum-demand pulses from the channel preceding the replaced channel and passes the unused drum-demand pulses to the following GFI channel section. A spare GFI channel can be used to replace a GFI channel in standby status for comparison of results during tests, thus allowing localization during a test to either the GFI channel section or common section used in the test. The standby GFI common section supplies drum-demand pulses to the standby GFI channel sections just as the active GFI common section supplies drum-demand pulses to the active GFI channel sections. Since the data-available pulse from the spare GFI channel section replacing an active GFI channel section follows the same route through the duplex switching to the active GFI common section, the same site identity code is generated for the active spare GFI channel as for the active GFI channel which it replaces. Similarly, a spare GFI channel in standby status transmits a data-available pulse through the same line as the channel it replaces, thereby causing the same site identity to be added to the GFI drum word as is normally added for the replaced standby GFI channel section.

SECTION 3

CROSSTELL INPUT ELEMENT

3.1 ELEMENT DESCRIPTION

3.1.1 Element Function

The crosstell (XTL) input element of AN/FSQ-7 Combat Direction Central enables a central to receive information from adjacent centrals on aircraft moving out of their coverage area into the coverage area of this central. (See fig. 1-118.) Adjacent centrals transmit this type of information from the G/G output storage section of their Output System over audio-frequency telephone channels. The XTL input element receives this information in serial form and converts it into parallel form for transfer to the XTL field of the Drum System to which the Central Computer System has access.

3.1.2 Element Logic

The XTL element contains three major divisions: the XTL channel input sections (one per telephone channel), the spare channel input section, and the XTL common section which is common to all channel input sections. (See fig. 1-119.) An XTL channel input section receives XTL messages from the telephone recovery equipment associated with it, converts the message data from serial to parallel form, checks the data for errors in transmission, and then stores the converted and checked message until the XTL field of the Drum System can accept it.

The XTL common section transfers data from an XTL channel input section onto the XTL field upon

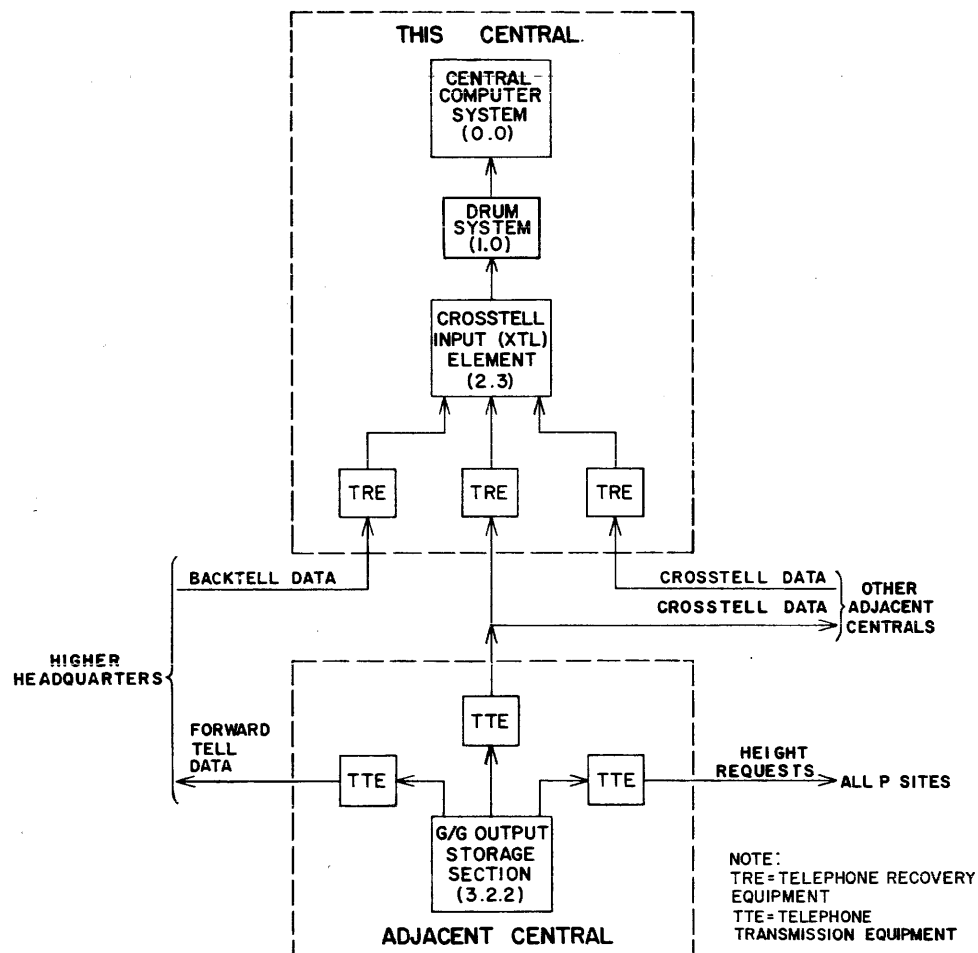


Figure 1-118. Crosstelling

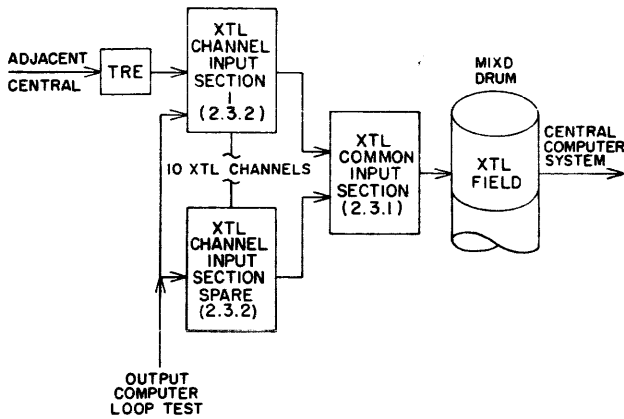


Figure 1-119. Crosstell Input Element

request from the Drum System. The common section adds clock time, site identity, and parity bits to each XTL message placed on the XTL field.

Duplexing in the XTL element is restricted to the XTL common section. (See fig. 1-120.) Failure or faulty operation of the XTL common section would render the entire XTL element useless; the XTL common section is therefore duplicated, with the result that one XTL common section operates with computer A and one with computer B. Computer A or computer B, hence XTL common section A or B, is placed in active or standby status by the duplex switching console. The XTL channel input sections are simplex, since failure of one channel input section would not seriously impair the XTL operation. The designation of active or standby status of any one of the XTL channel input sections is controlled by the simplex maintenance console. The XTL channel input sections on active status supply XTL information to the XTL common section which is also on active status. Similarly, the XTL channel input sections on standby status supply XTL or test data to the standby XTL common section.

3.1.3 Data Form

A XTL message containing data in serially interleaved form is transferred over telephone facilities at a 1,300-pulse-per-second rate. The five half-words assembled by the adjacent central into a XTL message are interleaved so that corresponding bits of each half-word are transmitted in sequence. Thus, the sign bits of the five half-words are transmitted in the sequence S_1 , S_2 , S_3 , S_4 , and S_5 , then the first bit of each half-word in a similar sequence, etc. (See fig. 1-121.) An even-parity bit is added to each half-word, making a total of 85 data bits transmitted. Another 7 time pulses are required to satisfy the timing requirements of the XTL element, making the entire message 92 bits long and requiring

92/1300 second (approximately 71 milliseconds) for transmission.

Since XTL messages are transmitted over a party line connected to all centrals adjacent to the transmitting central, each message must contain address bits, either addressing a single central or indicating that all centrals should accept the message. The destination address is contained in bits R11 through R15 of the fourth half-word and appears as pulses 64, 69, 74, 79, and 84 of the interleaved message.

3.2 ELEMENT OPERATION

3.2.1 General

The operation of the XTL element can best be described by considering its operation in three phases: read-in, checks, and transfer to drum. Only one XTL channel input section is described, since all channel input sections are identical in operation.

3.2.2 Read-In

An XTL channel input section receives separate sync, data, and timing signals from the telephone recovery equipment associated with the telephone channel

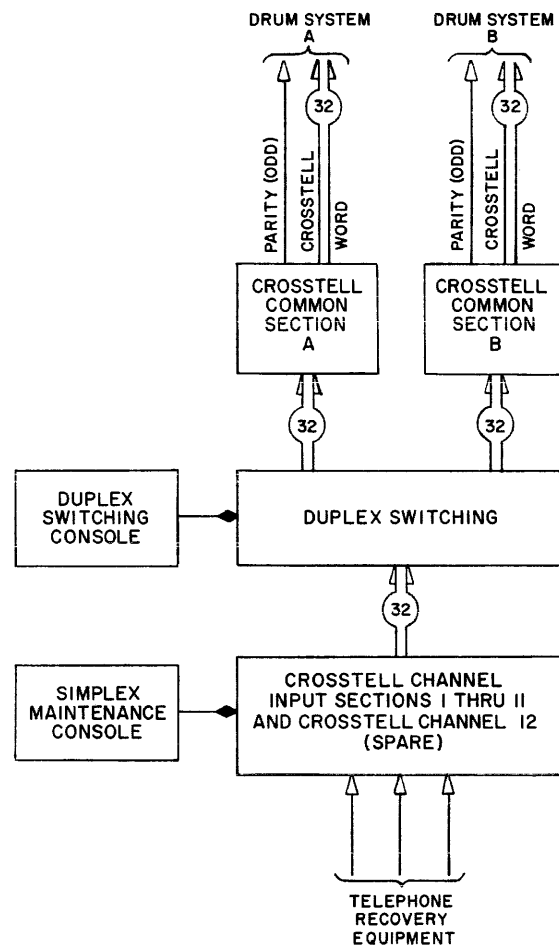


Figure 1-120. Crosstell Duplexing

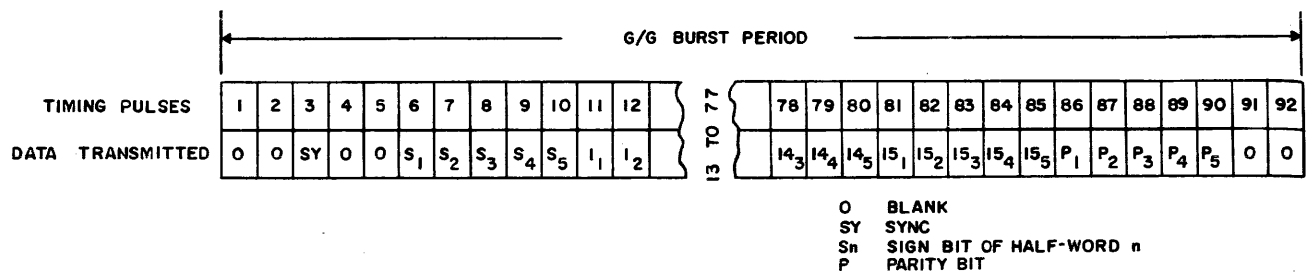


Figure 1-121. Crosstell Message, Interleaved Telephone Line Layout

feeding that section. (See fig. 1-122.) These signals are applied to the data converter and synchronizer, where they are synchronized with drum timing without changing their frequency of occurrence.

Read-in begins with the passage of a sync pulse through the input synchronizer. The sync pulse clears the data distributor, buffer storage, parity check, address check, the 25-core counter, and the shift pulse generator. The first data bit is received after two blank timing pulses. This data bit is written into the first 7-core shift register of buffer storage under control of the distribution pulses from the 5-core counter in the data distributor. The 5-core counter produces a different distribution pulse each time it is stepped by a timing pulse. The first time it is stepped, a data bit is written into the first buffer storage register. The second time it is stepped, a data bit is written into the second buffer storage register, and so on. In this manner, the sign bits of the five original half-words are each written into a separate register in buffer storage. As the fifth bit is written into buffer storage, the data distributor steps back to its starting condition and generates an end-carry pulse (5-counter = 5). Therefore, the sixth data bit will be written into the first buffer storage register, and the sorting of interleaved bits into five separate half-words continues throughout read-in.

The data distributor end-carry pulse (5-counter = 5) is applied to the core shift pulse generator to produce a buffer-shift pulse. This buffer-shift pulse advances the data in buffer storage one core after every five data bits are written into buffer storage. The shift leaves the first core in each buffer storage register empty and ready to receive another data bit. A step-25-counter pulse is also generated from the 5-counter = 5 pulse which allows the 25-core counter to maintain a running count of the number of buffer storage shifts. The data in buffer storage and the count of the 25-core counter are advanced 260 times per second (1300/5) while data is being received from the data counter receiver.

The seventh shift of buffer storage transfers data from buffer storage into main storage. Thereafter, main storage must be shifted whenever buffer storage is shifted until read-in is completed. Therefore, after 25-

counter = 7, the shift pulse generator applies a shift pulse to both buffer storage and main storage each time it receives a 5-counter = 5 pulse.

When 17 shifts have been completed, the entire XTL message is contained in the XTL channel input section; the first 10 bits of each half-word are in a main storage register, and the last 6 bits plus parity are in a buffer storage register. The message can now be fast-shifted from buffer storage to main storage since the shift rate need no longer depend upon the incoming data rate. Accordingly, at 25-counter = 18, fast shift at a 50,000-pulse-per-second rate begins. The fast-shift pulses, also applied to the 25-core counter, end at 25-counter = 25 with the entire message in main storage. The entire fast shift occurs in 0.14 millisecond. Read-in is now complete, with buffer storage cleared to receive the beginning of the next XTL message.

3.2.3 Checks

Each XTL message is checked for correct address and parity during read-in. The address check takes place from 25-counter = 12 through 25-counter = 16. The address bits of half-word 4 are compared serially with the address of the receiving central in the address check circuit. The first four address bits indicate a specific central. If any of the first four address bits does not compare with the address of the receiving central, a wrong-address signal is sent to parity check, preventing generation of the read-out-address-parity-check level necessary for transfer of the message to the XTL field. If there is a 1 in the fifth address bit position of half-word 4, the message is addressed to all centrals. Accordingly, detection of a 1 in that position at 25-counter = 23 by address check generates an all-receive signal, which overrides any wrong-address signal sent to parity check by allowing generation of the address-parity-check level.

Parity check receives the data bits of each half-word separately and performs a parity count on each half-word. An incorrect parity count of one or more half-words of the message prevents generation of the address-parity-check level necessary for transfer of the message to the XTL field. Thus, parity check combines the results of the parity checks and address checks per-

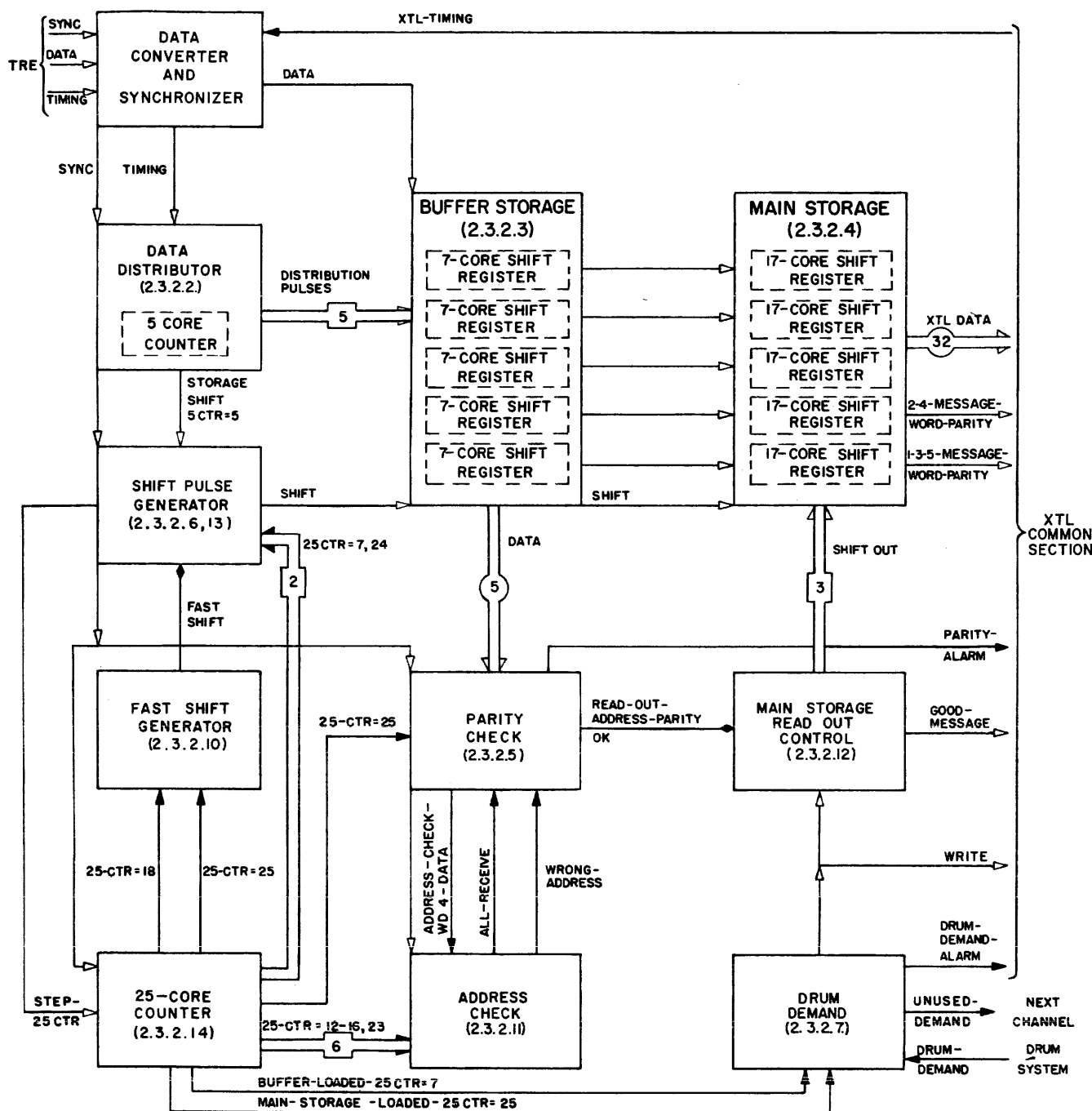


Figure 1-122. XTL Channel Input Section

formed on each XTL message during read-in. In addition, if a parity error is detected, a parity-alarm signal is sent to the simplex maintenance console at 25-counter = 25. The detection of an incorrect address or of a parity error prevents the transfer of the XTL message to the Drum System.

3.2.4 Transfer to Drum Field

The size of an XTL message requires the use of three drum registers on the XTL field for the storage of each

message. Therefore, the XTL field is written on the OD side by marker status mode. (Refer to Ch. 4.) As a slot of three empty registers on the XTL field moves into writing position, the Drum System generates a drum-demand signal. This signal interrogates the XTL channel input sections to find a channel ready to transfer data onto the drum field. The signal is applied to the drum demand circuit of the first channel. If that channel is ready to transfer a message, the transfer is initiated.

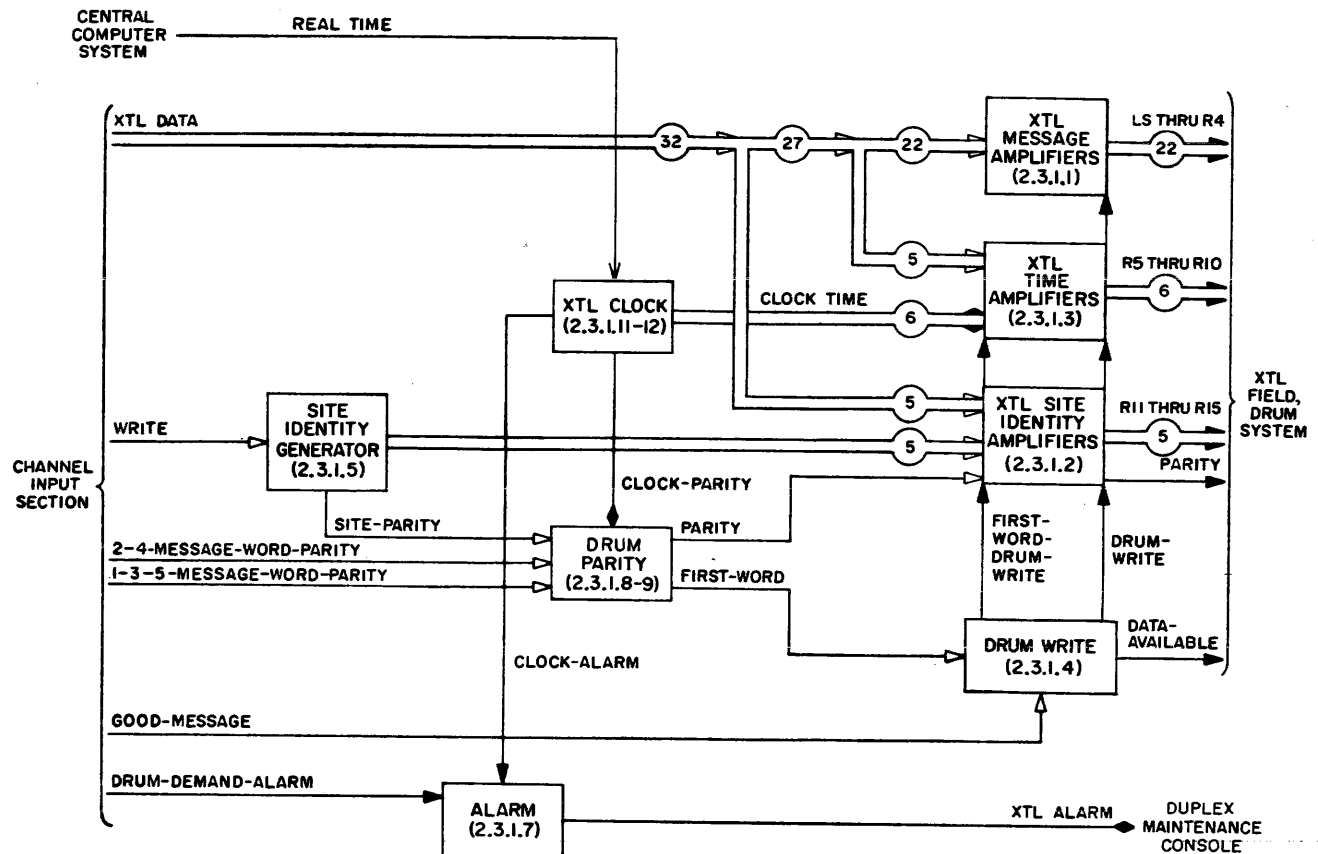


Figure 1-123. XTL Common Section

If the channel is not ready, the drum-demand signal is passed on to the next channel.

The drum demand circuit of an XTL channel input section accepts the drum-demand pulse if it has received a main-storage-loaded signal (25-counter = 25) within 28 milliseconds before receiving the drum-demand signal. The drum demand circuit generates a write pulse which is applied to main storage read-out control and to the XTL common section. The write pulse applied to main storage read-out control generates a good-message signal and three shift-out pulses at 10-microsecond intervals if the address-parity-check level is also applied. The shift-out pulses transfer the contents of main storage to the XTL common section, transferring half-word 1 first, then half-words 2 and 3, and finally half-words 4 and 5. The parity bits are read out along with the half-words to which they refer.

The transfer of data through the XTL common section can best be described by following the writing of each drum word into an XTL slot. (See fig. 1-123.) The XTL clock is continually applying 6 bits, representing clock time, to the XTL time amplifiers and a clock-parity signal to drum parity. The site identity generator is connected so that, when pulsed by a write signal, it applies 5 bits representing site identity to the XTL site

identity amplifiers and a site-parity signal to drum parity. Half-word 1, when read out of main storage, is applied to the XTL message amplifiers. Its parity bit is applied to drum parity. Thus, drum parity can generate a parity bit for the first drum word, and the data for the first drum word is applied to the three XTL amplifiers.

The good-message pulse from main storage read-out control is applied to drum write to generate a drum-write pulse, gating the information for the first word onto the drum field. However, the first-word-drum-write pulse must also be generated to gate the clock time and site identity bits into their amplifiers. The first word pulse supplied by drum parity (but ultimately derived from the write pulse applied to the site identity generator) allows generation of the first-word-drum-write pulse. Thus, half-word 1, clock time, site identity, and the parity bit are gated through the XTL amplifiers into the first drum register. The data-available pulse is supplied by the drum write circuit.

The second drum word, when written 10 microseconds later, contains half-words 2 and 3. Half-word 3 is applied directly to the XTL message amplifiers; half-word 2 is split up among the three XTL amplifiers. The second good-message pulse applied to drum write causes

the generation of a drum-write pulse and a data-available pulse. Since no first-word pulse is applied to drum write, the clock time bits are not gated into the time amplifiers. In addition, no site identity bits are generated to be gated into the site identity amplifiers. Therefore, the bits of half-word 2 applied to the XTL time amplifiers and to the XTL site identity amplifiers are transferred to the Drum System with no interference from clock time or site identity bits. The parity bit for the second drum word is generated by drum parity from the parity bits of half-words 2 and 3.

The third drum word, containing half-words 4 and 5, is written 10 microseconds later in a manner identical to the writing of the second drum word. Half-word 4 follows the transfer path used for half-word 2 of the second drum word.

3.2.5 Alarm Conditions

The entire read-out process just described can be delayed for as much as 20.5 milliseconds if the entire XTL field must be searched for an empty slot. If no empty slot is found to receive the message stored in an XTL channel input section before the succeeding message begins shifting into main storage, a read-out alarm is generated in the channel input section. This alarm is generated by the drum demand circuit if a main-storage-

loaded signal (25-counter = 25) is received, followed by a buffer-loaded signal (25-counter = 7) with no intervening drum-demand pulse. The interval between the two 25-core counter signals is approximately 28 milliseconds, sufficient time for a complete search of the XTL field. Thus, a read-out alarm is generated only if the XTL field becomes completely filled or if the drum demand transmission circuits are defective.

The telephone line parity-alarm signals from each XTL channel input section are transmitted to the simplex maintenance console, where they illuminate the parity neons for each incoming word. (See fig. 1-123.) A telephone line absence alarm neon is also installed at the simplex maintenance console to indicate failure of the telephone line data circuit feeding a particular channel input section. The read-out alarms from each of the channel input sections are combined with the clock alarms generated within the XTL common section and illuminate their respective neons at the duplex maintenance console. A clock alarm is generated if the XTL clock fails to synchronize with the real-time pulses supplied by the real-time clock in the Central Computer System.

3.3 INFORMATION FORM

Each XTL message, when written onto the

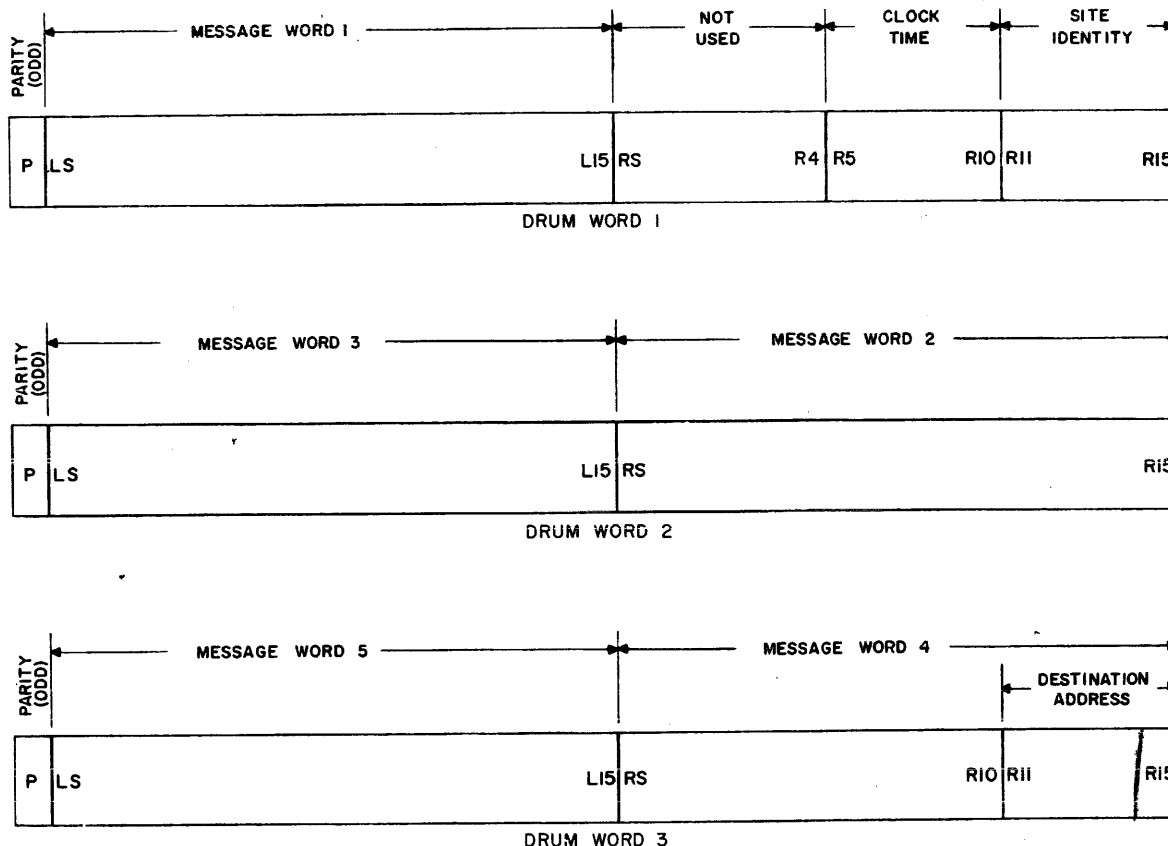


Figure 1-124. Crosstell Message Drum Field Layout

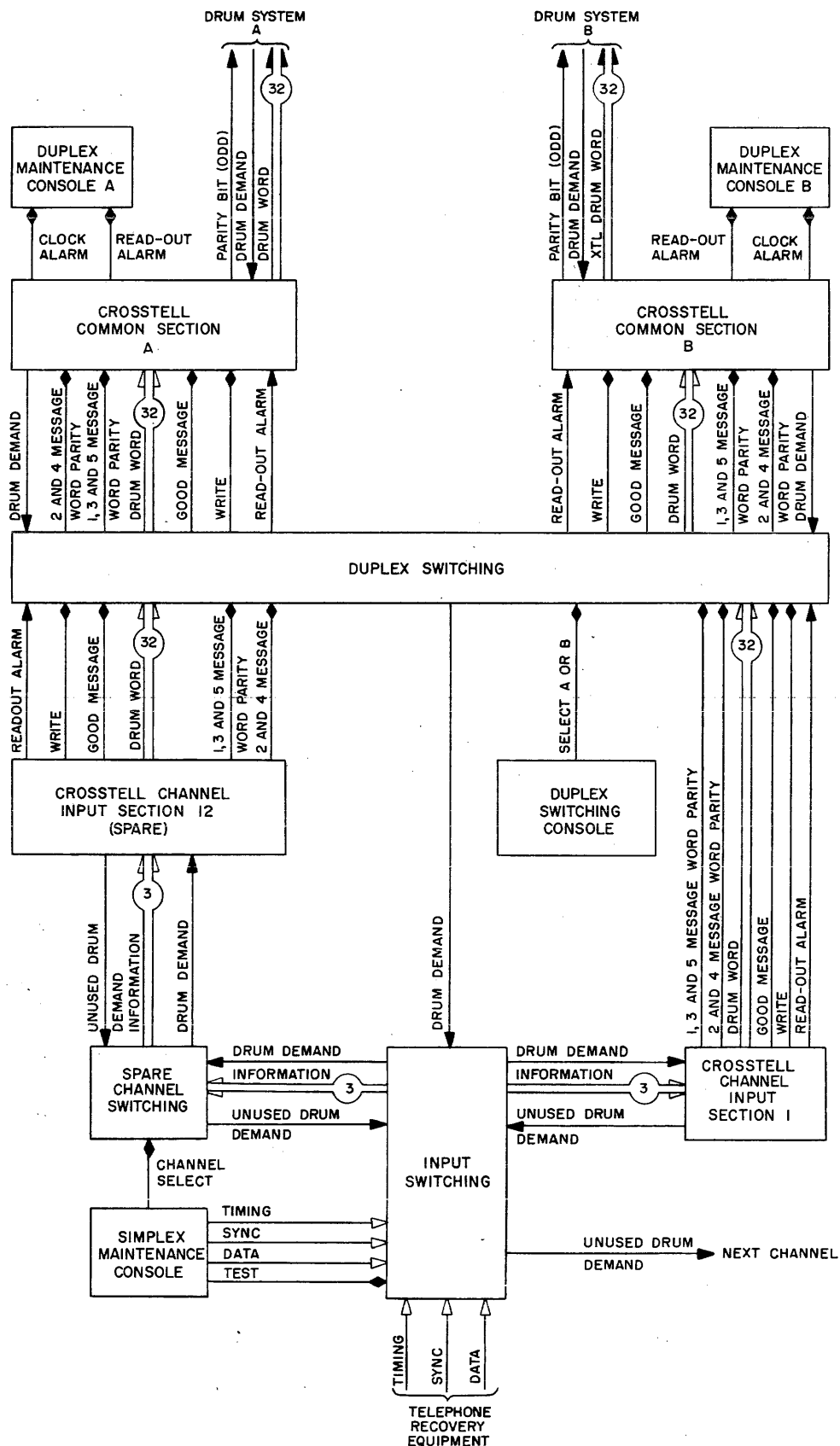


Figure 1-125. Crosstell Element, Simplex and Duplex Switching

XTL field, is contained in three drum registers. (See fig. 1-124.) No specific details on the information layout within the half-words is possible, since these may vary with the type of information being transmitted.

3.4 XTL DUPLEXING

The XTL common section is the only portion of the XTL element that is duplexed. Since the XTL common section receives information from all the XTL channel input sections and transfers this information to the Drum System, failure would completely disable the XTL operation. Therefore, the XTL common section must be duplicated to maintain maximum operating reliability of the XTL element. Spare XTL channel input sections can be switched to replace another malfunctioning XTL channel input section. Thus, duplexing of the XTL channel input sections is not necessary. Simplex and duplex switching operations can best be described by following one simplex and one duplex switching operation. (See fig. 1-125.)

Simplex switching of the XTL channel input sections is accomplished in much the same manner as simplex channel switching in the LRI and GFI elements; however, instead of switching channels on a per-module basis (two channels at a time, as in LRI and GFI), simplex switching of the XTL channel input sections is done on a per-channel basis. This permits the substitution of the spare XTL channel input section for another without interrupting the operation of the channel (which may be satisfactory) adjacent to the replaced channel, as is necessary when simplex-switching on a per-module basis. Substitution of the spare XTL channel input section for an active XTL channel input section which is to undergo test requires two switching operations at the simplex maintenance console. (See fig. 1-125.)

The channel selector switch for the spare XTL channel input section at the simplex maintenance con-

sole must be turned to a number corresponding to the channel to be replaced. The channel-select signal from the simplex maintenance console energizes relays within the spare channel switching device, resulting in completion of the telephone line data and control signal paths of the channel about to be replaced. The test signal from the simplex maintenance console is then applied to the input switching which directs the telephone line data and active control signals to the spare XTL channel input section. At the same time, the paths for the test data from the simplex maintenance console are completed for the replaced XTL channel input section. The drum-demand and the unused drum-demand pulses are routed through the input and spare channel switching to permit normal drum demand interrogation of the XTL channel input section without disturbing the normal interrogation sequence.

Duplex switching, or the designation of active or standby status for the XTL common sections, is accomplished by the select-A or select-B signal from the duplex switching console. Since the outputs of all the XTL channel input sections are identical, they can easily be switched to either XTL common section on active status. (If several XTL channel input-sections are in standby status, their outputs will be directed to the standby XTL common section.) The write signal from each XTL channel input section initiates the generation of site identity in the first drum words. The read-out and clock alarms from XTL common sections A and B are connected to duplex maintenance consoles A and B, respectively. The status of each duplex maintenance console, and hence the status of the XTL common sections, is determined by the duplex switching console. Therefore, the alarm signals from the active XTL common section appear at the active duplex maintenance console, and the alarms from the standby XTL common section are applied to the standby duplex maintenance console.

SECTION 4

MANUAL DATA INPUT ELEMENT

4.1 DRUM ENTRY SECTION

4.1.1 Introduction

The manual data input (MDI) element contains two sections, each of which processes different information. The two sections include the drum entry section and the direct IO buffer entry section. (See fig. 1-126.)

The one apparent difference between the drum entry section and the direct IO buffer entry section is that the former presents information to the Central Computer System via the Drum System whereas the latter presents its data directly. The difference arises from the nature of the information handled by each section.

Time buffering must be performed for the information from both sections, since information is delivered to the two sections at random intervals. This time buffer function is performed for the drum entry section by the manual input field of the Drum System; the direct IO buffer entry section has its own time buffer. The direct IO buffer entry section does not utilize a drum field as a time buffer, since the assembly of each item of information it handles into a drum word would be extremely inefficient. Instead, the direct IO buffer entry section uses a core storage array to assemble information presented to it into a block of words for direct transfer to the Central Computer System. These words are interpreted by the Central Computer System in a way similar to the interpretation of an instruction word; each bit of a core array word may be a separate message whose position within the word indicates its preassigned range of meanings and whose content indicates which meaning of those preassigned to it is being conveyed. If such a 1-bit

message were to be assembled into a drum word for transfer to the Central Computer System by status identification, the bulk of the drum word would be occupied with bits distinguishing the possible meanings of the message from the possible meanings of all other messages; only one bit would contain the designation of the information to be conveyed by the message. By contrast, the information handled by the drum entry section is of three types, each type requiring a number of bits to convey each item. The assembly of one of these items into a drum word is relatively efficient, since the item itself occupies a large portion of the drum word so formed.

4.1.2 Section Description

4.1.2.1 Function

The drum entry section receives data from three computer entry punches (type 026 printing card punches modified to read as well as to punch IBM cards) and from the Display System, forms each item of data into a drum word, and writes each word onto the manual input field by status mode. Since each word contains identity bits indicating the type of information within the word, the Central Computer System can read this data by status identification, thereby preparing separate tables of items of the same type in core memory. The drum entry section establishes the priority of transfer from each source to the manual input field to ensure transfer of only one word at a time and adds the identity bits to each word, thereby indicating the type of information contained within it.

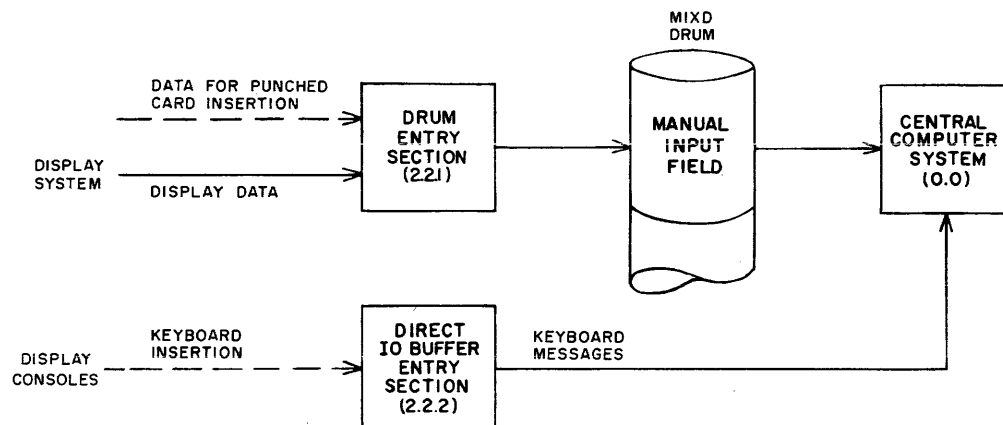


Figure 1-126. Manual Data Input Element

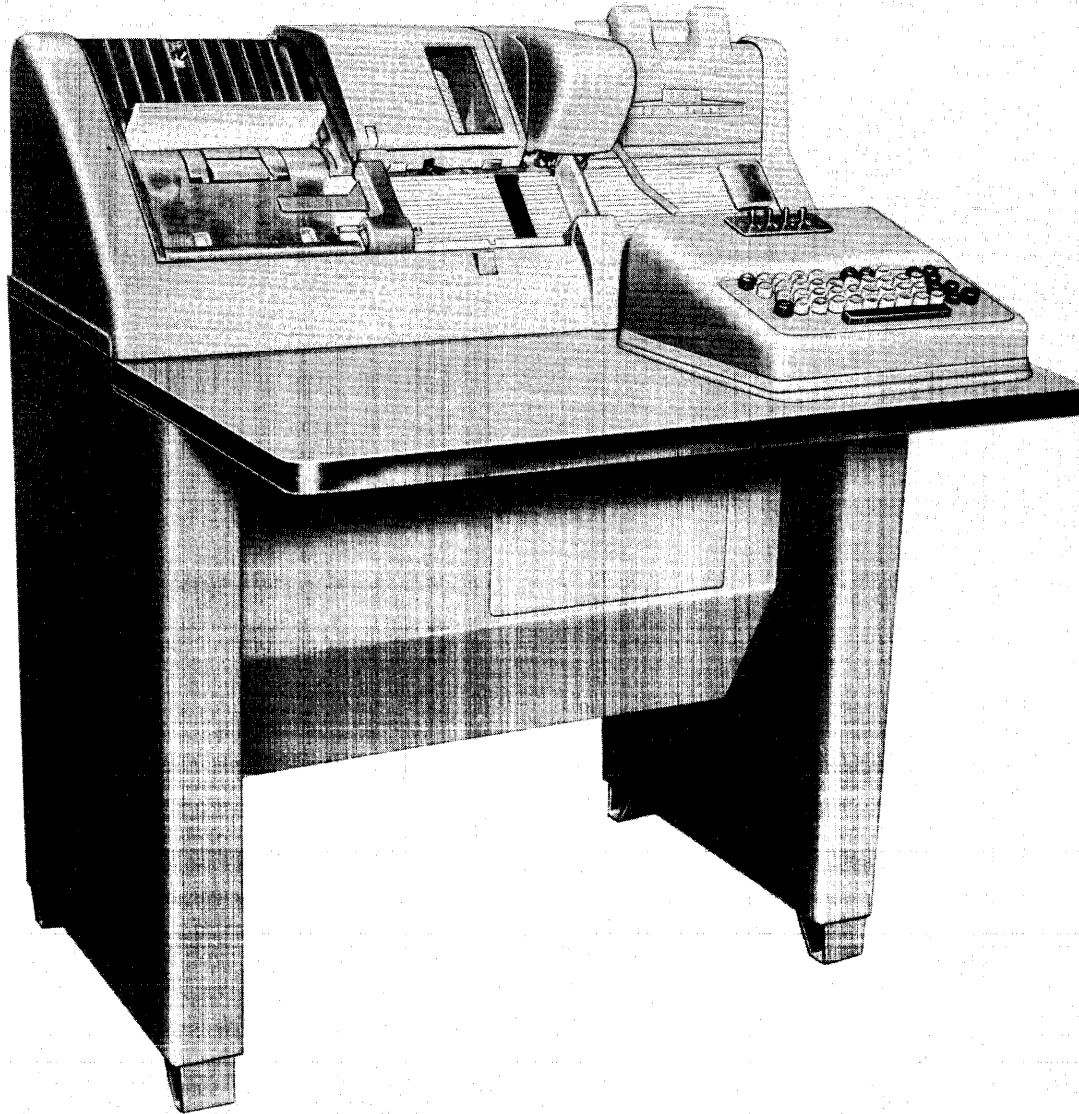


Figure 1-127. Computer Entry Punch, Type 026 Modified

4.1.2.2 Computer Entry Punch Data Sources

The data sources for the drum entry section include computer entry punches. Each computer entry punch is an IBM type 026 printing card punch modified so that it can punch a card, print out the information on the card, and also read the information punched on the card. (See fig. 1-127.) The items of information inserted via computer entry punches may include such things as flight plans of friendly aircraft, early-warning radar and Ground Observer Corps target reports,

weather reports from air bases, antiaircraft operational status and capacity reports, air base operational status reports, and radar site status reports. In effect, any item of information of use to the Central Computer System in an air defense problem which is not introduced in any other way is inserted from punched cards through a computer entry punch. A computer entry punch may read the data from a card punched on another machine as well as from a card just prepared by the computer entry punch.



A computer entry punch can read approximately 13 cards of 80 rows each per minute onto the manual data input (MDI) field or 16 cards of 60 rows each per minute. If the card is punched by the computer entry punch, the information is punched into the card and printed out above the 12 row at the punching station under control of the keyboard. The card moves from right to left past the punching station so that it is punched one column or one character at a time. When the card has passed completely through the punching station, it moves to the reading station, where the data is read from the card one column at a time. Thus, the computer entry punch reads 12 bits at a time from the card; this is in contrast to the card reader (one of the IO units included within the Central Computer System) which reads row by row, or 64 bits at a time. The computer entry punch method of reading is more suited to reading information in Hollerith code, whereas the card reader is more suited to reading information in binary form. A column count is added to each 12-bit character read from a card to allow the Central Computer System to reassemble the characters from one card into a complete message. In addition, a card count (up to 5) is added to allow assembling the information from as many as 5 cards.

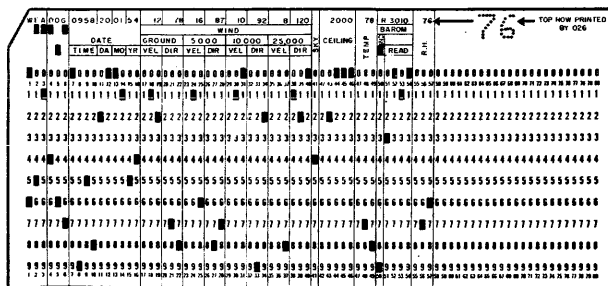


Figure 1-128. MDI Data Card

An example of an MDI data card is shown in figure 1-128. The information punched on the card is in Hollerith code; each column on the card contains one character. Not all the data on a card is read onto the MDI field; columns 11 through 16 of the example shown, containing the date of the report, would probably not be read. This information is placed on the card in order to keep a record of reports received but is of no use to the Central Computer System. Control of the placement of information on the MDI data card and of the reading from that card is exercised by the program drum of the computer entry punch in accordance with the type of data recorded on the card. The punch operator selects the program drum and the program on that

* drum (if an alternate program is on it) which corresponds to the type of information to be presented. The program drum in turn controls skipping, card advance, and automatic duplication, as needed for the card form being used. The card information which is both prepared and inserted by computer entry punch (to save time) is verified after its introduction into the Central Computer System. Less vital information is prepared on a key punch and passed through a verifier before being read by the computer entry punch.

4.1.2.3 Display Data Sources

The drum entry section also receives data from the Display System. Each item of display data presented to the drum entry section identifies a single target moving within the coverage area of the central. If the Display System has information only on the location of a target, that target is identified by presenting its X-Y coordinates (rectangular co-ordinates within the subsector) to the drum entry section. However, if there is more complete information on a target (including its location and velocity), the Display System also has a track number which uniquely identifies the target. In this case, the target is identified by presenting its track number to the drum entry section.

The presentation of either type of item from the Display System is called for either by a light gun or by an area discriminator. These units respond to the light generated by the intensified display of a target on a situation display tube in much the same manner as does the photomultiplier tube mounted over the gap-filler input (GFI) mapper console. (Refer to Sec 2.) The output signal from a light gun or from an area discriminator gates the item of information identifying a particular target out of the Display System. A light gun designates one target at a time, generally in conjunction with a keyboard message sent to the Central Computer System via the direct IO buffer entry section. A light gun is utilized and the associated keyboard message is made up by a display console operator to request some action by the Central Computer System on the target specified by the light gun. An area discriminator reports on all targets of a particular type when the Central Computer System program requests such information. Both the light gun and the area discriminator are discussed in connection with the Display System. (Refer to Ch. 6.)

4.1.3 Section Operation

4.1.3.1 General

The operation of the drum entry section is best described by following the transfer of information from a computer entry punch and then following the transfer of information from the Display System initiated by a light gun and by an area discriminator.



4.1.3.2 Computer Entry Punch Information Transfer

When a computer entry punch reads a column of a punched card, it supplies 24 bits to the MDI register and input switching circuit and an information-ready signal to the MDI selector and computer entry punch interlock. (See fig. 1-129.) The 24 bits of information include 12 bits representing the contents of the column read, 7 bits representing the number of the column on the card, and 5 bits identifying the card within a series of up to 5 cards containing related information. This information is applied to the MDI register and input switching circuit but is not passed through it until the MDI selector and computer entry interlock receives a drum-demand pulse indicating that an empty register on the MDI field is moving into writing position. The drum-demand pulse and the information-ready level from the computer entry punch apply the appropriate selection level to the MDI register and to the MDI information transfer control. This latter circuit generates a data available pulse for the MDI field and a transfer-MDI-information pulse which gates the computer entry punch information through the MDI register and onto the MDI field. The MDI register adds to each word from a computer entry punch, four bits identifying that computer entry punch, and two bits identifying the information within the word as computer entry punch information. Two bits within an MDI drum register containing a computer entry punch word are not used.

Successive columns on a punched card can be read by a computer entry punch at approximately 70-millisecond intervals. Although this is sufficient time to search the MDI field three times, the interval between successive transfers of computer entry punch words onto the MDI field may be longer than 70 milliseconds. This is true because display data, if available, is transferred to the Drum System in preference to the transfer of computer entry punch words. Therefore, the MDI selector and computer entry punch interlock prevents the computer entry punch from reading a succeeding column of a punched card until the information from the preceding column is transferred to the Drum System; the interlock level performs this function. If all computer entry punches have information ready for transfer, the order is computer entry punch 1, computer entry punch 2, and finally computer entry punch 3. Display data has priority over the three computer entry punches.

4.1.3.3 Light Gun Information Transfer *

Provisions exist for the use of up to 64 light guns in designating targets to be referred to the Central Computer System via the drum entry section. A given light gun can designate a target if it detects the display of that target on a situation display tube while a pass-light-gun-signals level is applied to the light gun. (See fig. 1-129.) The pass-light-gun-signals level is up when the following three conditions are satisfied:

- a. The MDI register and input switching circuit is ready to receive a word.
- b. A target which can be detected by a light gun is being displayed by the Display System.
- c. The direct IO buffer entry section is not being read by the Central Computer System.

If all these conditions are satisfied and a light gun is operated to designate a target, a light gun signal from that light gun is applied to the encoder matrix.

The encoder matrix produces 8 bits identifying the originating light gun (with a 4-out-of-8-bit code) and an encoder signal, both applied to the MDI register input control. This circuit applies an MDI-register-set pulse to the MDI selector and computer entry punch interlock, preventing the transfer of computer entry punch information, and a transfer-SD-information pulse to the Display System. The transfer-SD-information pulse, synchronized with the intensification in the Display System of information on a single target, gates the appropriate target identification information out of the Display System. This SD information is applied to the MDI register and input switching circuit along with the console identity bits from the MDI register input control. The target designation and console identity bits are passed to the MDI field when the MDI selector and computer entry punch interlock receives a drum-demand pulse. The drum-demand pulse, coming after the MDI-register-set pulse, generates the MDI-register-selected level, which passes the SD information into the MDI register and causes the MDI information transfer control to generate the transfer-MDI-information pulse and the data-available pulse.

The operation of the Display System makes it impossible to supply items of display data more often than once every 60 microseconds. However, the requirements of manual operation and the number of light guns reduce the average rate of transfer over a period of time to a maximum of one item per 40 milliseconds. Since there is information on only one target in the Display System at any one time, it is impossible for two light guns to designate different targets for simultaneous transfer; the minimum interval between transfers of different targets is the 60-microsecond interval just mentioned. However, it is remotely possible that two light guns may request transfer of the same target at the same time. If this should happen, one word is placed on the MDI field containing target identification information and console identity bits; the console identity bits will contain more than four 1's, thus rendering the console identity ambiguous and causing rejection of the word by the Central Computer System. A light gun may initiate transfer of only one target each time it is operated. The uses of light guns are described more fully in connection with the Display System. (Refer to Ch. 6.)

4.1.3.4 Area Discriminator Information Transfer

An area discriminator differs from a light gun in several ways, although the basic principle of their oper-

ation is identical. An area discriminator begins transfer operations under control of the Central Computer System program; a light gun is manually operated. Once its operation has started, an area discriminator may initiate transfer of a number of targets; a light gun must be manually positioned and operated each time a target is to be transferred. Finally, an area discriminator may initiate transfers during the reading of the direct IO buffer entry section whereas a light gun may not; this last difference is accounted for by the fact that a light gun must also write a bit in the direct IO buffer entry section; this is not required of an area discriminator.

Area discriminator output signals are delivered to the light gun and area discriminator gating controls whenever the photomultiplier tube of the area discriminator detects a target displayed to it on a situation display tube. (See fig. 1-129.) However, no signals are applied to the encoder matrix unless the Central Computer System program requests reports from an area discriminator. The *Operate* (area discriminator) instruction, *PER* (20)₈, initiates the transfer of reports from the area discriminator. Upon execution of this instruction, the area discriminator is synchronized with Display System operation by the light gun and area discriminator gating controls and then initiates transfer of those targets detected by it over a period of 2.5 seconds. At the end of this 2.5-second period, no more transfers are initiated by the area discriminator until it is again selected by the Central Computer System.

Each time a pulse is applied on the area discriminator (AD) output line during the period of its selection, an AD signal is applied to the encoder matrix if the MDI register and input switching circuit can receive a word and if the target detected by the area discriminator is one on which the Display System allows light gun action. Once an AD signal is applied to the encoder matrix, the transfer operation proceeds exactly as described for light gun information transfer. The minimum interval between successive transfers initiated by the area discriminator within its 2.5-second operation period is 60 microseconds.

4.1.4 Information Forms

Information from the drum entry section is placed on the MDI field of the MIXD drum by status, with one item per drum register. The layout of information within each word is determined by the type of information. A computer entry punch word is shown in figure 1-130, part A. A word of this type contains the following information:

- a. Card information, LS through L11, containing the information punched in one column of a card in the order 12 row through 9 row
- b. Computer entry punch identity, L12 through L15, identifying the computer entry punch which has read the information supplied (computer entry punch 7 identified by a 1 in bit L15, computer

entry punch 2 by a 1 in bit L14, computer entry punch 3 by a 1 in bit L13)

- c. Card count, R2 through R6, indicating which card in a sequence of up to five cards is being read (uses a 1-out-of-5-bit code)
- d. Column count, R7 through R13, identifying the column whose contents are reproduced in positions LS through L11
- e. Information type, R14 through R15, indicating computer entry punch information by its contents of 00.

A display data word containing a track number is shown in figure 1-130, part B. A word of this type contains the following information:

- a. Track number, LS through R5, identifying a target for which a track has been developed
- b. Console identity, R6 through R13, identifying the originating light gun or area discriminator (uses a 4-out-of-8-bit code)
- c. Information type, R14 through R15, indicating track data (with a track number) by its contents of 01.

A display data word containing X and Y coordinates is shown in figure 1-130, part C. A word of this type contains the following information:

- a. X co-ordinate, LS through L10, indicating the east-west displacement of the target from the coverage area reference point
- b. Y co-ordinate, L11 through R5, indicating the north-south displacement of the target from the coverage area reference point
- c. Console identity, R6 through R13, identifying the originating light gun or area discriminator (using a 4-out-of-8-bit code)
- d. Information type, R14 through R15, indicating radar data (X and Y co-ordinates) by a 1 in the content of R15 (a correlated report being one on which a track has been initiated).

The significance of track data versus radar data messages is discussed in the description of the Display System. (Refer to Ch. 6.)

4.1.5 Duplexing

The bulk of the drum entry section is duplex, with duplicate control and transfer circuits in computer A and in computer B. However, the data sources are simplex; these include the computer entry punches, the area discriminators, and the light guns. The computer entry punches supply information to the drum entry section whose status (active or standby) matches the status of the computer entry punch. The status of a given computer entry punch is controlled by a unit status switch mounted on the punch. Similarly, the status of the area discriminators (of which there are two, one usually active and the other a spare) is controlled by unit status switches on the area discriminator consoles. Light guns

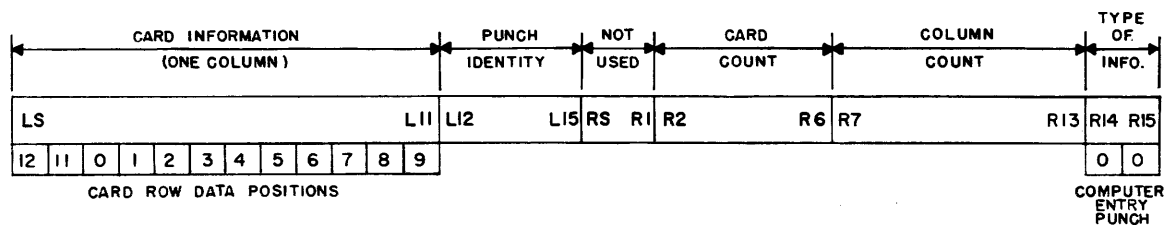
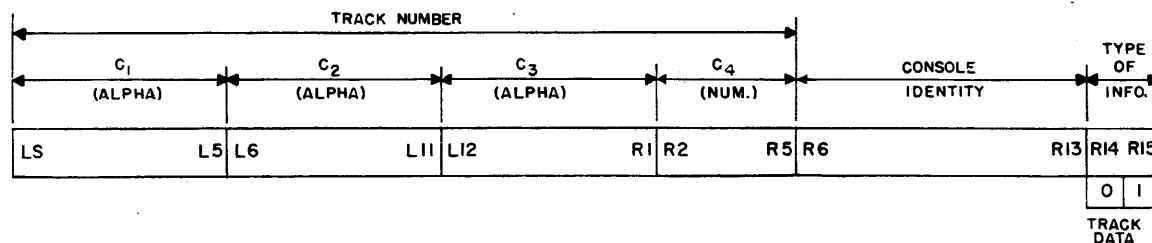
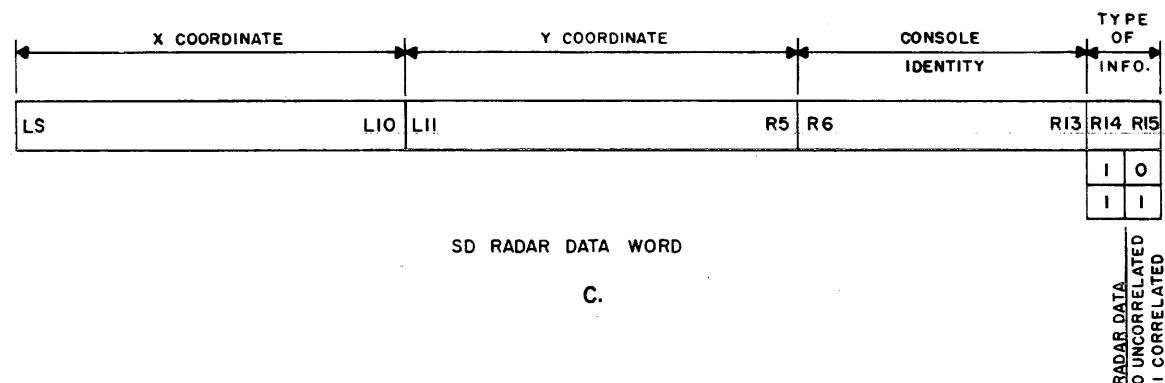
COMPUTER ENTRY PUNCH WORD
A.SD TRACK DATA WORD
B.SD RADAR DATA WORD
C.

Figure 1-130. MDI Message Drum Field Layouts

have the same status as the display consoles at which they are used, determined by the unit status switches at those consoles. Data source units in standby status receive signals from and supply outputs to the standby computer and can therefore be used in maintenance operations.

4.2 DIRECT IO BUFFER ENTRY SECTION

4.2.1 Section Description

4.2.1.1 Function

The direct IO buffer entry section provides display console operators with the means of inserting into the Central Computer System decisions or requests which direct the operation of that system in processing an air defense problem. These decisions or requests are decided upon by display console operators from the information displayed to them at their consoles and inserted manually by the operators. A message so

inserted may contain target designations (other than location or track number), target descriptions, tactical weapons assignments, requests for information, or directions for further processing. The type of message which may originate from a particular console depends upon the assignment of that console to the performance of a particular function in directing the air defense of an area.

Direct IO buffer entry section messages are composed by display console operators by depressing push-buttons on MDI input data selection control panels (keyboards), usually attached to display consoles. (See fig. 1-131.) Each pushbutton on an MDI keyboard, when depressed, sets one or more tape cores to the 1 state. These tape cores are contained within the direct IO buffer entry section in a core matrix of 128 by 32 cores. Thus, the section assembles all the messages made up at different keyboards into a block of 128 words for reading by the Central Computer System.

4.2.1.2 Keyboard Messages

The function of a given console in the air defense of an area places a limit on the types of messages which may be inserted into the core matrix of the direct IO buffer entry section from that console. The function of the console also limits the variety of messages within a single type that can originate at that console. For example, a particular console might need to request from the Central Computer System information which is not normally displayed at that console. Therefore, one type of

message which can originate from that console is an information request message. If the console operator can utilize only three kinds of information not normally displayed, four different information request messages can be originated at that console, one message indicating that no information is requested and another message for each kind of information which may be needed.

To simplify insertion of the request, the console keyboard contains one or more panels on which are mounted several pushbuttons, one for each kind of in-

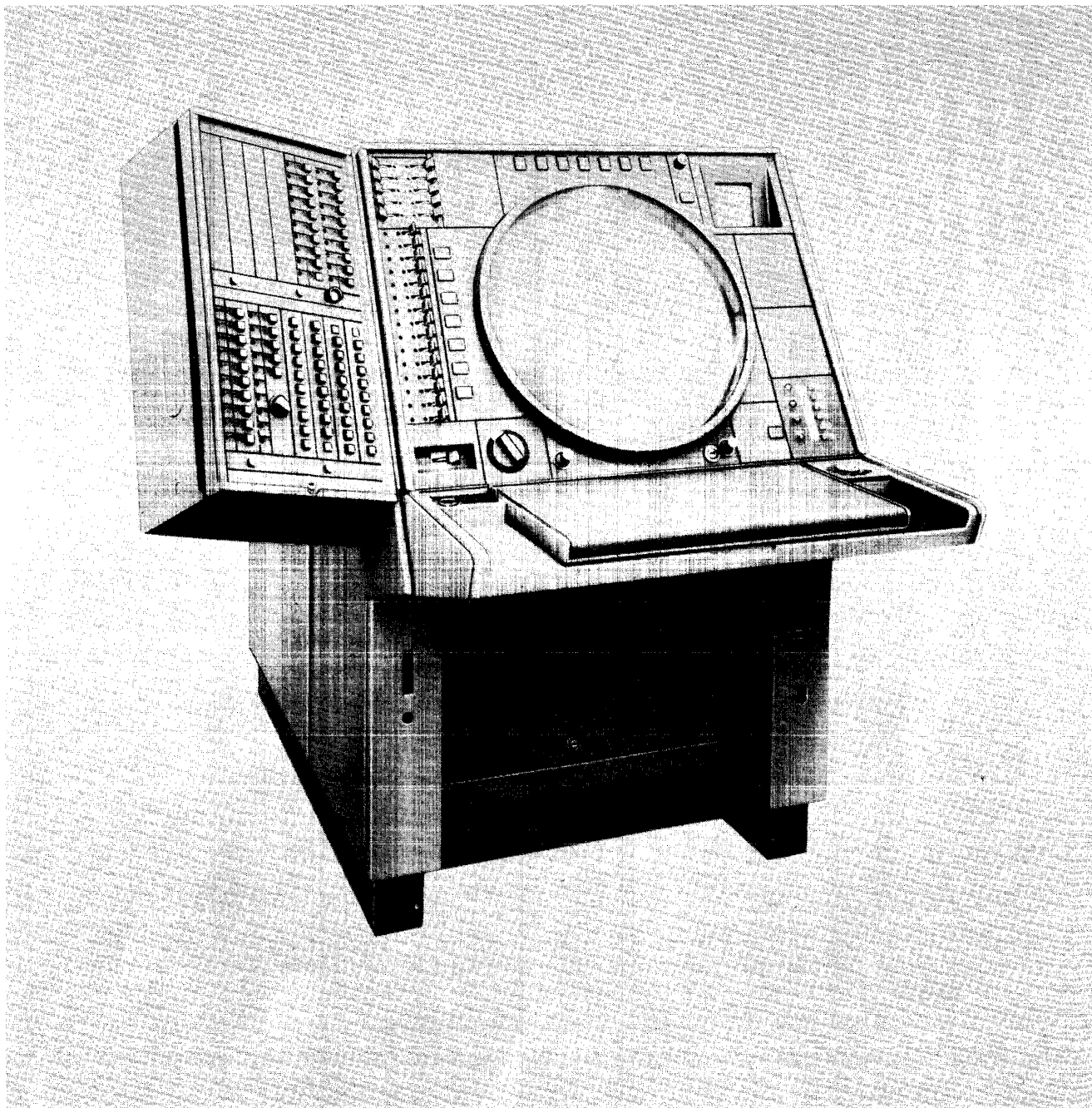


Figure 1-131. MDI Keyboard Mounted on Situation Display Console

formation request message. These pushbuttons control the setting of several cores in the core matrix; the position of the cores in the matrix indicates, by prearrangement, that these bits contain information requests from the console being discussed. The contents of these bits indicate the kind of information requested. The number of cores which must be controlled by these pushbuttons depends upon which of two alternatives is chosen: either only one kind of information can be requested at any one time or all three kinds may be requested in any combination. If the first alternative is chosen, the pushbuttons must control the setting of two cores whose contents may represent one of the numbers from 0 through 3 in binary form. The wiring of the pushbuttons to the cores is such that depressing a given pushbutton sets the two cores to the binary representation of the number indicated by that pushbutton. If, on the other hand, the pushbuttons must be capable of requesting all three kinds of information, then each pushbutton must set a separate core.

The exact makeup of a particular MDI keyboard and the correspondence of pushbuttons to cores in the core matrix depend upon the functions of the console to which the keyboard is attached. A portion of an MDI keyboard is shown in figure 1-132. Each vertical line of pushbuttons usually originates one type of message. The 10-pushbutton panels may control the setting of only 4 cores each; each pushbutton might insert a decimal digit in binary form in the 4 cores. In this case, the pushbutton which inserts a decimal zero simply releases the other nine pushbuttons. The rotary switch on a keyboard panel is used for indicating the approximate heading and speed of a target. The rotor switch indicates azimuth with a precision of 45 degrees by setting three cores; the pushbuttons can set three other cores to an octonary representation of the approximate speed of the target. A panel of ten warning lights may also be mounted on an MDI keyboard.

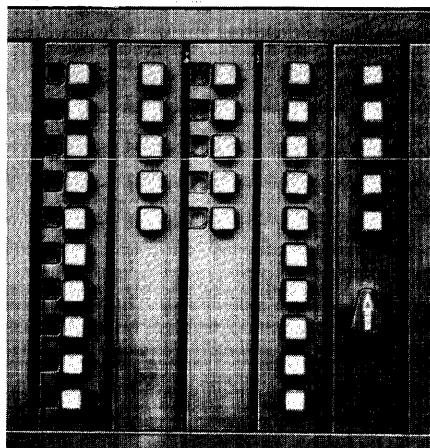


Figure 1-132. MDI Keyboard

A message pushbutton on an MDI keyboard, once depressed, continually sets the message core or cores it controls to the 1 state. Those message pushbuttons which are not interlocked with the other pushbuttons on the same panel can be released by pressing the RELEASE pushbutton on the keyboard, thus allowing the message cores controlled from that keyboard to be cleared when they are read by the Central Computer System. If the pushbuttons are not released after the message is read once, no difficulties result, since the Central Computer System will not act on the message unless another core in the matrix is set to 1. This core, called an action core, is set by the ACTION pushbutton, a snap-action switch on the MDI keyboard. The operator sets up a message on the keyboard and depresses the ACTION pushbutton to indicate that the message is complete. When the Central Computer System reads the core matrix and finds a 1 in the action core assigned to that keyboard, the message from that keyboard is acted upon. However, the reading of the core matrix clears the action core, which remains cleared until the ACTION pushbutton is pressed again. Thus, the Central Computer System is prevented from acting upon an incomplete message, or a message already acted on, by the absence of a 1 in the action core for that keyboard.

There is one condition under which the Central Computer System acts upon a keyboard message whose action core does not contain a 1. Those messages which refer to a specific target require the use of a light gun to designate the target to the Central Computer System via the drum entry section. Since this is the case, certain messages are complete only when a light gun transfer has been completed. Accordingly, a console equipped with a light gun is assigned a core which is set by the light gun signal that also initiates the transfer of a target designation through the drum entry section. Thus, if the Central Computer System has read a 1 out of either the light gun core or the action core of a particular keyboard, the message set up by that keyboard is acted upon. In effect, the operation of the light gun at a console is equivalent to operating the ACTION pushbutton.

4.2.2 Section Operation

4.2.2.1 Message Insertion

Manual data input (MDI) messages are made up and indicated as complete by the ACTION pushbutton or light gun operation. (See fig. 1-133.) The correspondence of ACTION pushbuttons to action cores and of keyboard switches to the other cores in the core matrix is established by jumper wiring on the interconnection unit. The signals from the light guns are permanently wired so that the signal from a particular light gun always sets the same light gun core in the matrix.

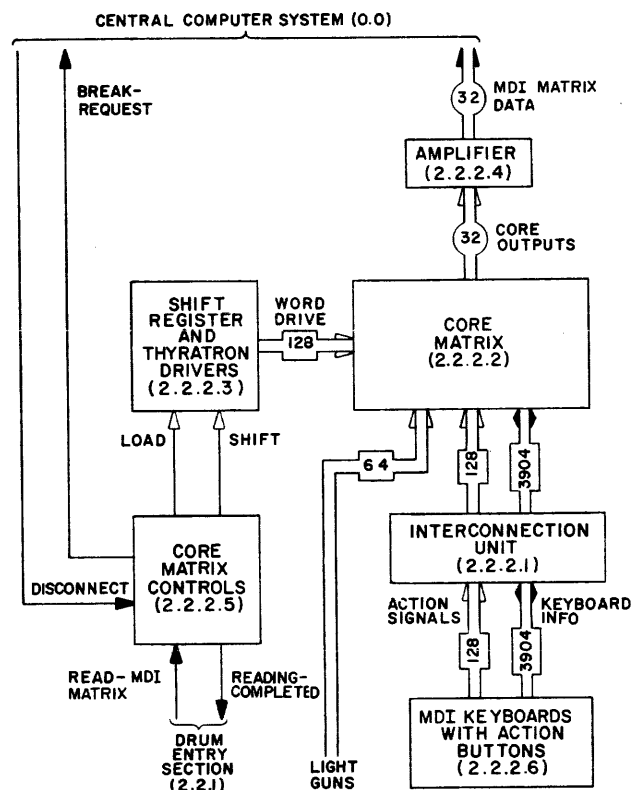


Figure 1-133. MDI Direct IO Buffer Entry Section

4.2.2.2 Core Matrix

The core matrix consists of 128 rows of 32 cores each. The first two rows contain all light gun cores with one core per light gun. The next 4 rows contain all action cores, allowing the use of 128 ACTION pushbuttons and, therefore, 128 keyboards that originate independent messages. The following 6 rows of cores indicate to the Central Computer System which simplex equipment (i.e., channel sections in LRI and GFI) is operating with computer A or computer B. The remaining 16 rows of cores contain the messages from the keyboards. There need not be any relationship between a row of information cores and a particular keyboard. In fact, it is more likely that the interconnect unit will be wired to place those messages of the same type (e.g., all information request messages) into the same row or rows of the core matrix rather than to place all messages from one keyboard into the same row or rows of the core matrix. This presorting of messages by type within the core matrix could simplify the Central Computer System program used to interpret keyboard messages. The source of each message would then be indicated by its position within the block of messages of the same type rather than by its overall position within the block of words read from the core matrix.

4.2.2.3 Transfer to Central Computer System

The readout of the manual input core matrix to the Central Computer System is initiated by an IO program that selects the core matrix. Execution of the *Select* (manual data input matrix) instruction, *SEL* (06)₈, followed by an *RDS* 200₈ instruction, applies a read-manual-data-input-matrix pulse to the drum entry section. (See fig. 1-19.) This pulse is forwarded to the core matrix controls circuit in the direct IO buffer entry section. (See fig. 1-133.) The core matrix controls circuit writes a 1 in the first core of the 128-core shift register within the shift register and thyatron drivers. Thereafter, the core matrix controls circuit applies a shift pulse to the shift register and thyatron drivers and a break-request pulse to the Central Computer System a 20-microsecond intervals. Each time the 128-core shift register is shifted, an entire row of the core matrix is read out to the amplifier and thence to the IO buffer register. The rows are read out in this order: light gun cores, action cores, and information cores, until the number of words specified by *RDS* instruction have been read. The Central Computer System then ends readout by sending a disconnect pulse to the core matrix controls circuit.

In normal operation, the Central Computer System program can associate a keyboard message completed by the setting of a light gun core and the target designation transferred by that light gun action. This can be done because the position of the light gun core bit within the words containing light gun bits identifies the light gun; the program must then find the MDI drum word whose console identity code identifies the same light gun. If the light gun core is not set, the keyboard message is not acted upon by the Central Computer System. Therefore, if a light-gun-initiated transfer to the MDI field were allowed during the reading of the manual input core matrix, it would be possible for the Central Computer System to obtain the MDI drum word but to ignore the keyboard message directing the action to be taken on that target. The console operator would receive an erroneous indication that the entire message (target designation and keyboard information) was accepted. To prevent this possibility, light gun actions are inhibited during the reading of the manual input matrix.

It is possible for the Central Computer System program to read fewer than 128 words from the manual input matrix. If the program is so written, a disconnect pulse is supplied by the Central Computer System after it receives the prescribed number of words. Since reading always begins with the first row of the core matrix, the effect of reading fewer than 128 words is the loss of whatever information is stored in the rows not read. If operational requirements indicate that these rows are

*delay between readings of MDI matrix
at least 1/2 sec. minimum 1/2 sec.*

not needed for information storage, the program ignores these unused rows. No parity bit is included in the words read from the MDI matrix.

4.2.3 Duplexing

The direct IO buffer entry section is almost entirely duplex, with duplicate circuits in both computer A and computer B. However, portions of this section are simplex; in particular, the lines which set action and light gun cores in the MDI core matrix are switched to set only those cores in the matrix of the equipment whose status matches the status of the originating unit (either keyboard or light gun). Thus, although the keyboard message pushbuttons set cores in both matrices when

depressed, each message is indicated as complete only in the matrix of the computer whose status is the same as the status of the keyboards originating the messages. (See fig. 1-134.)

The unit status lines from the various simplex units in AN/FSQ-7 Combat Direction Central are applied to cores in both the active and standby matrices to indicate the status of these units to the two computers. A unit in active status sets its core in both matrices to the 1 state. No difficulties result from this procedure since the standby computer recognizes the existence of a 1 in a unit status bit position as an indication that that unit is active and not operating with the standby computer.

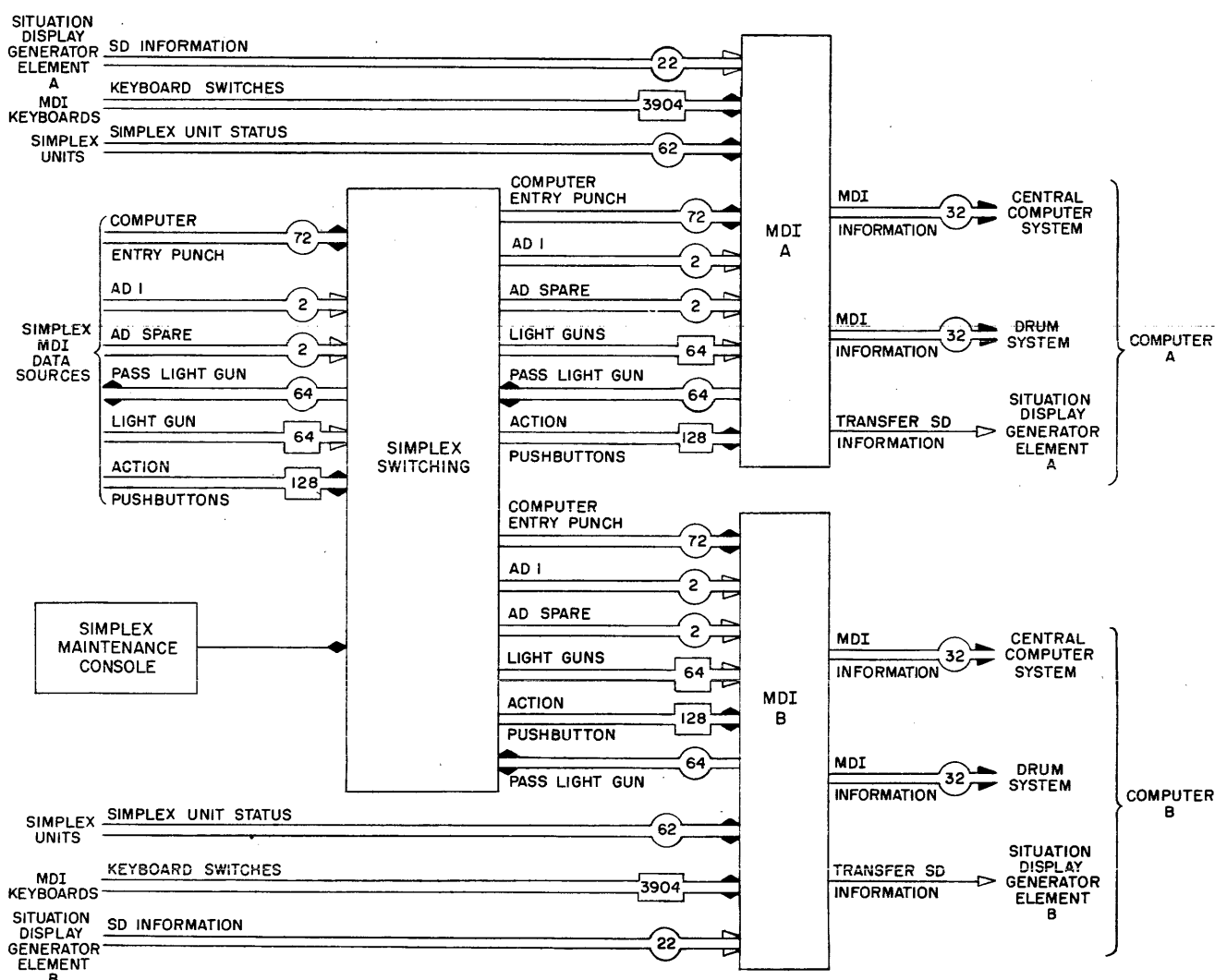


Figure 1-134. MDI Element, Simplex and Duplex Switching

CHAPTER 6

DISPLAY SYSTEM

SECTION 1

SYSTEM DESCRIPTION

1.1 SYSTEM FUNCTION

The Display System of AN/FSQ-7 Combat Direction Central is the medium through which all data collected by the Input System and processed by the Central Computer System must be presented to personnel directing the defense of the subsector against air attack. The data which must be collected includes the following: the locations of unidentified, friendly, and hostile aircraft within the area; the status of defensive weapons that can be employed against hostile aircraft; and information on conditions affecting the usefulness of each weapon such as visibility or other weather conditions.

By presenting this information in digested form to the personnel directing air defense, the Display System makes it possible for the operators to make informed decisions on the actions to be taken in each case. Their decisions are indicated to the Central Computer System for implementation via the manual data input element of the Input System. (Refer to Chapter 5, Section 4.)

The information to be presented by the Display System can be divided into two types: one type, presenting a geographical picture of air movements within the subsector, must be capable of changing rapidly to reflect accurately the air movements themselves; the other type, presenting such relatively static information as weather conditions or weapons status, need not be changed as often as the first type. The two types of information are known as situation display information and digital display information, respectively. Situation display information provides a complete picture of air movements in a plan-position-indicator (PPI) form. It is identical in function to the plotting-board display of a manual air defense filter center. Digital display information provides statistical data in a tabular form. It is identical in function to the tote-board display of a manual air defense filter center.

1.2 SYSTEM LOGIC

1.2.1 Overall

The nature of the information handled by the Display System dictates its division into two subsystems, situation display and digital display. Both subsystems receive processed information from the Central Computer System via the Drum System. This information is

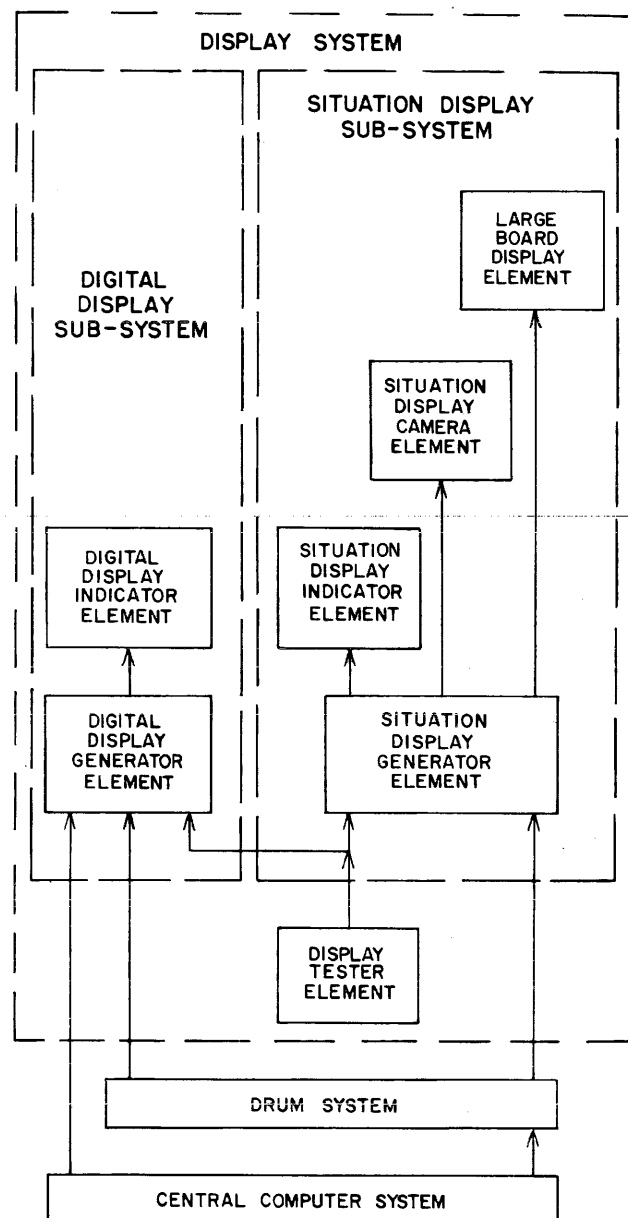


Figure 1–135. Display System, Logical Divisions

supplied to the generator element in each subsystem. Each generator element in turn converts the information presented to it into the form required by the indicator element of that subsystem. (See fig. 1-135.)

The situation display subsystem includes the situation display generator element (SDGE) and the situation display indicator element (SDIE). The situation display generator is a centrally located unit which converts processed situation display information into the form required by the situation display indicator element. Each situation display indicator element is housed in a display console. (See fig. 1-136.)

The digital display subsystem includes the digital display generator element (DDGE) and the digital display

indicator element (DDIE). The digital display generator element is a centrally located unit which converts digital display information into the form required by the digital display indicator element. The digital display indicator element is also housed with the situation display indicator section in a display console. However, some digital display indicator sections are separately housed in what are known as auxiliary consoles and some in the command post desk.

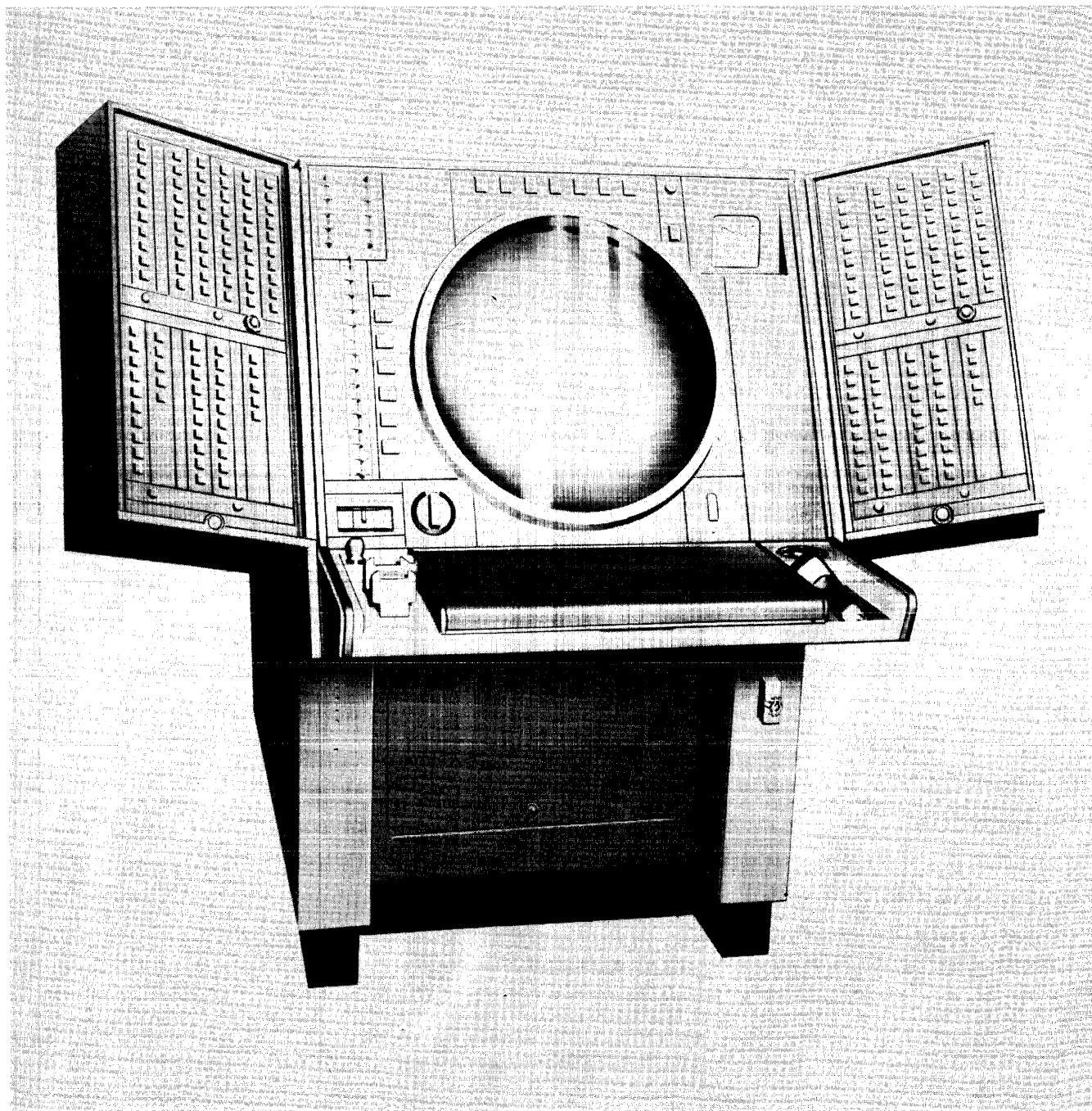


Figure 1-136. Display Console

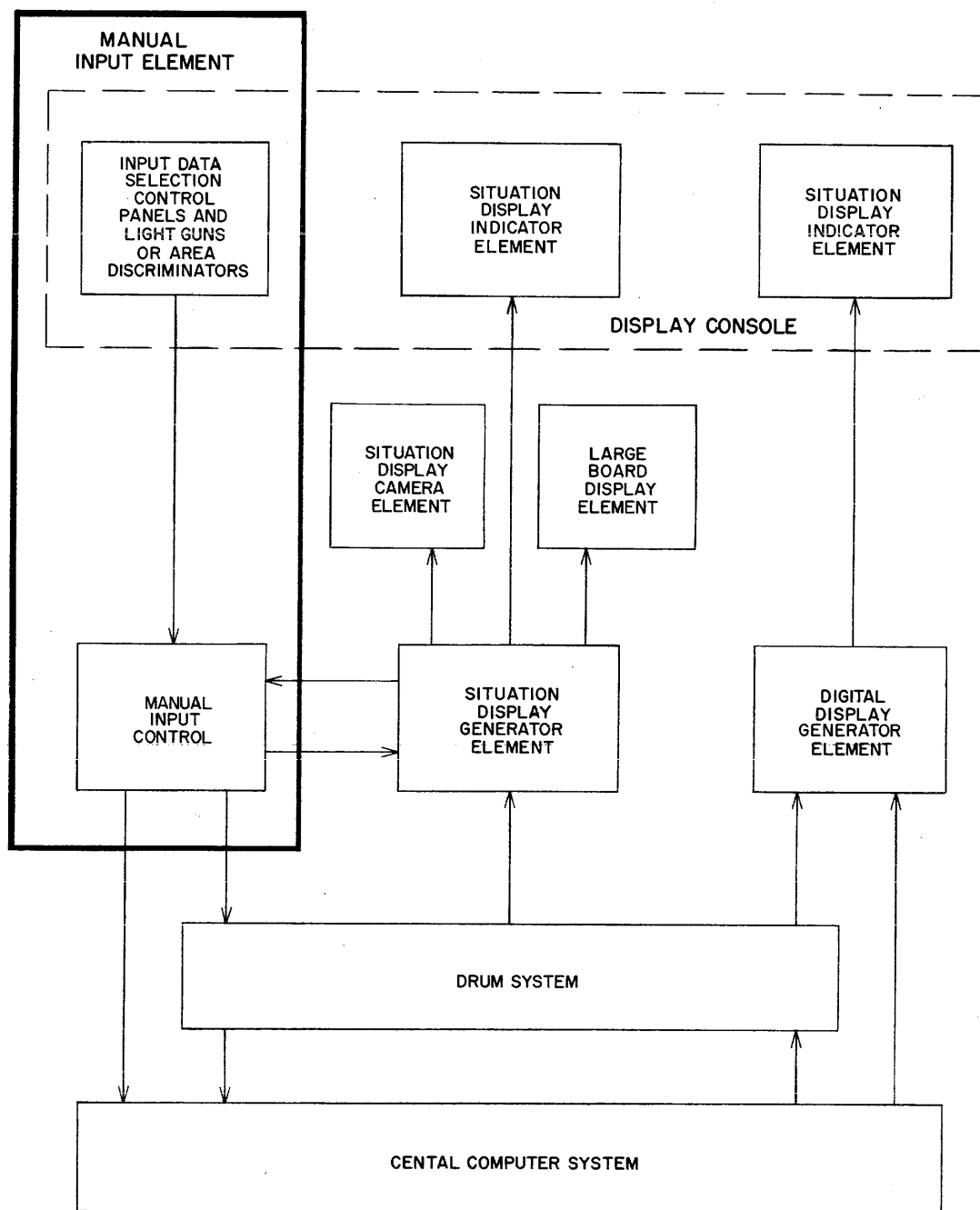


Figure 1-137. Display System Relation to Manual Input Element

Most display consoles house an SDIS and a DDIS as well as a panel of pushbuttons for inserting information via the manual data input element into the Central Computer System. (See fig. 1-137.) Some 64 consoles also have light guns for designating specific targets through the manual data input element. Two special consoles are included for specific purposes. One console provides the situation display which is photographed to provide a large board display by projection of the photograph. Another console is equipped with a camera which

makes permanent records of air defense situations for later analysis or for use in training of console operators.

1.2.2 Duplexing

Although the number of display consoles within the Display System makes duplexing unnecessary, the SDGE and the DDGE are duplex, since the failure of either one would render inoperative all corresponding indicator sections. (Refer to Chapter 1, 3.3.) To provide for the possibility that a given console will fail, each

console (containing an SDIS, a DDIS, or both) is equipped with a unit status switch which determines whether it shall receive signals from the active or the standby generator elements. If a console becomes defective during an air defense problem, it is switched to standby, and another console replaces it in active status. Figure 1-138 shows the general plan of the switching arrangements between the duplex and simplex units of the Display System.

The switching arrangement is such that once a console operator indicates whether a given console is to receive active or standby data by setting the unit status switch at that console, the proper data is fed to the console, regardless of which set of generator elements is ACTIVE. It should be noted that a console may be operated in standby status for training purposes, to allow for maintenance of that console, or simply if it is a spare console not in use at a given time.

1.3 DISPLAY TUBES

1.3.1 General

The operation of the Display System is largely determined by the types of cathode-ray tubes on which the displays are generated. Therefore, it is possible to obtain a general understanding of the Display System and its operation from a description of those display tubes. Thus, this section concludes with a description of the situation display tube and the digital display tube. Sections 2 and 3 describe the operation of the situation display and digital display subsystems, starting with descriptions of the forms in which information is presented

to those two subsystems. Section 4 discusses the flow of information through the entire Display System and its relation to the manual input element of the Input System. However, the differences between the situation and the digital display subsystems can be most easily understood by examining the display tubes used in each subsystem.

1.3.2 Situation Display Tube

The 19-inch situation display tube used in the situation display subsystem must be capable of showing a great deal of information about every target. A complete message may include: a point designating the target grouping of information about a target, known as a location; a vector designating the target direction and speed; a group of characters (letters, numbers, or pictorial symbols) designating target identification, height, raid and track numbers; and any other information which the Central Computer System has compiled. (See fig. 1-139.) Further, the display must allow for rapid, regular changes to show the changed position of targets under surveillance.

Two of the needs which the 19-inch situation display tube serves require design features not present in a standard cathode-ray tube; these are the forming of characters and the positioning of these characters within a message which must itself be positioned on the tube face. Characters are generated in the situation display tube by a stencilling process: i.e., the writing beam is passed through a selected aperture in a character-forming matrix before it strikes the tube face. (See fig. 1-140.) Just as a stencil allows paint to be deposited

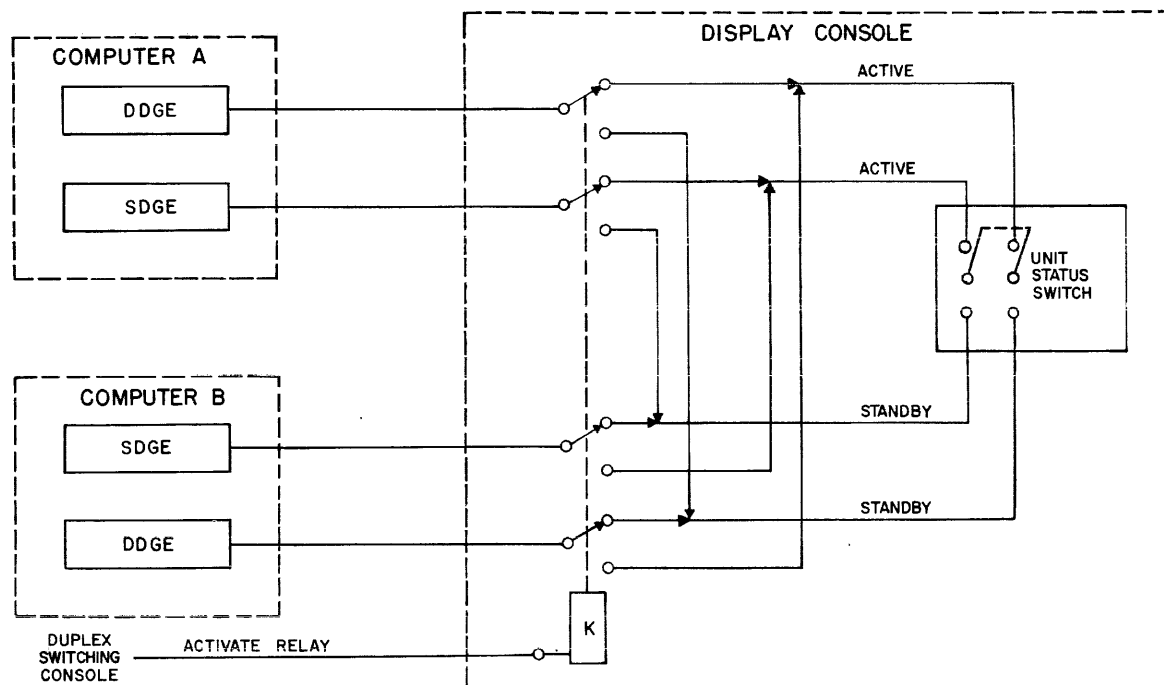


Figure 1-138. Situation Display Console Duplexing

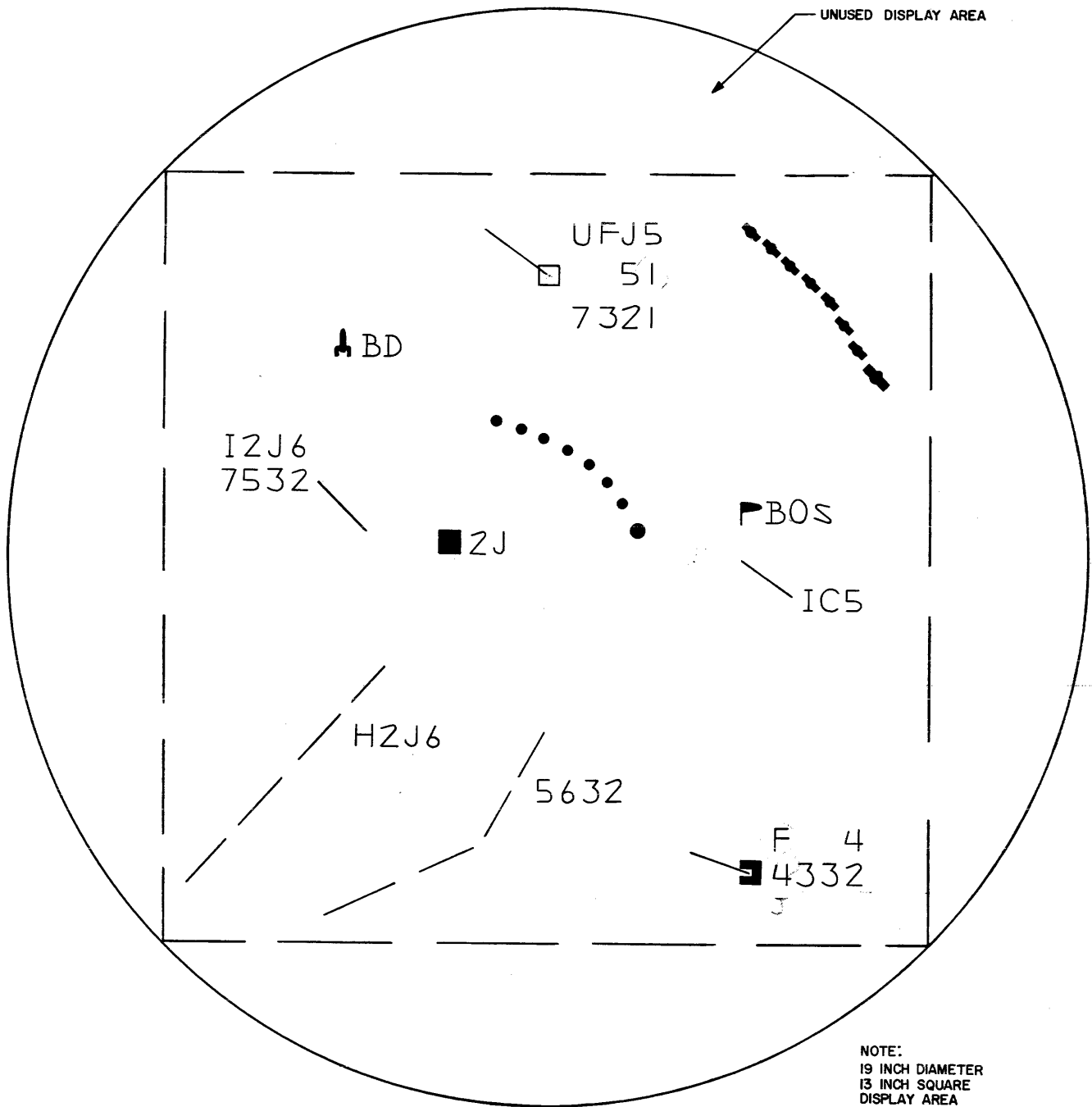


Figure 1-139. Typical Situation Display

only on the area not masked by the stencil, the aperture imparts the shape of the desired character to the writing beam. The character selection process uses a set of electrostatic deflection plates to direct the writing beam through the selected aperture. The positioning of the characters within a message is handled by using another set of electrostatic deflection plates. An electromagnetic deflection yoke then positions the entire message on the tube face.

The design features of the 19-inch situation display tube are shown in figure 1-141. It is a cathode-ray tube having a 19-inch face coated with a medium-persistence phosphor (P14). The phosphor emits a blue flash when illuminated by the writing beam, followed by a yellow afterglow persisting long enough for human perception but not long enough to obscure subsequent displays. The tube contains a set of character selection plates (electro-static deflection) and an eight-by-eight



Figure 1-140. Character Matrix, Situation Display Tube

character-forming matrix for character generation. The beam is defocused during character generation, making the beam large enough in cross section to cover and be shaped by the character aperture, by applying a defocus signal to the focusing anode of the tube.

The deflection introduced for character selection is corrected and the character positioned within a message display by a second set of plates, the character positioning plates (electrostatic deflection). The target vectors are also generated by the character positioning plates; voltages proportional to the horizontal and vertical components of the target vector are applied to these plates to sweep the focused electron beam in the vector direction over a distance proportional to the vector magnitude. The maximum length of the vector (and the maximum displacement of a message character from the vector origin) is limited to less than two inches. An electromagnetic deflection yoke determines the horizon-

tal and vertical positioning of a message display on the face of the tube.

The characteristics described above indicate that the following signals are required for each situation display message:

- Character selection voltages
- Character positioning and vector generating voltages
- Message positioning (X and Y co-ordinate) voltages
- Defocus gate during character generation
- Intensification gates (writing beam normally cut off; intensification gates turn on the writing beam).

1.3.3 Digital Display Tube

The 5-inch digital display tube used in the digital display indicator element must be capable of continuously displaying tabulated information written on it until new information is obtained or until other information is requested by an operator. The tabulated information may describe weather conditions, number of interceptors available, or other relatively slow-changing information. Like the situation display tube, the 5-inch digital display tube displays characters (letters, numbers, or pictorial symbols) to present this information in an easily understood form. (See fig. 1-142.) However, the persistency needed for digital display requires components not present in the situation display tube.

Characters are formed for the situation display and the digital display in exactly the same way. A set of character selection plates (electrostatic deflection) and a character-forming matrix are required. (See fig. 1-143.) The continuous display is handled by an electrostatic storage mesh and a flood gun. After the char-

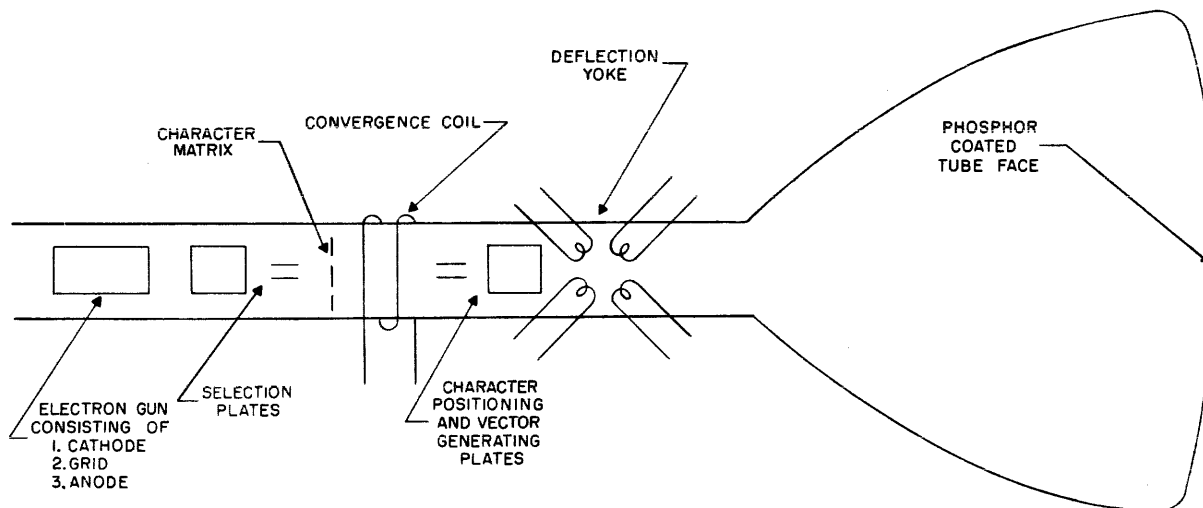


Figure 1-141. Situation Display Tube, Simplified Diagram

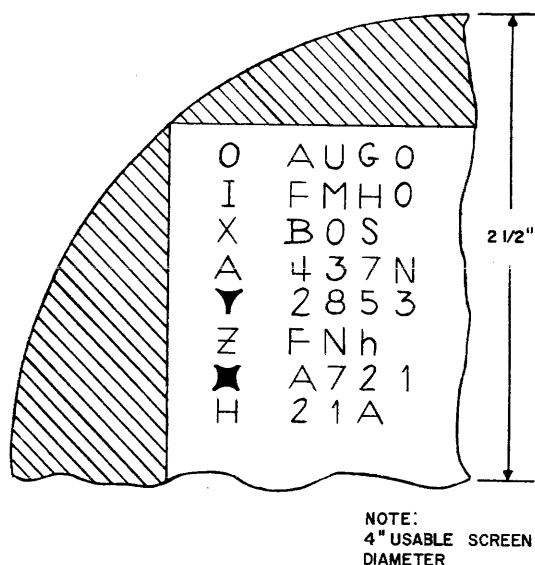
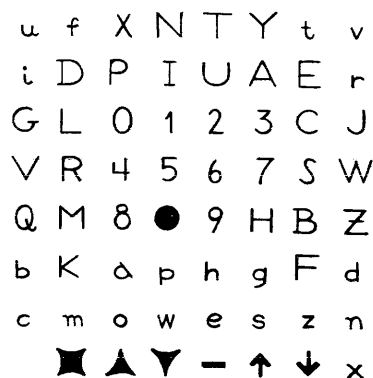


Figure 1-142. Typical Digital Display

Figure 1-143. Character Forming Matrix,
Digital Display Tube

acters of a digital display message are selected, they are positioned and written on the electrostatic storage mesh, making the mesh transparent for electrons passing from the flood gun to the phosphor on the digital display tube face. Unwritten portions of the storage mesh remain opaque to the flood gun electrons. Thus, only those characters written onto the storage mesh are displayed. Once written, their display continues until the storage mesh is erased, rendering it completely opaque to the flood gun electrons. New characters are then written on the storage mesh for display.

The 5-inch digital display tube, shown in figure 1-144, is a cathode-ray display tube using a short-persistence phosphor (P1). The tube contains a set of character selection plates and an eight-by-eight character-forming matrix for character generation. Unlike the situation display tube, the digital display writing beam is always defocused for character generation, making it large enough to cover and be shaped by the selected aperture in the character-forming matrix. The writing beam, after being deflected by the character selection plates, is returned to the axis of the tube by another set of plates, the compensation plates (electrostatic deflection). The beam is finally positioned on the storage mesh by a third set of plates, the character positioning plates (electrostatic deflection):

The characteristics indicate that the following signals must be supplied for each digital display:

- Character selection voltages (used also on correction plates with opposite sense)
- Character positioning voltages
- Erase gates to clear the storage mesh just prior to writing
- Intensification gates (no writing unless an intensify gate unblanks the writing beam).

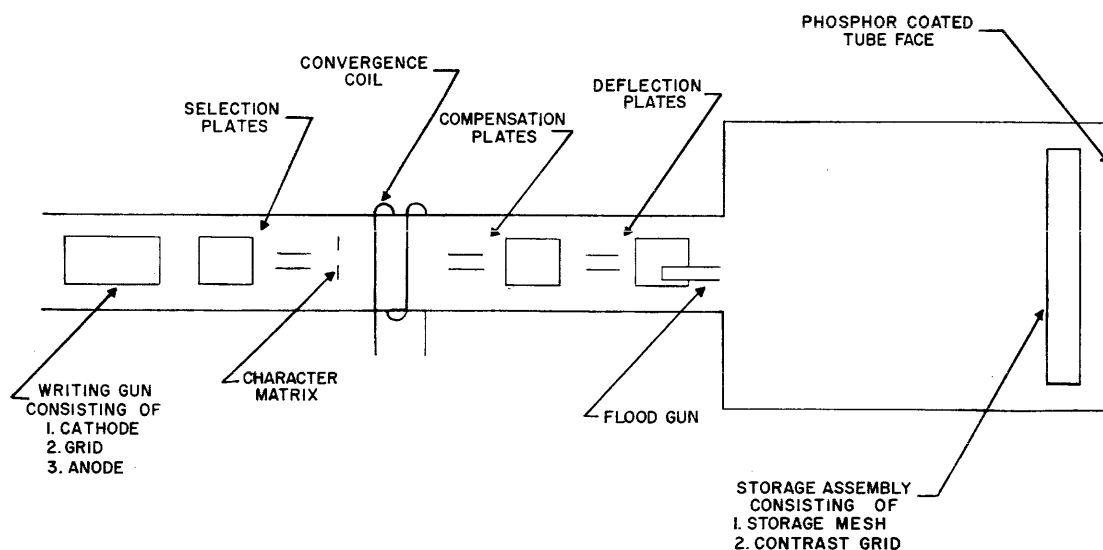


Figure 1-144. Digital Display Tube, Simplified Diagram

SECTION 2

SITUATION DISPLAY SUBSYSTEM OPERATION

2.1 INTRODUCTION

The situation display subsystem accepts information from the Drum System in digital form and presents this information in visible form at various display consoles. The Drum System acts as a time buffer between the rapidly operating Central Computer System and the more slowly operating Display System. The cathode-ray tube on which this information is displayed has already been described. The form in which information is presented to the situation display subsystem and the operations necessary to convert that information into the form required by the situation display tube will be the subject of this section.

However, before discussing the exact form in which information is supplied to the situation display subsystem, it is necessary to point out that two types of information are handled by that subsystem. Although both types require a geographical reference, one type includes only semidigested data on air movements, where as the other type contains extremely concise groupings of all data on air movements. The first type, known as radar data, is produced by the Central Computer System from the radar information supplied by the Input System and stored on the radar data drum for delivery to the Display System. The second type, known as track data, is produced by the Central Computer System from radar data together with height-finder reports and other available information and is stored on the track display drum for delivery to the Display System. These two types of information are handled differently by the situation display subsystem and produce different types of displays. (Refer to table 1-68.)

To facilitate the handling of these two types of information, the situation display subsystem alternately ac-

cepts all radar data, then all track data, then all radar data again, and so on. Display of information by the situation display subsystem is continuously recurring while AN/FSQ-7 Combat Direction Central is processing air defense data and is independent of program control.

2.2 SITUATION DISPLAY MESSAGES

2.2.1 Radar Data Messages

2.2.1.1 Information Contents

Radar target returns reported to AN/FSQ-7 Combat Direction Central are prepared by the Central Computer System for presentation by the Display System to observers. When a target report is entered into the Central Computer System, the received target polar co-ordinates referring to the radar set are converted into rectangular co-ordinates based on the subsector reference point. The target location (in rectangular co-ordinate form based on the subsector reference point) is then placed on one of the nine fields for the radar data drum as a radar data message.

When the radar data message is received by the Display System, it is eventually shown as a point on a 19-inch situation display tube, displaced from the center of the tube face by horizontal and vertical distances (X and Y co-ordinates, respectively) proportional to the target displacement from the subsector reference point along a parallel and meridian, respectively. For targets to the west or to the south of the subsector reference point, the X or Y co-ordinates are negative. In effect, the message is displayed just as a point is plotted on a 4-quadrant graph. Therefore, a radar data message must contain the X and Y co-ordinates of the target.

TABLE 1-68. DRUM SYSTEM LAYOUT OF SITUATION DISPLAY MESSAGE

MESSAGE CLASS	DRUM FIELDS	REGISTER (WORD)	MESSAGE SLOT	SLOTS PER FIELD	TOTAL MESSAGES
Track data	Track display (6)	32 bits	8 words	256	1536
Radar data	Radar data (9)*	24 bits	1 word	2048	18,432**

* Only eight fields are read during each display cycle.

** 16,384 messages may be read in one cycle.

The source of the information and the status of the report are included in each radar data message. Message sources may be of two types, either radar sets or IFF sets. A target reported by IFF must be friendly.

Such a target must be differentiated from targets reported by radar, which may or may not be friendly. The report status, showing a report as correlated or uncorrelated, indicates whether a track (producing a track data message) may be or has been initiated.

A radar data message contains the following information:

- a. X and Y co-ordinates, 1 sign bit and 10 magnitude bits for each co-ordinate, making a total of 22 bits for both co-ordinates
- b. Source, one bit
- c. Status, one bit

The entire message occupies 24 bits. It should be noted that if the X co-ordinate is zero, the message is not displayed.

The method used in reading and writing the radar data drum provides another classification of radar data messages. They are classed according to their age as present radar data or past radar data. Of the nine fields on the radar data drum, only eight will be read by the Display System during a drum display cycle. The ninth field is held for writing of new messages from the Central Computer System. Considering for the moment that the fields are arranged in a ring and numbered from 1 to 9, they will be written on in numerical order. Thus, field 1 will usually contain older information than field 2, and so on. When field 9 has been written on, writing will again commence on field 1, replacing the oldest information on the drum with the newest information.

Radar data drum reading by the Display System starts with the field which comes after the particular field being written on by the Central Computer System;

the reading then proceeds to the other seven fields in numerical order (with field 1 coming after field 9 in a continuous series). Radar data messages are thus displayed in order of their age, the oldest first and the most recent last. To indicate the age of the messages as they are displayed, the last field read by the Display System (the most recent information) supplies a display-bright signal to the Display System, while the preceding seven fields (containing the older information) almost always supply a display-dim signal. The most recent report on a given target, called a present radar data message, is displayed brightly on the situation display tubes; preceding reports on that target, called past radar data messages, are displayed dimly.

2.2.1.2 Effects of Programming on Display

The number of reports (radar data messages) on a specific target displayed by the Display System depends on the relation between the time spent by the Central Computer System in writing on each field and the time between successive reports on one target. If the program switches fields after each radar scan, it is possible to have one message on a target in each field; the result would be a display of eight successive target positions. If the fields were switched more slowly than the radar scan rate, two present radar data messages might be displayed at the same time.

Writing is transferred from one field to the next by the Drum System under program control. (Refer to table 1-69.) The *Operate* instruction, *PER* (76)₈, resets the scan counter, causing field 1 to be used for writing, and fields 2 through 9 to be used for display. The *Operate* instruction, *PER* (77)₈, steps the scan counter (adds 1), causing field 2 to be used for writing, while the other fields are displayed in the order 3 through 9, then 1. Subsequent steps of the scan counter shift writing and reading by one field. After writing

TABLE 1-69. DISPLAY SYSTEM INSTRUCTIONS

INSTRUCTION CODE	INSTRUCTION	REMARKS
<i>PER</i> (76) ₈ —	<i>Operate</i> (reset scan counter)	Causes writing on RD Field 1 ^{SET BY SDR INST.}
<i>PER</i> (77) ₈ —	<i>Operate</i> (step scan counter)	Shifts writing ^{SCAN COUNTER} to next RD field
<i>PER</i> (35) ₈ —	<i>Operate</i> (start DD)	Initiates digital display writing cycle
<i>PER</i> (20) ₈ —	<i>Operate</i> (area discriminator)	Initiates track initiation by area
<i>PER</i> (31) ₈ —	<i>Operate</i> (SD camera, mode 1)	Starts SD camera in mode 1
<i>PER</i> (32) ₈ —	<i>Operate</i> (SD camera, mode 2)	Starts SD camera in mode 2.
<i>BSN</i> (35) ₈ ×	<i>Sense</i> (SD camera on)	Branches to × if SD camera is on.
<i>BSN</i> (37) ₈ ×	<i>Sense</i> (display)	Branches to × if given during reading of TD drum.

field 9, stepping the scan counter would transfer writing operations back to field 1, completing the cycle, but resetting the scan counter is more reliable and is preferred.

Since the Display System reading cycle is asynchronous with Central Computer System operations, it is possible that the program will shift writing operations to a field as it is being read for display. Should this happen, display of that field is suppressed by the Drum System and the display begins with the next field, continuing through the field which had been written on by the Central Computer System before the shift.

Each radar data message is further identified as

belonging to a specific category. There are eight categories of radar data messages, each containing X and Y co-ordinates, distinguished in the Display System by their combined source, status, and age. The eight categories are displayed differently; a symbol represents the source and status of a message, while the symbol brightness indicates the age of the message. (See fig. 1-145.) The categories to be displayed at a given console are under the control of the operator, who may choose the categories (of those wired to the console) to be displayed and may suppress all others.

2.2.1.3 Radar Data Message Drum Layout

A radar data message is contained in 24 bits; the radar data drum is divided into 9 fields of 24-bit registers instead of the usual 6 fields of 32-bit registers. The 24 bits of a radar data message are laid out in a register as shown in figure 1-146. The first 11 bits contain the X co-ordinate of the message. The next bit is the source bit, containing a 1 if the source was IFF or a 0 if the source was radar. The status bit follows the source bit and contains a 1 if the message is correlated or a 0 if the message is uncorrelated. The final 11 bits contain the Y co-ordinate. The age of the message is indicated by the intensification signal (display-bright or display-dim) from the field containing the message. The intensification signal is determined by the position of that field with respect to the field being written by the Central Computer System.

2.2.2 Track Data Messages

2.2.2.1 General

Track data messages are delivered from the Central Computer System to the Display System via the track display drum. A track data message can cause the display of almost all the information on a target which can be compiled by the Central Computer System. A complete situation display may include up to 13 characters, a point, and a vector, positioned as a group on the face of the situation display tube to represent the

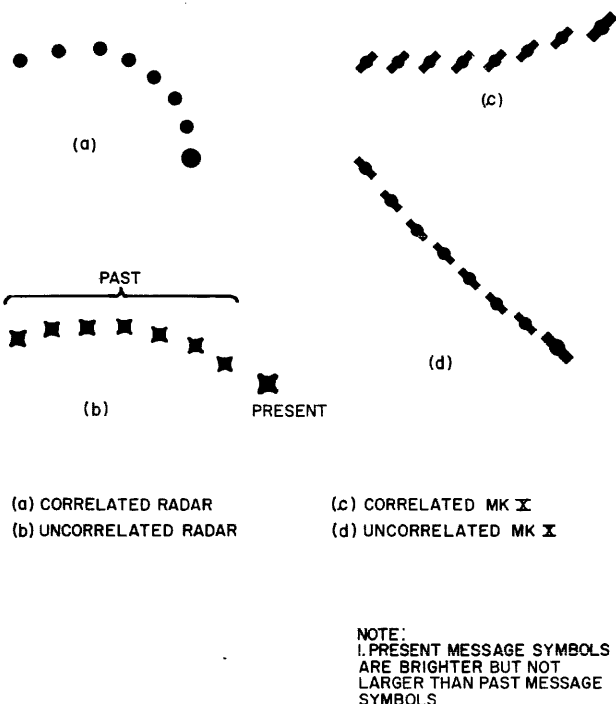


Figure 1-145. Typical Radar Data Message Displays

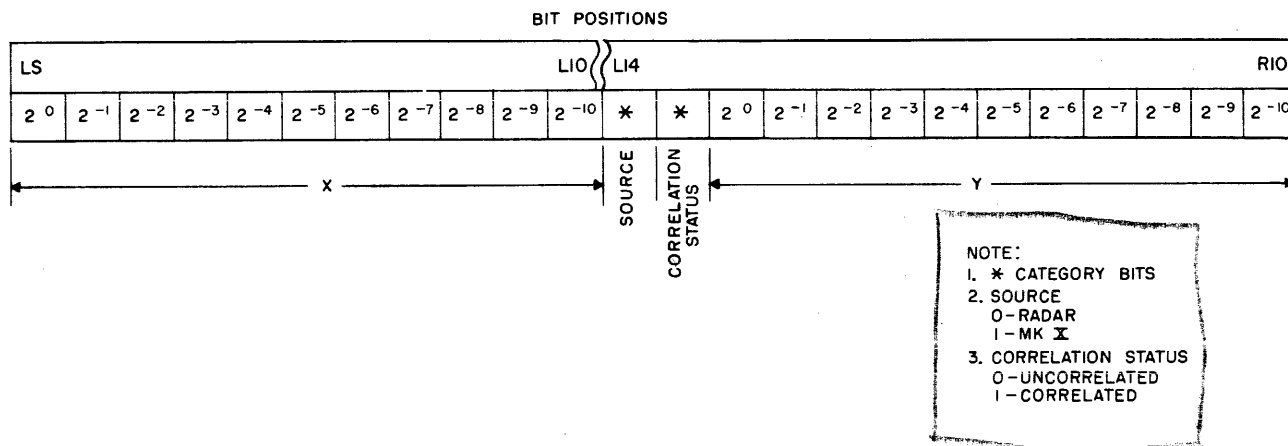


Figure 1-146. Radar Data Message Drum Layout

target position with respect to the sector reference point. To contain all this information, a track data message comprises eight 32-bit words (as compared to the single 24-bit word for a radar message).

The length of a track data message allows flexibility of use; by generating some or all of the characters it is able to display, the message can cause displays of targets in track and also of such matters as the location of an air base, the intercept point for a particular hostile, or any item of information requiring only 1 vector and up to 13 characters for its presentation. Another type of track data message is capable of causing the display of four independently positioned vectors and a group of four characters. This second type can be used for the display of such things as the flight history or flight plan of a target, a geographical boundary, or a mountain range. The two types of messages are designated tabular messages (13 characters plus vector) and vector messages (4 vectors plus a group of 4 characters). Tabular messages are further designated as track messages, which show targets in track, and as information messages, which show air bases or other geographical features.

Both types of track data messages are capable of being addressed to any of the more than 100 situation display consoles in the Central Computer System. They can also be sorted into 31 categories (similar to the 8 categories of radar data messages), which can be displayed or suppressed at the discretion of the operator

at a console. Three of the eight 32-bit words of each track data message are reserved solely for addressing by category and display assignment.

Although tabular and vector track data messages are of the same length, their contents differ sufficiently to require separate discussion of each type.

2.2.2.2 Tabular Messages

A tabular message contains eight 32-bit words. These words control the display of up to 13 characters plus a vector, and sort the message into one of 31 categories and up to 90 display assignment bit (DAB) assignments. Three of the words are reserved for the addressing of the message by category and display assignments. A fourth word contains the X and Y co-ordinates which position the message on the face of the tube, leaving 4 words to control the selection of the 13 characters and the generation of the vector. (Eighteen characters can be selected by each message, although only 13 characters may be displayed at any one time.)

The character-forming matrix of a situation display tube contains its characters in an eight-by-eight array. A single character can be selected by row and column with two octonary numbers, each written in three binary bits. Thus, the selection of one character requires six bits, three for each character-forming matrix co-ordinate. (See fig. 1-147.) Character address co-ordinates are designated as x and y, as opposed to the message position co-ordinates, X and Y.

OCTONARY ADDRESSES OF CHARACTERS
ON CHARACTER MATRIX

CHAR	x	y	CHAR	x	y	CHAR	x	y	CHAR	x	y
A	1	2	Q	4	7	7	1	0	+	7	4
B	2	7	R	5	0	8	6	7	◀	0	4
C	2	1	S	2	0	9	0	7	≡	1	4
D	5	2	T	0	3	↖	0	6	×	5	5
E	2	2	U	0	2	—	1	6	●	4	5
F	2	6	V	4	0	◻	4	6	+	2	5
G	4	1	W	3	0	◼	3	6	○	3	5
H	1	7	X	6	3	◻	5	4	■	7	7
I	7	2	Y	1	3	=	2	4	▶	3	4
J	3	1	Z	3	7	↘	6	5	⊥	4	3
K	5	6		7	1	↖	7	5	⌋	5	3
L	5	1	2	0	1	•	0	5	⊥	2	3
M	5	7	3	1	1	—	1	5	⌋	3	3
N	7	3	4	6	0	●	6	4	⊙	4	2
O	6	1	5	7	0	◻	6	6	▶	3	2
P	6	2	6	0	0	+	7	6	BLANK	4	4

CHARACTER MATRIX

OCTONARY

OCTONARY

x	4	5	6	7	0	1	2	3		
y	3	⊥	↖	X	N	T	Y	⊥	⌋	011
2	⊙	D	P	I	U	A	E	▶		010
1	G	L	0	1	2	3	C	J		001
0	V	R	4	5	6	7	S	W		000
7	Q	M	8	■	9	H	B	Z		111
6	◻	K	◻	+	↖	—	F	◼		110
5	•	×	↘	↖	•	—	+	○		101
4	⌋	●	+	◻	≡	=	▶			100
	100	101	110	111	000	001	010	011	x	y

BINARY

Figure 1-147. Octonary Character Addressing, Situation Display Tube

To select 13 characters, 78 bits (6 x 13) are required, comprising 2-1/2 words. Vector generation requires 14 bits (6 absolute magnitude bits and 1 sign bit for the \bar{X} component, and the same amount for the \bar{Y} component), nearly filling a third word. The remaining word contains selection co-ordinates for 5 other characters, allowing each tabular message to contain up to 18 characters plus a point feature and a vector feature. These characters can be selected, at the discretion of the operator, for display in the 13 allowable character positions, in place of some of the first 13 characters generated. The position of each character within the message display is determined by the position of the selection co-ordinates of that character within the message.

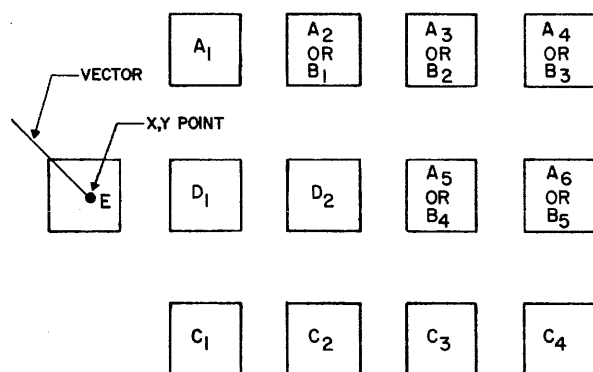


Figure 1-148. Tabular Message Display Format

The complete pattern of a tabular message when displayed on the situation display tube is shown in figure 1-149. The four-by-three format of characters, the point, the vector, and the character at the vector origin comprise the maximum display for a tabular message. A complete display of this kind describes a target and a track and results from a type of message called a tabular track message. Some tabular messages, called tabular information messages, contain no vector (since they do not refer to moving targets) and may employ from 1 to 13 of the 18 characters. (See fig. 1-149.)

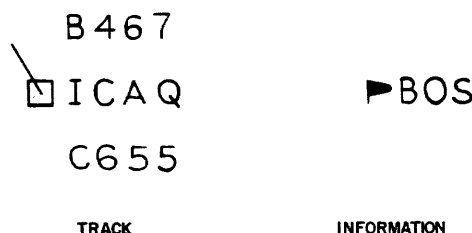


Figure 1-149. Typical Tabular Message Displays

The information which a message may represent is classified into seven features: the point feature, the vector feature, and five others designated by letters A through E. Since the positions which the characters may occupy within a message display are assigned to parti-

TABLE 1-70. FEATURE AND CHARACTER POSITION NOMENCLATURE
FOR TABULAR TRACK DATA MESSAGES

MESSAGE TYPE	MESSAGE DISPLAY SELECTION	FEATURE	POSITIONS	FEATURE DISPLAY SELECTION
Track	Category and/or DAB	Vector	Origin at X, Y	Bypassed
		Point	X, Y	Light gun
		A	A ₁ -A ₆	FSS*
		B	B ₁ -B ₅ **	FSS
		C	C ₁ C ₄	FSS
		D	D ₁ -D ₂	FSS
		E	X, Y	FSS
Information	Category and/or DAB	A	A ₁ -A ₆	Bypassed
		C	C ₁ -C ₄	Bypassed
		D	D ₁ -D ₂	Bypassed
		E	X, Y	Bypassed

* Feature selection switches.

** Positions time-shared with A₁-A₆.

cular features, the character positions are designated by the letters of the features which may be displayed in those positions and are differentiated by subscripts. Thus, the feature displayed at the vector origin is designated as the E feature and is the only feature displayed in that position. On the other hand, the C feature may appear in character positions C₁ through C₄. Further, in character positions A₂ through A₆, the B feature may be displayed instead of an A feature.

Tabular track messages may contain features A through E. All 18 characters for each message are delivered to each console, where the operator may suppress any feature except the vector. This selection control prevents the superposition of those features that share the same character display positions. Tabular information messages may contain only features A, C, D, and E, over which the console operator has no display control. (Refer to table 1-70.) Features that are not under the control of the operator are called bypass features. (The point feature is selected only when a light gun is used.)

In addition to the information just discussed, a tabular message also contains five control bits. The control bits indicate whether:

- A particular track data message is tabular or vector
- A particular tabular message is an information message or track message
- The message should be displayed or not (prevents intensification when empty slots are read from the track display drum)
- A light gun can or cannot be used in referring the message to the Central Computer System via the manual data input element
- The characters of a message should be displayed above, below, or to the left or right of the origin of the vector feature.

The position bit plus the direction of the \dot{X} or \dot{Y} components of the vector determines the placement of the four-by-three format of characters with respect to the

WORD	BIT POSITION																																															
	LS	LI	L2	L3	L4	L5	L6	L7	L8	L9	LI0	LI1	LI2	LI3	LI4	LI5	RS	RI	R2	R3	R4	R5	R6	R7	R8	R9	RI0	RI1	RI2	RI3	RI4	RI5																
0	*	A ₁		A ₂		A ₃		A ₄		E		*	A ₁		A ₂		A ₃		A ₄		E																											
		X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰			Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰															
1	X												Y																																			
	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸	2 ⁻⁹	2 ⁻¹⁰	2 ⁻¹¹	2 ⁻¹²					2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸	2 ⁻⁹	2 ⁻¹⁰	2 ⁻¹¹	2 ⁻¹²																		
2	CATEGORY						*	DAB																																								
	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰																																											
3	DAB																																															
	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58																
4	DAB																																															
	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90																
5	X												C ₂				C ₃				C ₄				Y												C ₂				C ₃				C ₄			
	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰																
6	*	B ₁		B ₂		B ₃		B ₄		B ₅				B ₁		B ₂		B ₃		B ₄		B ₅																										
		X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰			Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰															
7	*	D ₁		D ₂		A ₅		A ₆		C ₁				D ₁		D ₂		A ₅		A ₆		C ₁																										
		X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰	X ₂ ²	X ₂ ¹	X ₂ ⁰			Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰	Y ₂ ²	Y ₂ ¹	Y ₂ ⁰															

* = CONTROL BIT			
WORD	POSITION	CONTENT	MEANING
0	LS	0	TRACK MESSAGE
		1	INFORMATION MESSAGE
	RS	0	TABULAR FORM
2	L5	0	SUPPRESS DISPLAY
		1	ALLOW DISPLAY
6	LS	0	LIGHT GUN MAY BE USED
		1	LIGHT GUN MAY NOT BE USED
7	LS	0	CHARACTERS TO LEFT OR RIGHT OF VECTOR
		1	CHARACTERS ABOVE OR BELOW VECTOR

Figure 1-150. Tabular Message Drum Layout

origin of the vector. The placement is always such that the vector points away from the characters in the format.

2.2.2.3 Tabular Message Drum Layout

The eight words of a tabular message are designated as word 0 through word 7. (See fig. 1-150.) Word 0 contains the first two control bits and the character selection co-ordinates, x and y , for character positions A_1 through A_4 and E . All x , X , and \bar{X} co-ordinates are in the left half-word, while all y , Y , \bar{Y} co-ordinates are in the right half-word. The control bit in the LS position is a 1 if the message is an information message and a 0 if the message is a track message. The control bit in the RS position is 0 for tabular messages and 1 for vector messages.

Word 1 contains the X and Y co-ordinates of the message, 13 bits each, of which 10 are used to position the entire message on the face of the situation display tube. The 10 bits of each half-word that is used are selected by the expansion and off-centering relays in each display console.

Bits LS through L4 of word 2 indicate the category within which the message falls. Although 5 binary bits can count from 0 through 31, 0 is not used and 31 is used as a test category, leaving 30 categories for normal messages. The control bit in the L5 position contains a 1 if the message is to be displayed and a 0 if there is to be no display.

The remaining bits of word 2 plus all of words 3 and 4 are display assignment bits (DABs). There are 90 bits, almost enough to assign each console to a separate DAB position. Those tabular or vector message DAB positions that contain a 1 will force the display of the message on the consoles permanently assigned to those bit positions. The display occurs at a console whether the operator selects that message category for display or not. A display of this kind is called a forced DAB display to distinguish it from those DAB displays which may be selected by category switches. In a tabular track message routed to a particular console, the operator may suppress display of any or all features except the vector. However, he has no feature selection control over tabular information messages.

Word 5 contains the vector generation information, \bar{X} and \bar{Y} , plus the character selection co-ordinates for character positions C_2 through C_4 . Bit LS of word 6 is a control bit. If LS contains 0, a light gun may be used on the message; if it contains a 1, the light gun will have no effect. Bit RS is not used. The remaining 15 bits in each half word 6 contain the character selection co-ordinates for character positions B_1 through B_5 (shared with display positions A_2 through A_6).

Bit LS of word 7 is the position bit which, together with bit LS or RS of word 5 (the sign bits of \bar{X} and \bar{Y}), determines the position of the character format with

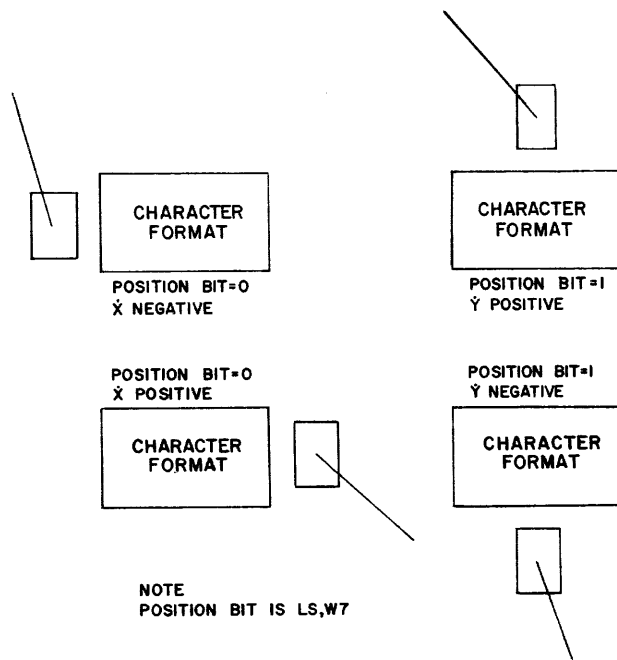


Figure 1-151. Effect of Position Bit on Character Format

respect to the vector. (See fig. 1-151.) Bit RS of word 7 is not used. The remaining 15 bits in each half of word 7 contain the character selection co-ordinates for character positions D_1 and D_2 , A_5 and A_6 , and C_1 .

The eight words described comprise a slot on the track display drum with the eight registers of each slot contiguous. The track display drum has a capacity of 1,536 slots.

2.2.2.4 Vector Message

A vector message containing eight 32-bit words controls the display of up to 4 independently positioned vectors plus a group of 4 characters on a situation display, its addressing to one of more than 100 consoles, and its sorting into one of 31 categories. As in the tabular message, three of the eight words of a vector message are reserved for message addressing by DAB and by category. A fourth word contains the x and y character selection co-ordinates for the four characters. Each of the remaining four words controls the positioning and generation of a vector.

Each of the 4 characters in the message is selected from the 63 characters and 1 blank or void on the matrix by 2 octonary numbers that indicate the row and the column of the character. The blank or void is comparable in use to the space bar on a typewriter. Each octonary number, designating an x or a y character selection co-ordinate, is written in three binary bits. Thus, each character is selected by 6 bits, with 24 bits required for all 4 characters. The characters in a vector

message group are known as G characters. (Refer to table 1-71.) The position of each character with respect to the other three (character position G_1 through G_4) is implicit in the position of its x and y co-ordinates in the vector message. On the other hand, the position of the character group is determined by X_4 , Y_4 , the origin of the fourth vector of the message, and the signs of its components, \dot{X}_4 , \dot{Y}_4 . (See fig. 1-152.)

With one word occupied with the G characters and three words reserved for addressing, four words remain for the generation and location of four vectors. One word is used for each vector by representing its origin co-ordinates and its components with fewer significant digits than is done for the vector of a tabular message. The vector origin is located by X_n and Y_n co-ordinates of 10 bits each (as compared to the 13 bits each for the

TABLE 1-71. MESSAGE COMPONENT NOMENCLATURE FOR VECTOR TRACK DATA MESSAGES

MESSAGE DISPLAY SELECTION	COMPONENT	POSITION (VECTOR ORIGIN)	COMPONENT DISPLAY SELECTION
Category and/or DAB	Vector 1	X_1, Y_1	Bypassed
	Vector 2	X_2, Y_2	Bypassed
	Vector 3	X_3, Y_3	Bypassed
	Vector 4	X_4, Y_4	Bypassed
	G characters	$G_1-G_4^*$	Bypassed

* G characters located with respect to X_4, Y_4 .

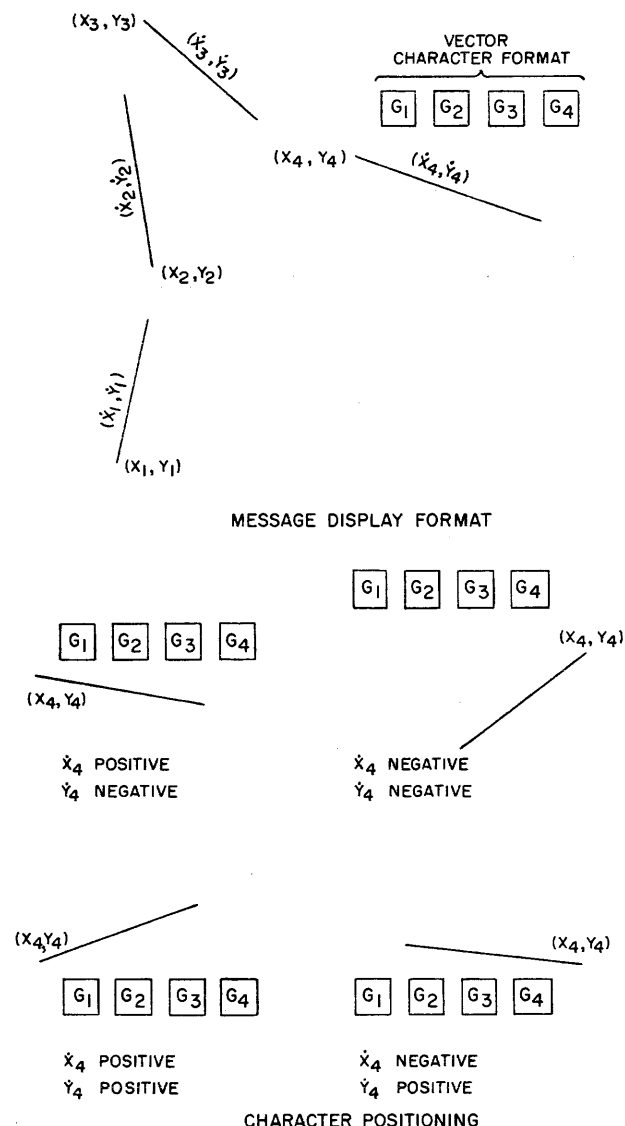


Figure 1-152. Vector Message Display Format and Character Positioning

X and Y co-ordinates of a tabular message). Vector components \dot{X}_n and \dot{Y}_n are described in six bits each (as compared to seven bits each for \dot{X} and \dot{Y} components of the tabular message vector). Thus, each of the 4 remaining words can contain a complete vector, with 20 bits for location and 12 bits for generation, making a total of 32 bits. The fourth vector co-ordinates, \dot{X}_4 and \dot{Y}_4 , and the sign bits of the fourth vector components X_4 and Y_4 , also position the G characters on the tube face. The G characters are positioned so that the fourth vector is above or below and pointed away from them.

Only two control bits are used in the vector message. The first specifies that the message is a vector message; the second determines whether or not the message will be displayed. The vector message, with its four independently positioned vectors, can be used to display the flight history of a target or the flight plan of a friendly plane (filed in advance of the flight), a geographical boundary, a mountain range, or the subsector boundary. (See fig. 1-153.) A vector message may also be placed around a track message as an attention device.

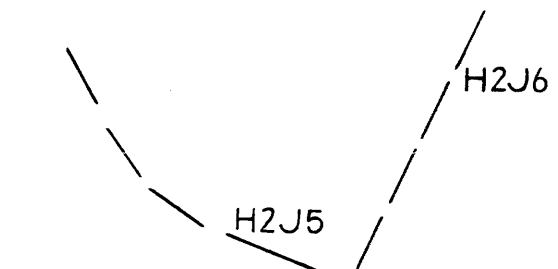


Figure 1-153. Typical Vector Message Displays

2.2.2.5 Vector Message Drum Layout

The eight words of a vector message on the track display drum are designated as words 0 through 7.

WORD	BIT POSITION																																
	LS	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	RS	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	
0																	*																
1																																	
2																																	
3																																	
4																																	
5																																	
6																																	
7																																	

* CONTROL BIT			
WORD	POSITION	CONTENT	MEANING
0	RS	0	TABULAR FORM
		1	VECTOR FORM
2	L5	0	SUPPRESS DISPLAY
		1	ALLOW DISPLAY

* CONTROL BIT

WORD	POSITION	CONTENT	MEANING
0	RS	0	TABULAR FORM
		1	VECTOR FORM
2	L5	0	SUPPRESS DISPLAY
		1	ALLOW DISPLAY

Figure 1-154. Vector Message Drum Layout

(See fig. 1-154.) the x and y character selection co-ordinates for character positions G_1 through G_4 are contained in word 0. The vector message, like the tabular message, has all x, X, and \bar{X} co-ordinates in the left half-words on the drum and all the y, Y, and \bar{Y} co-ordinates in the right half-words. The control bit in the RS position of word 0 must contain a 1 in a vector message. A 0 in that position designates the message as tabular. Bits LS, L13 through L15, and R13 through R15 of word 0 are not used in the vector message.

Word 1 controls the location and generation of the first vector. Bits LS through L9 contain the \bar{X}_1 location co-ordinate and bits L10 through L15 contain the \bar{X}_1 vector component. Bits RS through R9 contains the \bar{Y}_1 location co-ordinate while bits R10 through R15 contain the \bar{Y}_1 vector component.

Bits LS through L4 of word 2 indicate the category within which the message falls. Although 5 binary bits can count from 0 through 31, 0 is not used and 31 is used as a test category, leaving 30 categories for normal messages. The control bit in the L5 position contains a 1 if the message is to be displayed and a 0 if there is to be no display. An empty slot is prevented from intensifying the display tubes by having a 0 in the L5 position.

The remaining bits of word 2 plus all of words 3 and 4 are display assignment bits (DABs). There are 90 bits, approximately enough to assign each situation display console to a separate DAB position. Those tabular or vector message DAB positions that contain a 1 will force the display of the message on the consoles permanently assigned to those bit positions. The display occurs at a console whether or not the operator selected that message category for display. A display of this kind is called a forced DAB display to distinguish it from those displays which may be selected by category switches. No feature selection control at a console is possible with a vector message. Unlike tabular track message displays, in which an operator may suppress all features except the vector, a vector message is either fully displayed or not displayed at all.

The contents of words 5, 6, and 7 are identical to the contents of word 1, each word locating and generating one vector. The \bar{X}_4 and \bar{Y}_4 co-ordinates and the sign bits of the \bar{X}_4 and \bar{Y}_4 components in word 7 also position the G characters on the face of the situation display tube above or below the origin of the fourth vector, with the vector pointed away from the character group.

The eight words of a vector message comprise one slot on the track display drum with the eight registers of

TABLE 1-72. CONTROL-BIT FUNCTIONS, SITUATION DISPLAY MESSAGES

MESSAGE TYPE	FORM	WORD	POSITION	BIT	CONTENT	MEANING		
Radar data			L14	0		Radar source		
				1		IFF source		
			L15	0		Uncorrelated		
				1		Correlated		
Track data	Tabular	0	LS	0		Track message		
				1		Information message		
			RS	0		Tabular form		
				1		Vector form		
			2	LS-L4	—		Category bits	
					0		Suppress display	
			2	L5	1		Allow display	
					—			
		2	L6-R15	*		DABs		
		3	LS-R15	*		DABs		
		4	LS-R15	*		DABs		
		6	LS	1		Light gun inoperative		
				0		Light gun may be used		
		7	LS	0		Position characters in format to left or right of vector		
				1		Position characters in format above or below vector		
			Vector	0	RS	0		Tabular message
						1		Vector message
						2-4 (same as tabular form)		

* A binary 1 in the DAB position assigned to a particular console forces display of that message at that console.

each slot contiguous. Table 1-72 provides a summary of control-bit functions in all three types of situation display messages.

2.3 SITUATION DISPLAY GENERATOR ELEMENT

2.3.1 Element Function

The situation display generator element consists of the equipment that converts the information received from the Drum System into the forms required by the display tube. The functions that it performs are as follows:

- Separates the X and Y message positioning coordinates out of each message and routes these co-ordinates in digital form to the situation display indicator element.
- Converts all x and y character selection co-ordinates from their digital form into analog voltage form (in which numerical value is represented by amplitude) and routes these character selection voltages to the situation display indicator element.

- c. Converts the digital X and Y vector components into analog form and routes these voltages to the situation display indicator element.
- d. Generates character-positioning voltages for the characters in track data messages and routes these voltages to the situation display indicator element.
- e. Converts the category and display assignment bit (DAB) words of track data messages into display gates which are sent to the situation display indicator element to provide for manual selection of messages to be displayed at each console.
- f. Provides the defocus gate necessary for character generation in the situation display tube.
- g. Decodes the control or category bits in a message.
- h. Produces the feature display gates sent to the situation display indicator element to provide for selection of the features within a tabular message to be displayed at a particular console.
- i. Provides for the transfer of situation display information back to the Central Computer System through the manual input element.

The situation display generator element also has the overall function of timing the entire situation display subsystem, by supplying the situation display indicator element with the necessary inputs in the correct sequence and at the correct time.

2.3.2 Situation Display Cycle

The operation of the situation display generator element is largely determined by the timing and sequence of reading situation display messages from the drums, identified as the situation display cycle. This situation display cycle is designed to match the capabilities of the situation display generator element. The situation display cycle includes the reading both of the radar data drum and of the track display drum. Each drum is read alternately; the radar data drum is approximately 1 second and the track display drum is approximately 1.6 seconds. The entire situation display cycle lasts approximately 2.54 seconds.

Although words may be read from a drum at 10-microsecond intervals, situation display message words supplied to the situation display generator element at so rapid a speed would overload the Display System. Sixty microseconds are allowed for the display of a single radar data message and 1040 microseconds are allowed

for a track data message. These intervals between messages are provided by the process of reading situation display messages from their drum by precession. (Refer to Chapter 4, 2.3.2.3.)

The precession pattern used in reading the radar data drum involves skipping five registers between message reading. (Each register contains one message.) Since the drum moves successive registers into reading position 10 microseconds apart, this precession pattern provides 1-word messages to the situation display generator element at the required 60-microsecond intervals. Six complete drum revolutions are required to read one entire field of the radar data drum. Each revolution takes approximately 20 milliseconds; thus, each field is read in 120 milliseconds and all 8 fields are read in approximately 1 second (960 milliseconds).

The precession pattern used in reading the track display drum involves skipping 12 slots of 8 registers each between the reading of each slot. However, all 8 registers of a slot are read sequentially; 8 words are delivered to the situation display element at 10-microsecond intervals, spanning 80 microseconds. The 12 skipped slots introduce a dead period in information transfers from the drum of 960 microseconds between each track data message presented to the situation display generator element. Thus, 1040 microseconds (80 microseconds for message reading and 960 microseconds between readings) are allowed for the reception and generation of each track data message. Thirteen revolutions of the track display drum at approximately 20 milliseconds per revolution are required to read each field completely. With six fields to be read, the entire track display drum can be read in approximately 1.56 seconds.

The overall flow of information during a complete situation display cycle is as follows. Starting with the radar data drum cycle, 16,384 registers, each of which may contain a radar data message, are read into the situation display generator element at 60-microsecond intervals; then, 1536 eight-register slots, each of which may contain a track data message, are read into the situation display generator element at 1040-microsecond intervals. The situation display cycle then repeats and recurs continuously while an air defense problem is in progress. A summary of situation display cycle timing is given in table 1-73.

2.3.3 Element Operation

The situation display information that is essential for the functioning of the situation display system is supplied to the situation display generator element (SDGE) in accordance with the situation display cycle. Two types of input signals are required, control signals

TABLE 1-73. SITUATION DISPLAY CYCLE TIMING

DRUM READ	OPERATION	TIME (MICROSECONDS)	TOTAL TIME (SECONDS)
Track display	Read and display each message (message display cycle)	1040	
	Position each message	40	
	Select and position each character	20	
		25	
	Generate each vector	50	
	Intensify each vector	47 to 50	
	Switch between characters	5	
	All		1.56
Radar data	Read and display each message (message display cycle)	60	
	Select and position each message	30	
	Intensify each message	25	
	Switch between messages	5	
	All		0.96
	Switch between drums		0.02
	Full situation display cycle		2.54

and information signals. A third input is the information-transfer signal from the manual data input element of the Input System. This signal gates information stored in the SDGE through the manual data input element and back into the Central Computer System.

The information for the SDGE is passed through the input switch section and the gate section of the Display System. (See fig. 1-155.) The input switch section functions to switch the SDGE from normal operation (signals supplied by the Drum System) to test operation (signals supplied by the display tester element). The input gates section allows transfer of information from the Drum System when it is gated by signals from the situation display timer. From there, the words that contain information pertaining to a target are transferred to the word storage section, while the words that contain control or distribution data are transferred to the DAB storage and output section.

The word storage section temporarily stores the words of a track data message during a message display

cycle and supplies them to other parts of the situation display generator element on demand. The words containing the x and y character co-ordinates are transferred to the character selection section in digital form. In this section, they are converted to analog voltages which are forwarded to the situation display tubes where they are used to select characters on the character-forming matrix. The words that contain the \bar{X} and \bar{Y} co-ordinates, plus the character-correction-position signals from the character selection section, are transferred from the word storage section to the vector and format generator. This generator produces analog voltages to be used for positioning all characters (except the E character) in a track data message and for the generation of all vectors. These voltages are applied to the second set of deflection plates in the situation display tubes. The words that contain positioning information for the second, third, and fourth vector of a vector message are transferred to the XY register and drivers section, which stores and transfers the vector positioning information to the situation display tubes.

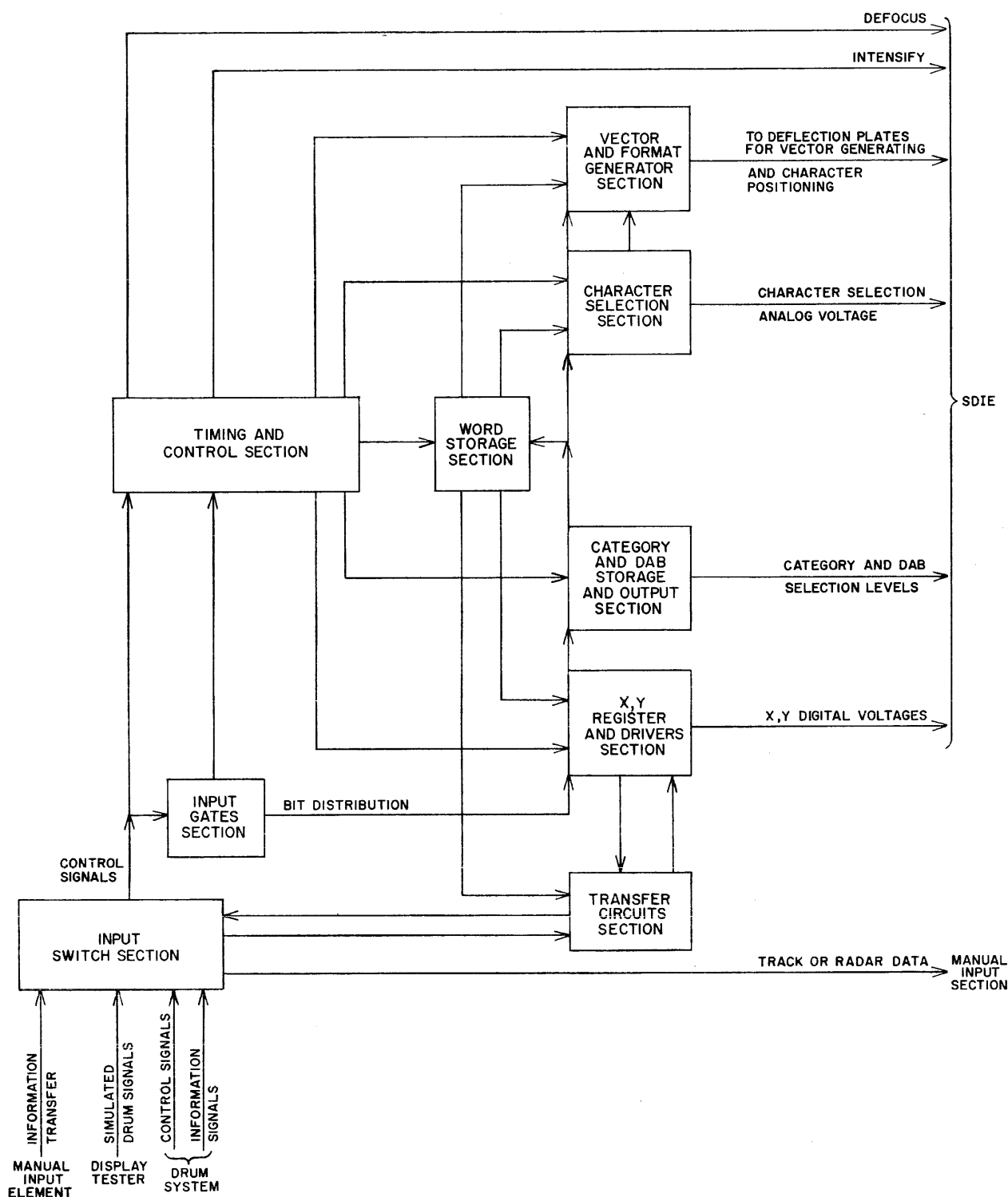


Figure 1-155. Situation Display Generator Element

The category and DAB storage section receives the remaining information, or words 2, 3, and 4, from the input gates section. This information consists of the category and DAB bits of a track data message or the category encoding signals of a radar data message. For

each message, the circuits in this section preclude display gates that are connected to the category display switches in a display console. Thus, while every character of every message is potentially capable of display at every console, the category and DAB selection gates may, by

their absence, prevent display of entire messages at some of the consoles.

tion being displayed on any other section. This section is called the maintenance display console.

2.4 SITUATION DISPLAY INDICATOR ELEMENT

2.4.1 General

Each console housing a section of the situation display indicator element receives the outputs from the situation display generator element and generates a display conditioned by the selection of categories, features, and combinations thereof made by the operator at that console. Since each section is functionally identical, a description of the operation of only one suffices.

In addition to the typical situation display sections, a unique situation display section also exists. The purpose of this section is to select and display the informa-

2.4.2 Display Selection

The intensification signal, which unblanks the writing beam of the situation display tube, is generated only if four different signals are applied simultaneously to the intensification unit shown in figure 1-156. These signals include:

- The intensify signal which accompanies all character selection and vector generation signals.
- A forced DAB or category gate or one of the DAB or category selection gates passed by the category switches at the console.

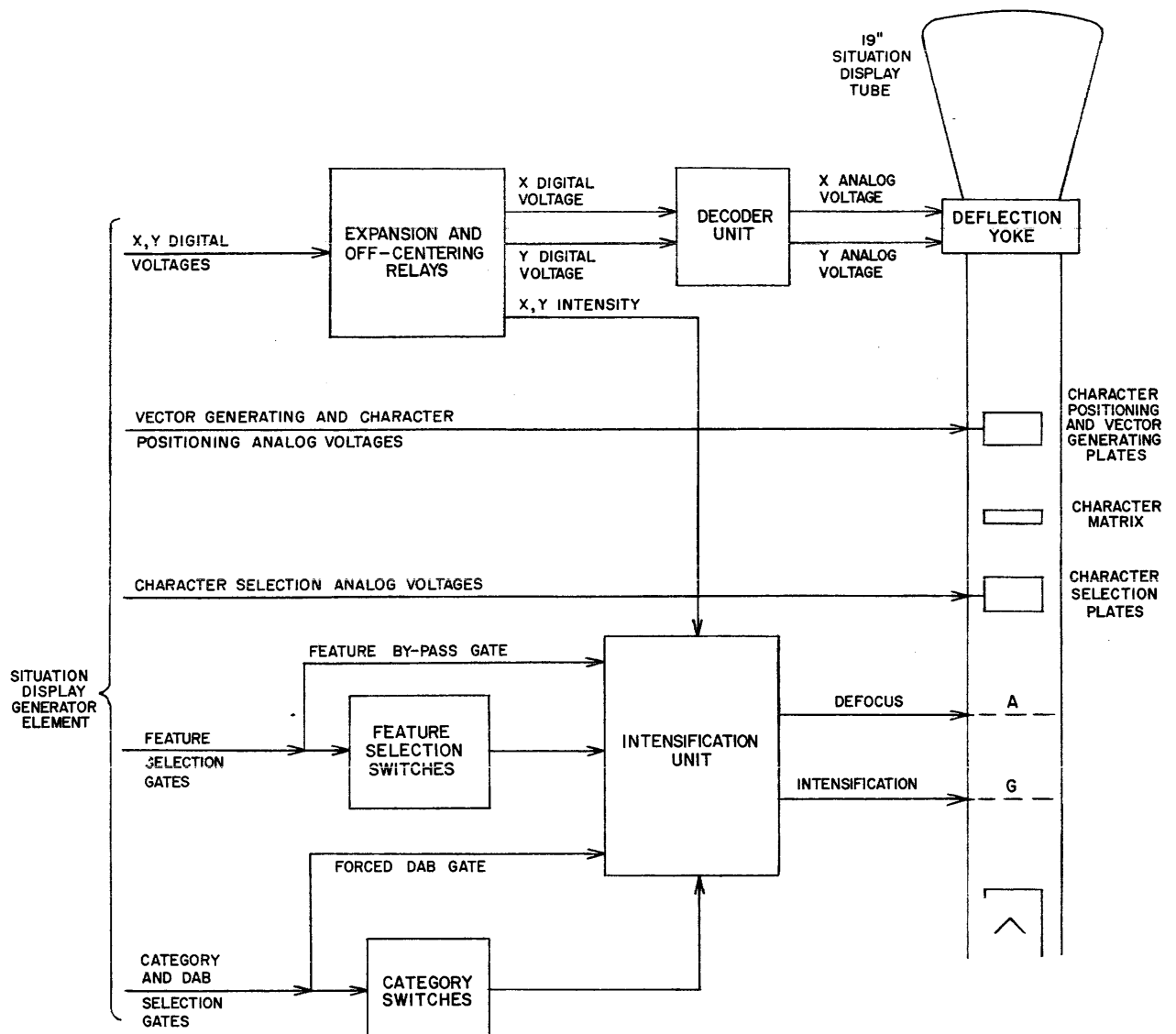


Figure 1-156. Situation Display Indicator Element

- c. A feature selection gate passed by the feature selection switches at the console or a feature bypass selection gate.
- d. An XY intensity signal generated at the console by the expansion and off-centering controls.

The first signal is always applied. The second is applied for all messages containing a 1 in the DAB position to which that console is assigned (forced DAB) and for all category gates wired as forced displays at that console. Nonforced or selectable messages are chosen for display by the operator by setting the category selection switches to supply one or more of the following display gates:

- a. Thirty-nine category display gates, 8 radar categories, 30 track data message categories, and 1 test category.
- b. Up to 50 mixed category and/or DAB display gates, each composed of up to 6 category and/or DAB display gates mixed to produce a composite selection gate.

Some of the 50 may be composite gates made up of 6 other composite gates, each selecting, for display, up to 36 categories and/or DABs at once. The operator has available 15 category switches, each of which can apply up to 4 display gates at once, thus allowing considerable flexibility in selection of displays. A total of 27 category and DAB display gates may enter each console. These gates will force display or be applied to the category switches for display selection, as determined by the wiring of a control panel within the console.

The third signal, the feature selection gate, allows selection for display of features within a message which has been selected (by category or DAB) for display. The fourth signal is used to prevent display of messages which have been expanded off the display area.

2.4.3 Expansion and Off-centering

After the generation and selection of messages for display is accomplished, the message must be positioned on the situation display tube face. The messages must be placed in positions (with respect to the center of the face) that correspond to the positions of the targets (with respect to the subsector reference point) to which the messages refer. The X and Y co-ordinates specifying message position are supplied to the indicator section still in digital form to allow for expansion and off-centering the display digitally. Expanding a display changes the distance between messages without changing character sizes or their positions within messages. A binary number can be doubled or quadrupled simply

by shifting its bits one or two positions left with respect to the binary point. Since expanded or off-center displays of portions of the subsector coverage area are desirable at most of the consoles to reduce the number of messages displayed, each console is provided with a set of off-centering control pushbuttons and an EXPANSION CONTROL switch.

The X and Y co-ordinates of a message are applied to digital-to-analog de-coders in each console through a set of relays. These relays are controlled by the three-position EXPANSION CONTROL switch, wired so that it may select one of four possible expansion scales in each of its three positions: CNDT (contracted), NORM (normal), and EXPD (expanded). The four possible expansion scales are x1, x2, x4, and x8 (times 1, 2, 4, and 8). In the case where the switch in the CNDT position selects the overall subsector area (x1 expansion for display), the bits of the X and Y co-ordinates are transferred to the X and Y decoders unshifted. In the NORM position, the switch would ordinarily provide for an x2 or an x4 expanded display, causing the X and Y co-ordinates to be delivered to the decoders shifted one or two places to the left. If the x2 scale is chosen in NORM position, the console can display only $\frac{1}{4}$ of the total area covered by the subsector. Each quarter is separately selectable for display at the console by control panel wiring within the console. Since a 50-percent overlap is provided between selectable areas, there are nine overlapping quarters of the x1 display coverage area which are separately selectable for display at x2 scale. Once the selection of a quarter for NORM display is made by control panel wiring, that quarter is always displayed when the EXPANSION CONTROL switch is placed in the NORM position.

If the x4 scale were chosen by the NORM position, only $\frac{1}{16}$ of the subsector area could be displayed at the console. One of 49 such portions (49 due to the 50-percent overlap provided) is selected for display by control panel wiring within the console. Once selection of a particular off-center area for display in NORM is made, this selection can be changed only by changing the control panel wiring.

A further expansion of display at a console is made by changing the EXPANSION CONTROL switch to the EXPD position. If the change from NORM to EXPD causes a change from x2 to x8, the area which can be displayed is $\frac{1}{16}$ of the NORM display area but only $\frac{1}{64}$ of the total subsector area. With the 50-percent overlap, there are 49 expanded display areas within the NORM display area. In order to select one of these areas for display at any one time, the off-centering pushbuttons (seven along the X axis and seven along the Y axis) become effective in the EXPD posi-

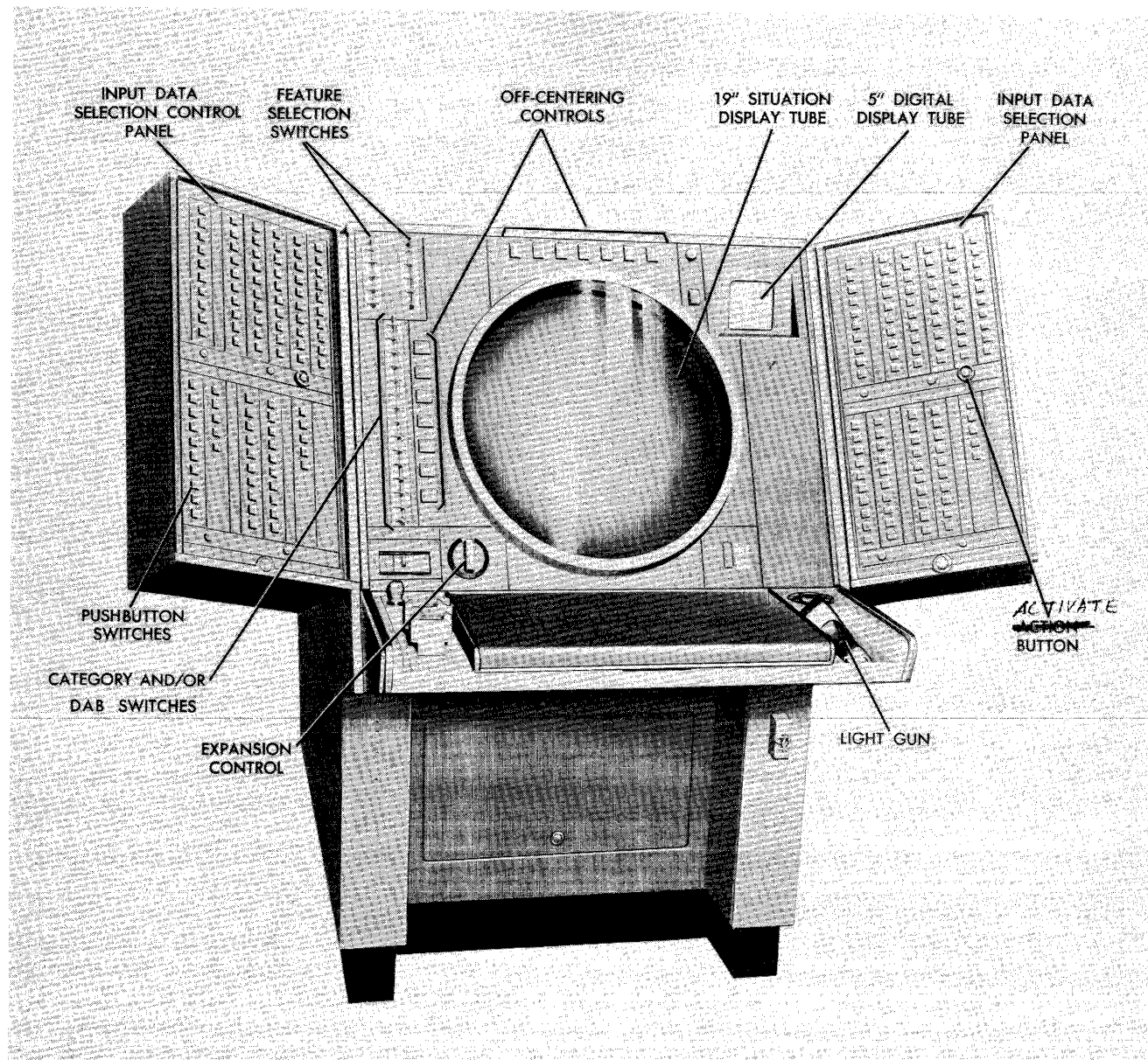


Figure 1-157. SD Console Controls

tion to perform the necessary selection. (See fig. 1-157.) By depressing an X and Y off-centering pushbutton whose positions together describe a particular point on the NORM display, that point is placed at the center of the EXPD display. The 14 pushbuttons provide 49 combinations, one for each expanded display area.

If the change from NORM to EXPD causes a change from x4 to x8, although the area selectable for expanded display is still only one sixty-fourth of the total subsector area, that area is one-quarter of the NORM display area. Thus, there are only nine selectable expanded display areas within the preselected NORM area and only six off-centering pushbuttons are needed

to select one of these areas for display. In this case, only three of each set of seven off-centering pushbuttons is operative. The choice of expansion scales for the settings of the EXPANSION CONTROL switch and hence of the operation of the off-centering pushbutton is determined by the function of each console.

The circuit operations involved in expanding a display and in selecting portions of that expanded area for display are performed by relays controlled by the EXPANSION CONTROL switch, control panel wiring, and the off-centering pushbuttons. If either the CNDT or NORM position of the switch selects a display other than x1, the selection of the portion of the subsector

area to be displayed is determined by control panel wiring. The off-centering pushbuttons are effective only when the switch is in the EXPD position and they then select only portions within the area previously selected by the control panel wiring for NORM display. The off-centering pushbuttons thus allow a console operator to examine each expanded block within the NORM display area of that console, if this should prove necessary. An understanding of the means by which expansion and off-centering are accomplished digitally requires some discussion of the addressing of the display area by the Central Computer System and of the interpretation of these addresses within a display console. It should be noted that expansion of a display changes the apparent distance between different messages without changing character sizes or vector lengths.

2.4.4 Message Position Addressing

The Central Computer System designates a point within the subsector area by a rectangular co-ordinate pair. The X co-ordinate in this is presented to the Display System in binary form. If the point to be designated is to the right of the subsector reference point, its X co-ordinate is positive (sign bit = 0); if the point is to the left of the reference point, its X co-ordinate is negative and in complement form (sign bit = 1, magnitude bits not directly interpretable). The leftmost X co-ordi-

nate has the value 1.000 000 000 (X > decimal -1): the rightmost X co-ordinate has the value 0.111 111 111 (X < decimal +1). There are, then, two sets of X co-ordinates, the positive set and the negative set in complement form. Each set of X co-ordinates increases in actual magnitude from the center toward the extremes. However, if the sign bits of the two sets are ignored, the represented magnitudes of both sets of X co-ordinates increase from left to right (000 000 000 to 111 111 111). (See fig. 1-158, part A.) The sign bit of a given X co-ordinate indicates only the set (positive or negative) into which the X co-ordinate falls.

If the negative set of X co-ordinates is considered the lesser set and the positive set considered the greater, the sign bit of an X co-ordinate can be reversed in sense (complemented) in the display console and then treated as the most significant bit of the X co-ordinate. A 1 sign bit can be interpreted as a 0 and cause positioning to begin from the leftmost position of the display tube face; a 0 sign bit can be interpreted as a 1 and cause positioning to begin from the center of the tube face. By comparison, the first magnitude bit of the X co-ordinate will produce no change in position if it is 0 but will cause a shift to the right of one-quarter of the tube width if it is 1. Thus, the sign bit can cause a displacement across half the tube face by changing from 1 to 0, whereas the first magnitude bit can cause a displacement across one quarter of the tube face by changing from 0 to 1. In effect, reversing the sense of the sign bit of a subsector area address co-ordinate converts the co-ordinate from the two sets of negative and positive values into a single continuous range of values varying between 0 000 000 000 and 1 111 111 111 (between decimal 0 and decimal 1 if a binary point is inserted before the first binary bit). (See fig. 1-158, part B.)

The discussion of message position addressing has considered only the case of an x1 display. If the display is to be expanded to x2, the original sign bit of each co-ordinate is not used for message positioning. Instead, the first magnitude bit assumes the function of displacing the message across half the tube face, but this bit must also be reversed in sense if the displayed area is to be centered on the subsector reference point (no off-centering). The range of co-ordinate values between $-1/2$ and $+1/2$ is now translated into the range of display tube addresses between 0 and 1. The sign bit (together with the first magnitude bit) functions only to suppress display of those messages which are not within the selected portion of the subsector area but which are positioned on the tube face by the less significant bits of their address co-ordinates.

If an x2 display is to be off-centered, the sign bit is still not used for message positioning but only for

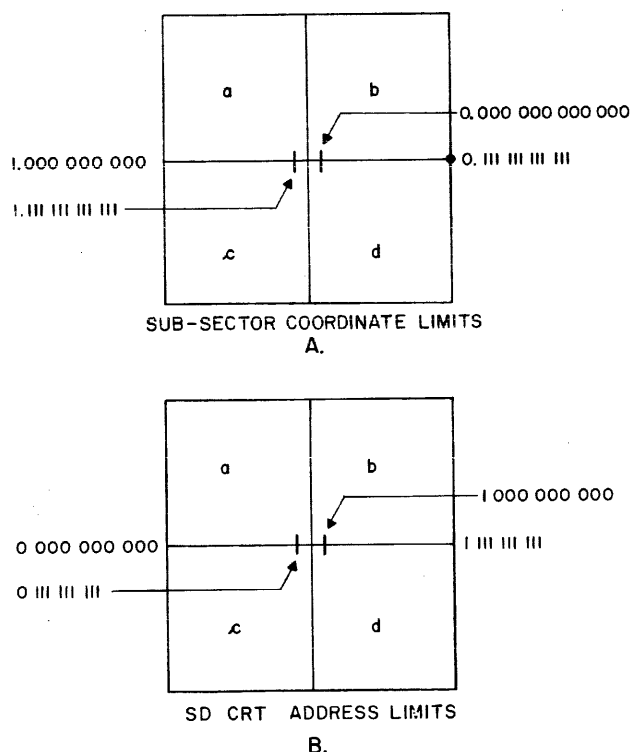


Figure 1-158. SD CRT and Subsector Co-ordinate Relations

suppression of intensification of incorrectly positioned messages. The off-centering is accomplished by not reversing the sense of the first magnitude bit (thus superimposing negative and positive sets of co-ordinates on each other) and by selecting for intensification only those messages whose sign bit and first magnitude bit indicate that they fall within the portion selected for display. The Central Computer System co-ordinates ranging in value from 1.000 000 000 to 1.111 111 111 and those co-ordinates ranging in value from 0.000 000 000 to 0.111 111 111 are both translated to correspond with the range of display tube addresses from 0 000 000 000 to 1 111 111 111; a message with X co-ordinate 1.001 100 110 and a message with the X co-ordinate 0.001 100 110 produce the same horizontal positioning on the display tube but only one is intensified in accordance with the selection of the negative or positive set of co-ordinates for display. The same reasoning applies in selection of a portion of any expanded display; the co-ordinates within that portion are translated into the range between decimal 0 and 1 for presentation to the X, Y decoders in the display console. The most significant bits not presented to the decoders and the first magnitude bit presented to the decoders are combined into an XY intensity signal to intensify only those messages which fall within the selected portion of the subsector area.

2.5 SITUATION DISPLAY CAMERA

The situation display (SD) camera provides a facility for photographing situation displays for later analysis or for use in training operations. (See fig. 1-159.) The SD camera is capable of photographing two independently selected sets of situation display data, called mode 1 and mode 2. The messages to be displayed in each mode are selected (by category and/or DAB) by switch settings on the SD camera console. Once these selections are made, a photograph of either mode 1 or mode 2 may be initiated by an *Operate* (SD camera) instruction or by pressing one of the pushbuttons controlling SD camera operation.

When the SD camera is program operated, *PER* (31)_s causes the camera to photograph mode 1; *PER* (32)_s causes the camera to photograph mode 2. An indication of the selected mode is given by a panel light on the console. An indication is also given if the selection of mode 1 or mode 2 is made manually.

When SD camera operation is initiated either by the programmed instruction or by pushbutton, exposure of the photograph does not begin until either a start-RD (radar data) pulse or a start-TD (track data) pulse is received. Actually, the camera shutter is opened when operation is initiated, but intensification of displays is

suppressed until one of the previously mentioned pulses is received. Intensification is applied only for the messages to be displayed within the mode selected. Upon receipt of the third start-RD or start-TD pulse, intensification is removed, ending the exposure. The camera shutter is then closed and the film automatically advanced to the next frame. The time required for one photograph varies between 2.8 and 4.3 seconds, depending upon the length of delay before a start-RD or start-TD pulse is received; i.e., 2.5 seconds for exposure of a complete display cycle, 0.3 second to index the film, and up to 1.5 seconds delay before starting exposure.

A separate instruction must normally be given to initiate the taking of each photograph. If a second photograph request is given while the SD camera is taking a picture, that request is held in a storage circuit provided in SD camera control until the first operation is completed. A third request arriving before the second is executed by simply replacing the stored second request which is lost. A request for photograph made by a pushbutton bypasses any stored program request and is executed prior to it.

Provision is made via a switch on the console to allow taking a multiple exposure on one film frame. The switch, SUPPRESS AUTOMATIC INDEX, prevents the film from advancing after each exposure and thus superimposes several exposures on one photograph.

The camera will operate normally as long as the film magazine contains more than five feet of film. If it contains five feet or less, a buzzer is energized and a lamp lighted. A pushbutton is provided adjoining the buzzer and light. Depressing the pushbutton locks the buzzer out of the alarm circuit for the duration of the alarm but leaves the lamp on. When the film magazine is reloaded, the alarm is de-energized and both lamp and switch are reset.

2.6 LARGE BOARD DISPLAY ELEMENT

A method of displaying a large scale image of a situation display is required in the command post of AN/FSQ-7 Combat Direction Central to present a summary of the air defense situation and two status totes to the subsector commander and his staff. The tote-board information changes very slowly and is manually projected on the status boards provided for this purpose. The air defense situation, consisting of different situation displays, must be continually redisplayed because of its continuously changing nature. To do this, the large board display element contains the equipment necessary to project a photograph on a special 5-inch situation display tube. This equipment consists of a camera projector monitor (a display console with a 5-inch situation display

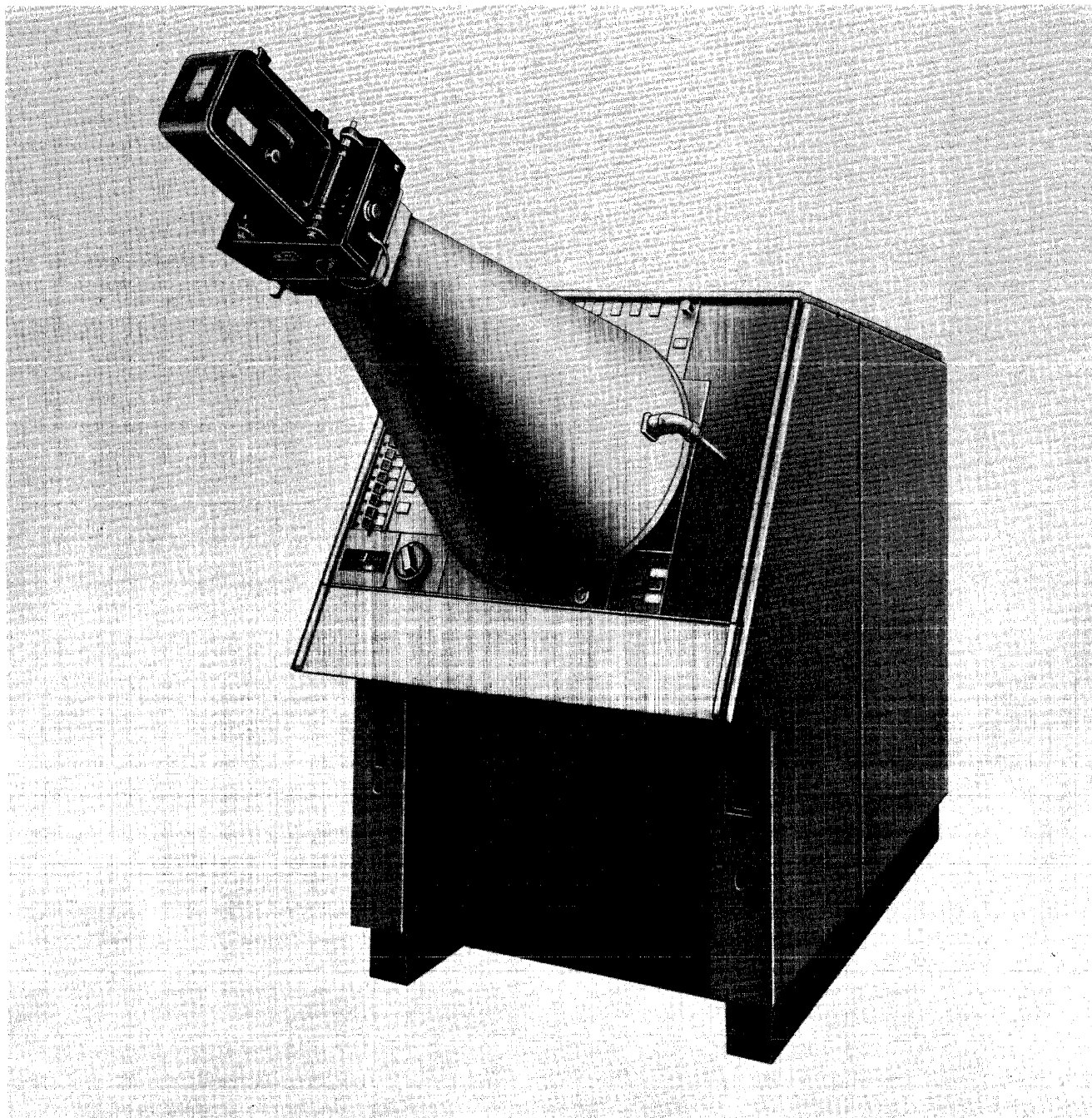


Figure 1-159. Situation Display Camera

tube) and a rapid processing photographic projection system.

The camera projection monitor provides a display like that of any other situation display console. The projection equipment automatically photographs this display, processes the film to provide a positive transparency, and projects an image of the display on a screen in the Command Post. Projection occurs with a delay no

greater than 30 seconds plus one display cycle (2.5 seconds) after the 5-inch situation display tube has been intensified.

Two camera projector monitors and two rapid processing photographic projector systems are provided in AN/FSQ-7 Combat Direction Central. One of these two units is in operation at any one time; the other is used as standby.

SECTION 3

DIGITAL DISPLAY SUBSYSTEM OPERATION

3.1 INTRODUCTION

The digital display subsystem accepts information from the Drum System in digital form and presents this information in visible form at various display consoles. The Drum System acts as a time buffer between the rapidly operating Central Computer System and the slower digital display subsystem (which takes an appreciable time to generate as many as 10,240 characters that may be required for one complete digital display). The cathode-ray tube on which digital information is displayed has already been described. (Refer to 1.3.3.) What remains for discussion is the form in which information is presented to the digital display subsystem and the operations necessary to convert that information into the form required by the digital display tube.

Unlike the situation display subsystem which produces a new display every 2.54 seconds, the digital display subsystem produces a new display only when ordered to do so by the program of the Central Computer System. (Refer to table 1-69.) The Central Computer System prepares the information for digital display and places it on the digital display field of the MIXD drum. The display of this information is initiated by the *Operate* (start digital display) instruction, *PER* (35)₈. The information, once written on the digital display tubes, is continually displayed on those tubes until new information is written onto them by another digital display cycle.

3.2 DIGITAL DATA MESSAGES

3.2.1 General

Digital data messages are presented to console operators on 5-inch digital display tubes. The term message, as used in the digital display subsystem, describes the entire block of information to be displayed on a single digital display tube. (In contrast to this usage, a single situation display message contains only one of the many items of information displayed on a single situation display tube.) Each message on the digital data drum field contains all the information intended for display by the digital display tube to which it is addressed.

3.2.2 Message Contents

A digital data message contains information which will produce displays of from 4 to 32 rows of characters, each row containing 5 characters. (See fig. 1-160.) The five characters in each row are designated by letters

H through L, to distinguish them from features A through G of track data messages. A message may contain as few as 4 rows of characters or as many as 32 rows, displayed in 2 columns of 16 rows each with some rows possible blank. The length of each message is determined by the amount of information that must be displayed by the digital display tube receiving that message.

For each row of a message, five characters must be selected and positioned. Further, successive rows must be positioned with respect to each other on the tube face. To select a given character on the character-forming matrix, the matrix address must be given. Each character on the eight-by-eight matrix can be addressed with two octonary numbers representing two character selection co-ordinates, x and y. (See fig. 1-161.) Since an octonary number can be written in three binary bits, each selection requires six binary bits. With five characters per row, a digital data message must contain 30 character selection bits for a complete row. The positioning of the characters within the row is inherent in the placement of their character selection co-ordinates within a message word.

With 30 bits required for the selection of 5 characters in a row, a digital data message can contain that information plus 2 control bits in one 32-bit word for each row. These control bits perform the general function of positioning a row on the tube face and the function of switching the writing process from one digital display tube to the next.

TABLE 1-74. CONTROL BIT FUNCTIONS,
DIGITAL DATA MESSAGES

POSITION AND CONTENTS		MEANING
LS	RS	
0	1	Start of new message. Display this word on upper left row of next indicator section.
0	0	Display this word immediately below the preceding row displayed.
1	1	Skip a row between the preceding row and the row in which this word is displayed.
1	0	Display this word on the upper row on the opposite side of the same indicator section.

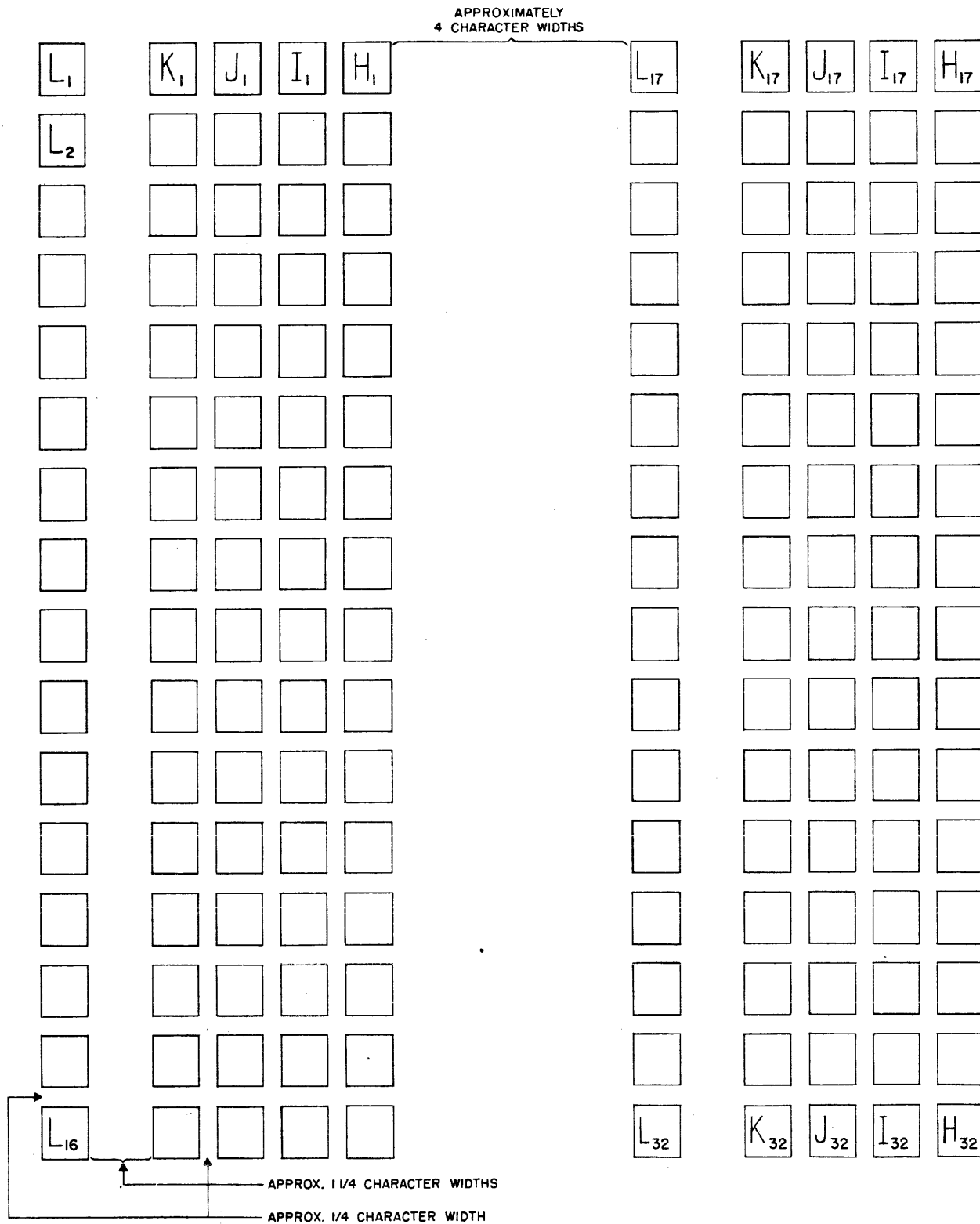
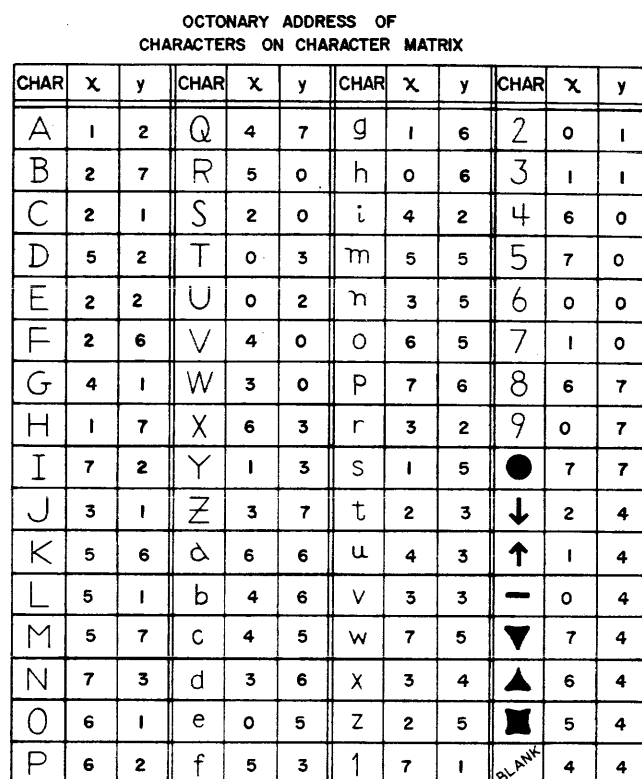


Figure 1-160. Digital Display Message Format



WORD	BIT POSITION																															
	LS	LI	L2	L3	L4	L5	L6	L7	L8	L9	LI0	LI1	LI2	LI3	LI4	LI5	RS	RI	R2	R3	R4	R5	R6	R7	R8	R9	RI0	RI1	RI2	RI3	RI4	RI5
1	*	L ₁			K ₁			J ₁			I ₁			H ₁			*	L ₁			K ₁			J ₁			I ₁			H ₁		
		x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰		y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰
2	*	L ₂			K ₂			J ₂			I ₂			H ₂			*	L ₂			K ₂			J ₂			I ₂			H ₂		
		x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰		y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰
3	*	L ₃			K ₃			J ₃			I ₃			H ₃			*	L ₃			K ₃			J ₃			I ₃			H ₃		
		x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰		y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰
4	*	L ₄			K ₄			J ₄			I ₄			H ₄			*	L ₄			K ₄			J ₄			I ₄			H ₄		
		x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰		y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰
32	*	L ₃₂			K ₃₂			J ₃₂			I ₃₂			H ₃₂			*	L ₃₂			K ₃₂			J ₃₂			I ₃₂			H ₃₂		
		x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰	x ₂ ²	x ₂ ¹	x ₂ ⁰		y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰	y ₂ ²	y ₂ ¹	y ₂ ⁰

POSITION AND CONTENTS		MEANING
LS	RS	
0	1	START OF NEW MESSAGE
0	0	DISPLAY THIS ROW IMMEDIATELY BELOW PRECEDING ROW DISPLAYED
1	1	SKIP A ROW BETWEEN THE PRECEDING ROW AND THE ROW ON WHICH THIS WORD IS TO BE DISPLAYED
1	0	DISPLAY THIS WORD ON THE TOP ROW ON THE OPPOSITE SIDE OF THE SAME INDICATOR SECTION

229

3.2.3 Drum Layout

The words of a digital data message are laid out as shown in figure 1-162, although successive words are placed in every 64th register on the digital display field. Each word controls the display of one row of five characters on a digital display tube. Considering only one word (since the layout of all words is identical), the x selection co-ordinates for each character are contained in bits L1 through L15; the y selection co-ordinates for each character are contained in bits R1 through R15. The two control bits are contained in the left sign (LS) and right sign (RS) positions in the word. Their functions are given in table 1-74.

3.3 DIGITAL DISPLAY GENERATOR ELEMENT

3.3.1 Element Function

The digital display generator element (DDGE) consists of the equipment necessary to convert each digital data message received from the Drum System into the form required by the digital display tubes. Functionally, it consists of the following steps:

- Converts all x and y character selection co-ordinates, presented in digital form, into character selection analog voltages for presentation to the digital display indicator element.
- Generates character positioning analog voltages (on the basis of the order of display of each character within a message word and on the basis of the positioning bits contained in that message word) and supplies them to the digital display indicator element.
- Produces erase gates (which remove old messages from digital display tubes just before their new messages are written onto them), indicator selection signals (which provide for the writing of each message onto the proper digital display tube), and intensification signals (which allow message display), as determined by the control bits in each word of a digital data message.
- Produces contrast signals to improve the visual quality of displays on tubes.

The digital display generator element also has the overall function of timing the operation of the digital display subsystem and of routing messages to the correct sections of the digital display indicator element.

3.3.2 Digital Display Cycle

The operation of the digital display generator element is largely determined by the timing and sequence of reading digital data messages from the digital display field of the MIXD drum. This sequence is identified as a digital display cycle. Since the digital display indicator sections are grouped together, there is only one display cycle to be considered. During the display cycle, all of

the indicator sections receive data from the generator element. The display cycle is initiated by the *Operate* (start digital display) instruction, *PER* (35)_s. Information once written on a digital display tube will continue to be displayed until it is erased and rewritten during a succeeding display cycle because of the storage properties of the digital display tube. The digital display cycle can be nonperiodic; that is, occurring at irregular intervals determined by the availability of new information.

Words are read from the drum at 640-microsecond intervals. This time interval is provided by reading the digital data drum field by precession. The precession pattern used in reading the digital display field of the MIXD drum involves skipping 63 registers between register readings. Each register contains one word of a digital data message. Successive words of a digital data message are thus delivered to the digital display generator element at intervals of 640 microseconds. This precession pattern is incompatible with the interleave of 64 by which the Central Computer System can write on the digital display field.

Since a digital data message may be from 4 to 32 words long, no specific time can be given for the display of a single message. However, an entire digital display cycle, the time required to read and display the contents of the entire field, occupies 1.34 seconds. A total of 64 drum revolutions, each taking approximately 20 milliseconds, plus 60 milliseconds switching time between drum fields, is needed to read the entire field. To allow the reading of successive words of one message by precession, the words of each message must appear on the drum field with an interleave of 64. (This is in contrast to track data messages where the registers containing one message are contiguous.)

TABLE 1-75. DIGITAL DISPLAY CYCLE TIMING

OPERATION	TIME (MICROSECONDS)	TOTAL TIME
Erase time, each tube		200 milliseconds
Read and display one word	640	
Select and position first character	35	
Select and position each succeeding character	20	
Intensify each character	90	
Switch between words	10	
Read and display a 4-word message		2.56 milliseconds
Read and display a 32-word message		20.48 milliseconds
Full digital display cycle		1.34 seconds

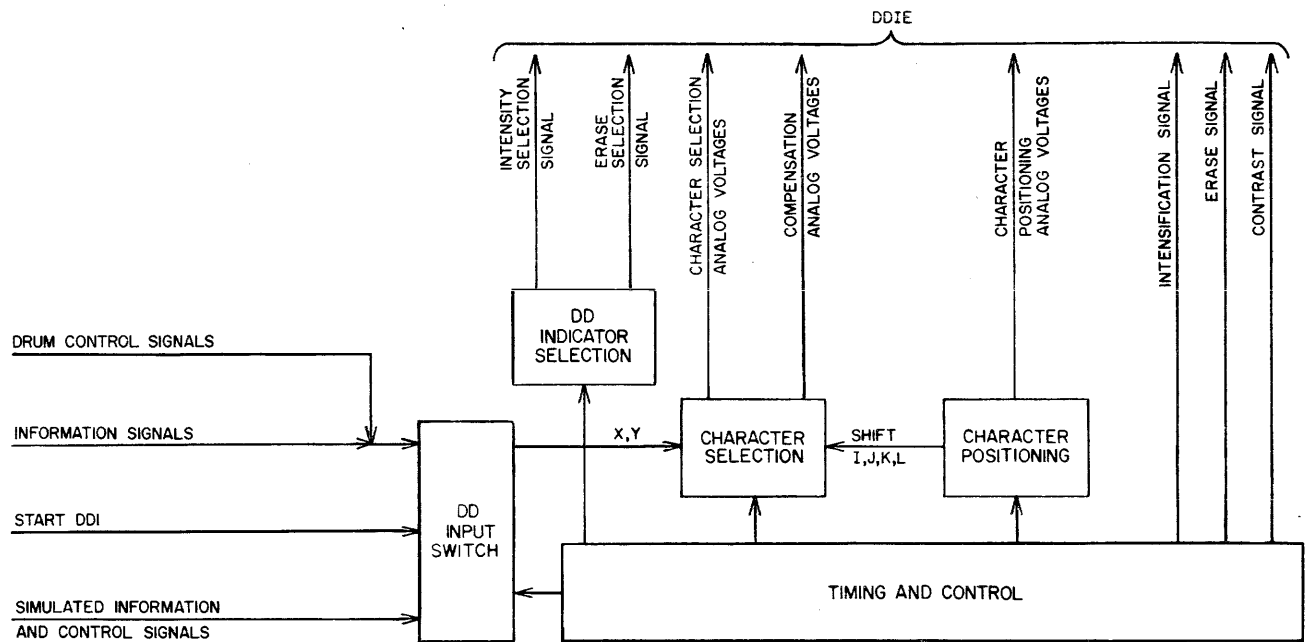


Figure 1-163. Digital Display Generator Element

The flow of information during a digital display cycle is then as follows. The cycle is initiated by the Central Computer System *Operate* instruction $PER (35)_8$ and begins when the first word of the first digital data message is in reading position. As many as 2,048 words may be presented to the digital display generator element at 640-microsecond intervals, over a period of 1.34 seconds. The cycle is then complete, and is not repeated until another $PER (35)_8$ instruction is executed by the Central Computer System. Table 1-75 summarizes the timing of operations within a digital display cycle.

3.3.3 Element Operation

The digital display generator element (DDGE) stores, processes, and distributes all digital display information. (See fig. 1-163.) The information normally supplied to the DDGE from the Drum System is of two types, control signals and display information signals. A third type of signal initiate-digital-display comes from the Central Computer System.

The information signals are carried on 32 lines to the input switch section. These lines carry the bits of each message word being read off the drum. With only one word read at a time, 32 lines suffice for word transfer. The control signals are carried over four lines.

Both the information signals and the control signals go through the input switch section. The function of this section is to operate in conjunction with the display tester element in substituting test signals from that element for the normal signals from the drums. The character selection section stores the x and y co-

ordinates of the five characters comprising one word of a digital display message and generates analog voltages for each character of the word in sequence. So that each selected character may be positioned properly, special positioning voltages must be provided. These voltages are supplied by the X and Y counters in the character position section. The character position section also provides shifting signals to the character selection section; the signals shift the character selection coordinates of one character at a time from storage to the analog conversion circuits. After the characters have been selected, they must be displayed on a digital display tube as a portion of a message.

Provisions must be made to display each message at the appropriate indicator section. The slot counter located in the indicator selection section directs each digital data message to one indicator section and erases that section before a new message is written onto it. The slot counter of the indicator selection section produces indicator selection signals which are changed by stepping the slot counter at the start of each message. Slot-numbered lines, carrying sexadecimally coded signals representing the contents of the slot counter, are distributed to the indicator sections. Each line allows writing of one section and erasure of another. The slot counter may count up to 127; at present, only the counts from 0 through 106 are used. The operation of the slot counter is best described by following it through a digital display cycle.

Execution of the *Operate* (start digital display) instruction, $PER (35)_8$, by the Central Computer System initiates a digital display cycle by clearing the slot coun-

ter. After an initial delay period of approximately 200 milliseconds, the slot counter is stepped from 0 to 1 by the reading of the first word of the first message. This word contains control bits which indicate a new message and cause the slot counter to be stepped. (Refer to table 1-74.) This first message is written on the indicator section selected by slot line 1. (This indicator section is erased during the initial delay period.) As the first word of the second message is read from the drum, the control bits in that word again step the slot counter, thus directing the second message to the indicator section selected by slot line 2. This process continues through the reading of the last message intended for display (at present, message number 105). A final word containing only the control-bit combination for stepping the slot counter is read from the drum. This last word steps the slot counter to 106, ending the digital display cycle.

It is necessary to erase each indicator section prior to writing on it. The time lapse between erasing and writing of each section is that introduced at the beginning of the cycle, or approximately 200 milliseconds. To maintain this time delay, the slot line which allows writing on a given indicator section is used to initiate erasure of the indicator section to be written 200 milliseconds later. However, because the amount of registers

per slot varies, the number of slots between writing and erasing of different sections will also vary. For this reason, slot counts are not used to introduce the time delay; instead, the reading of a fixed number of registers from the DD field is used for that purpose. Since 640 microseconds are required to read a register, approximately 320 registers are read between the erasing and writing of a given section.

The operation of the X and Y counters which position each character within a message is related to the operation of the slot counter. The slot counter is stepped by the control bits within the first word of a message. The five characters designated by that word must be displayed on the left half of the top line of the selected indicator section. In order to do this, the Y counter is cleared by the signal which steps the slot counter, thus selecting the top line of the digital display tube. The X counter is also cleared by the same signal, positioning the H character of the first word just to the left of the tube center. (See fig. 1-160.) On the next character signal from the timing and control section, the Y counter is still clear but the X counter is stepped by 1. Generation of the I character selection analog voltages then begins. The I character is positioned just to the left of the H character. The space between K and L characters is generated in a decoder which converts the X counter contents into analog voltages.

If the next word of the message is to be displayed immediately below the first line (as indicated by the control bits of the word), the timing and control section steps the Y counter once. A space can be skipped between lines (by the control bits) by stepping the Y counter twice.

For lines on the right half of the digital display tube, the X counter is set by an other-side signal from the timing and control section as determined by the control bits of the digital display message word. The other-side signal complements a flip-flop that controls which side of the display tube will be written. Therefore, a second other-side signal will cause writing on the left side again, but will also clear the Y counter and cause writing on the top line of the left side.

Besides the signals mentioned above, several others are sent by the digital display generator element to the indicator sections. These signals are:

- a. An intensification gate accompanying the selection of each character in a digital data message.
- b. An erase gate accompanying each message presented for display in the indicator sections.
- c. A contrast gate applied to each indicator section at 640-microsecond intervals to improve the legibility of previously written characters by reilluminating the phosphors on the tube faces.

INFORMATION NOT AVAILABLE

Figure 1-164. Auxiliary Console

All DDGE output signals except the slot line signals from the indicator selection section are applied simultaneously to all sections of the indicator element. Thus, although every character of every message is potentially capable of display on every indicator section, the slot line signals from the indicator selection section allow the display of a message only on the indicator section for which it is intended. More precisely, the slot line signals allow the erasure of only one indicator and the writing of only one other indicator at a time.

3.4 DIGITAL DISPLAY INDICATOR ELEMENT

The digital display indicator element consists of the indicator sections necessary to display the digital display information. These sections may be located on the situation display consoles, as shown in figure 1-157, or on auxiliary consoles, as shown in figure 1-164. Each section of the digital display indicator element receives the outputs of the digital display generator element and generates a new display as directed by slot line

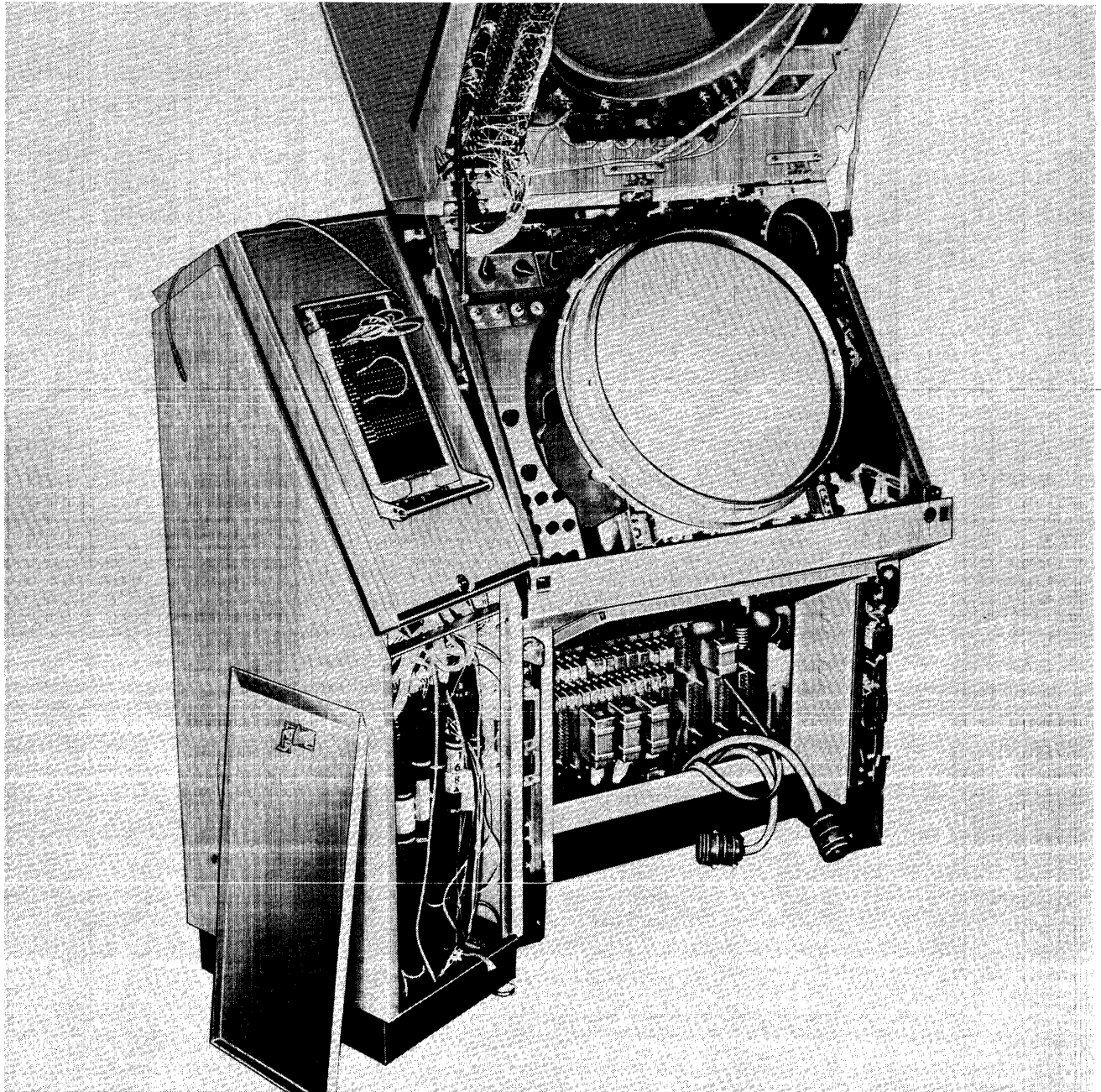


Figure 1-165. Maintenance Display Console

signals. Since all sections are identical functionally, a description of only one will suffice.

The character selection analog voltages are applied to both the character selection plates and the compensation plates of the digital display tube. Similarly, the character positioning analog voltages are applied to the character positioning plates of the digital display tube. With these two sets of voltages applied, the characters of a message will be selected and positioned. However, no display occurs unless the tube is unblanked by an intensification gate.

Prior to the writing of a message in a section, the section must be erased. Assume that indicator section 18 is being discussed. While an indicator section 320 registers ahead is written, the slot line of that section is also applied to the erase circuit in indicator section 18. This slot line signal, together with the erase gate from the DDGE, causes the erase unit in indicator section 18 to generate a digital display erase gate. The storage mesh of the digital display tube is erased, rendering it completely opaque to the flood-gun beam and making it ready to receive a new message.

The contrast gate is applied every 640 microseconds to the storage mesh of each tube to maintain the contrast between written and unwritten portions of the tube face. In effect, the written portions of the tube phosphor are re-energized by the flood-gun more intensively while the contrast gate is applied.

3.5 MAINTENANCE DISPLAY CONSOLE

The maintenance display console, shown in figure 1-165, allows visual checking of the operation of the SDGE and the DDGE or of an individual display console. One maintenance display console is provided for each SDGE-DDGE pair within the duplexed portion of the Display System. The maintenance display console provides for visual monitoring of all outputs of the two generator elements to which it is connected. Further, since the maintenance display console can be set up to duplicate the display that should be presented at another console, the maintenance display console can be used to check the operation of the second console through a visual comparison of the two displays.

The maintenance display console differs from other display consoles in two respects. Each maintenance display console is permanently connected to one pair of generator elements (SDGE and DDGE) and cannot be switched to receive signals from the other pair. The selection of messages for display at the maintenance

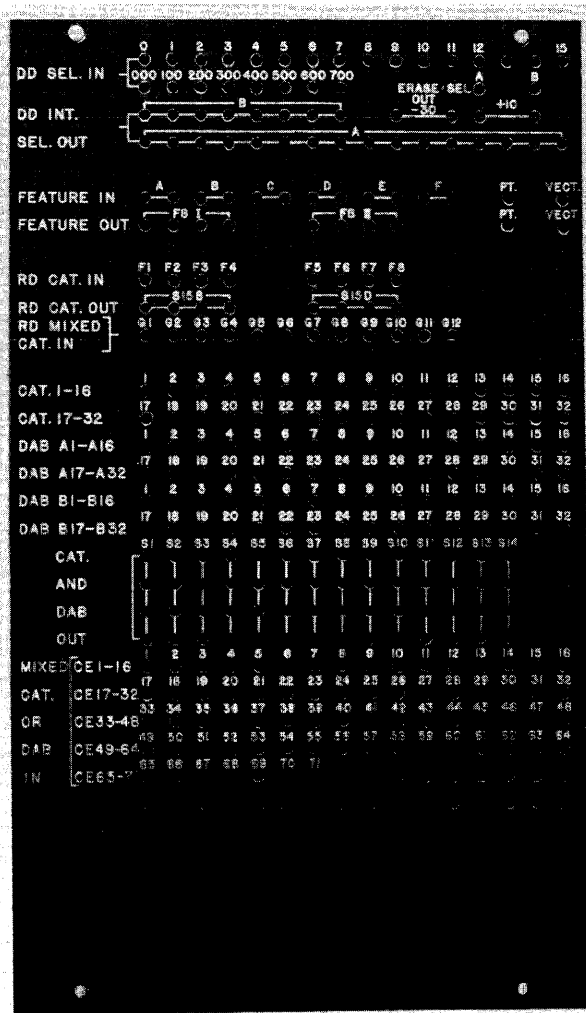


Figure 1-166. Maintenance Display Console Input Signal Control Panel

display console is made by plug connections on the control panel mounted adjacent to the console (instead of by internal console wiring). (See fig. 1-166.) The control panel provides complete flexibility in connecting slot line signals to the erase and intensify control circuits within the digital display indicator section and in connecting category and DAB lines, as well as feature selection lines to the situation display indicator section within the maintenance display console. This complete flexibility allows the setting up of the maintenance display console to duplicate the display of any other display console.

SECTION 4

INFORMATION FLOW

4.1 INTRODUCTION

The Display System exists within AN/FSQ-7 Combat Direction Central to transform data from the form in which it is prepared by the Central Computer System into visible form for display to personnel directing the defense of an area against air attack. The information to be displayed must be prepared for delivery to the Display System in the forms described in Sections 2 and 3. Radar data messages require very little special handling, since they contain only the XY co-ordinates of a radar report and an indication of the source (IFF or not) and the status (correlated or not) of that report. The information to be presented in a track data message or a digital data message requires the translation of certain information into character selection co-ordinates and the proper placement of those co-ordinates within each message. In addition, the placement of these messages on their drum fields requires some special consideration within the program for the Central Computer System.

All those fields of the Drum System through which information is delivered are written by the Central Computer System by the address mode. (Refer to Chapter 4, 2.3.1.) This provision enables the Central Computer System to reobtain specific items of information without interfering with their delivery to the Display System. Thus, those fields of the Drum System on which display information is placed serve both as time buffers between the Central Computer System and the Display System and as catalogued storage space for the Central Computer System.

The advantages of the dual function of display drum fields can be demonstrated as follows. The X , Y and \dot{X} , \dot{Y} components of each track data message are used by the Central Computer System in checking for correlation of radar reports with existing tracks. This information can be read from the TD drum into the Central Computer System as correlation is being performed. (Correlation of a radar report requires comparison of that report position to the predicted positions of all nearby tracks to determine which track, if any, the report belongs to.) If a radar report is not correlated with any track, a new track may be initiated if this report is the first on an aircraft within the subsector area. If the report does not correlate with a track, the track position and velocity must be corrected on the basis of the new information represented by the report.

The new X , Y and \dot{X} , \dot{Y} , co-ordinates for that track can then be placed on the TD drum, replacing the old co-ordinates without the necessity of regenerating the entire track data message.

Similar reasoning applies to the digital display field. The Central Computer System need only change those messages on the digital display field whose information content requires changing. If the location of each message (which is displayed at a particular console) on the digital display field is established in advance of operation, a single message can be changed if that is all that is required. Further, it is possible to have the Central Computer System check the data on the digital display field against its latest data to determine the need for changing the digital data on the field.

The information presented at a display console enables the operator at that console to make tactical decisions on the basis of all pertinent information collected by the Central Computer System. The operator at the console communicates these decisions, or requests for related information, to the Central Computer System via the manual input element of the Input System. (Refer to Chapter 5, Section 4.) The Central Computer System can thus present all data pertinent to a given decision at a given display console, then implement the decision indicated by the console operator via the manual input element. (See fig. 1-167.) The decision-making function must be left to operators, since a digital computer operates solely on a yes-no basis, with no ability to make a decision on incomplete data. However, the decisions made by an operator are checked by the Central Computer System as to validity and possibility of execution.

4.2 RELATION TO MANUAL DATA INPUT ELEMENT

4.2.1 Situation Display Subsystem

4.2.1.1 General

An operator at a situation display console may communicate a decision, a request for other information, or some information to be associated with a given situation display message via the manual data input keyboard mounted on the console. A number of prearranged actions are written into the program of the Central Computer System. Each action is performed only if called for by a manual input data selection panel message. Since the function of each console and the programmed actions required by that function can be predetermined,

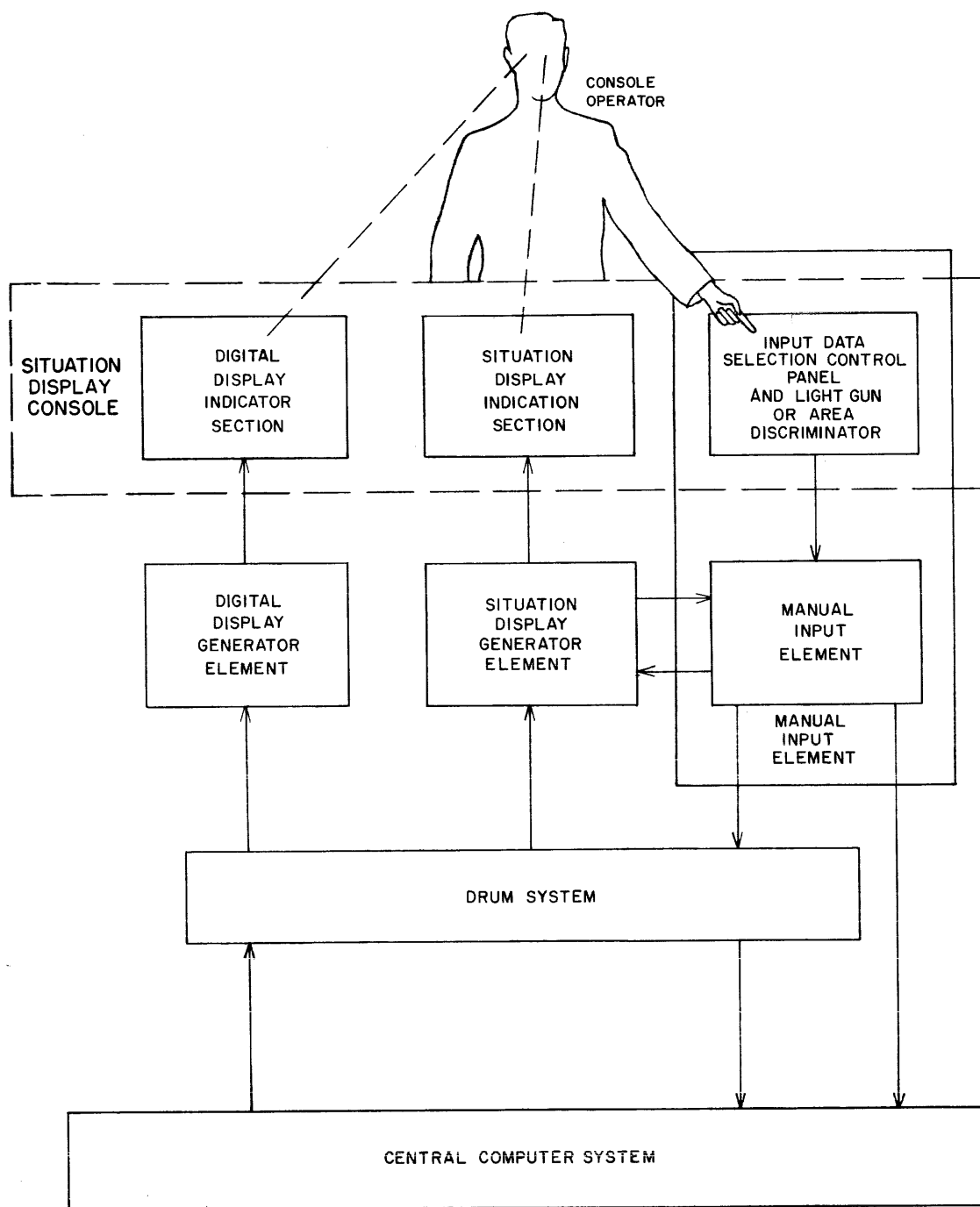


Figure 1-167. Information Return From Display System via Manual Data Input Element

the keyboard message from that console need only indicate which of the possible actions is to be performed at any one time. Thus, the console operator can depress a particular pushbutton with the knowledge that although only one binary bit may be changed from 0 to 1, a relatively complicated message has been conveyed.

Since the Central Computer System may read a data selection panel message before the operator has

completed making it up, some provision must be made to prevent action on this incomplete message. This is done by requiring the console operator to indicate the completion of a keyboard message in one of two ways, by depressing the ACTION pushbutton on the keyboard as the last step in making up the message or by pulling the trigger on the light gun at the console. Either method sets a core in the manual input core array which indicates that the message from that key-

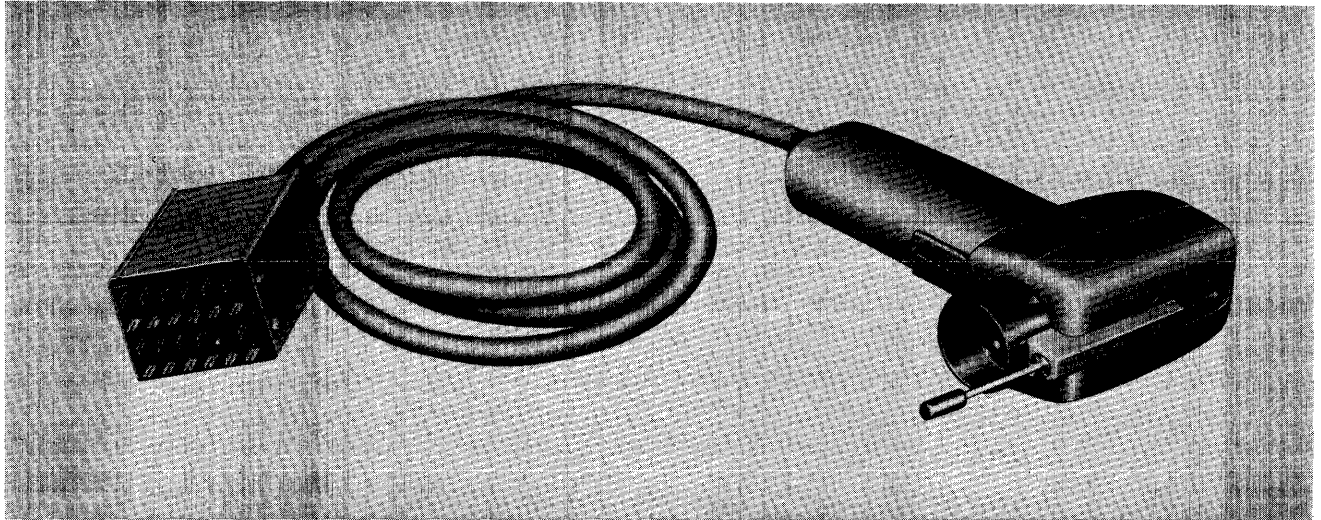


Figure 1-168. Light Gun

board is complete and may be acted on. The choice of method depends upon the type of message being sent via the keyboard. If the message requires no reference to a particular target the ACTION pushbutton is used. If the message must be related to a particular target, the light gun is used.

4.2.1.2 Light Gun Operation

If a console operator were required to identify a specific target to the Central Computer System by inserting an estimate of its X, Y co-ordinates or its track number (the C feature of a tabular track data message) on keyboard switches, errors in estimation or insertion might result. The light gun, a semiautomatic target designation device, eliminates this source of error. (See fig. 1-168.)

A console operator can specify some action on a selected target by setting up a keyboard message and then designating the target with the light gun at the console. The light gun is positioned so that the red aiming beam it projects just covers the present radar data message to be designated or the vector origin of the tabular track data message to be designated. The console operator then pulls the light gun trigger, which indicates keyboard message completion in the same manner as does depressing the ACTION pushbutton and intensifies all point features. When the selected target is written during a situation display cycle, the photomultiplier tube within the light gun detects the initial blue flash and generates a pulse coincident in time with the presence of identification information on that target in the situation display generator element (SDGE). The light gun pulse is applied to the SDGE as an information transfer signal, gating the target identification out of the SDGE through the manual input element onto the manual input drum field. The console operator is in-

formed that transfer has been initiated when the aiming beam on the light gun is extinguished and an indicator on its rear is lighted. The operator readies the light gun for another transfer by releasing the trigger on the light gun, restoring the aiming beam and extinguishing the rear indicator.

If the designated target is in radar data form, the manual data input element receives the X, Y co-ordinates in 11 bits each and 2 bits identifying the information as correlated or uncorrelated radar data. If a track data message is designated, the manual data input element receives the track number (six bits each for the first three alphabetical symbols and four bits for the final number) and two bits identifying the information as track data. The manual input element adds eight bits identifying the originating console to each target designation before placing it on the manual input drum field. The Central Computer System can thus relate a given target designation to the keyboard message directing the action to be taken on that target. It should be noted that transfer initiated by a light gun is possible only on track data messages containing the appropriate control bit. (Refer to table 1-72.)

4.2.1.3 Area Discriminator Operation

One type of Central Computer System operation obtains information from the Display System via the manual data input element without operator intervention. This operation is automatic track initiation. The required display information is obtained by the Central Computer System from an area discriminator, a large light gun covering an entire situation display tube face, which is triggered by an *Operate* instruction. (See fig. 1-169.)

Automatic initiation is performed in light air traffic areas since the decisions involved are purely routine

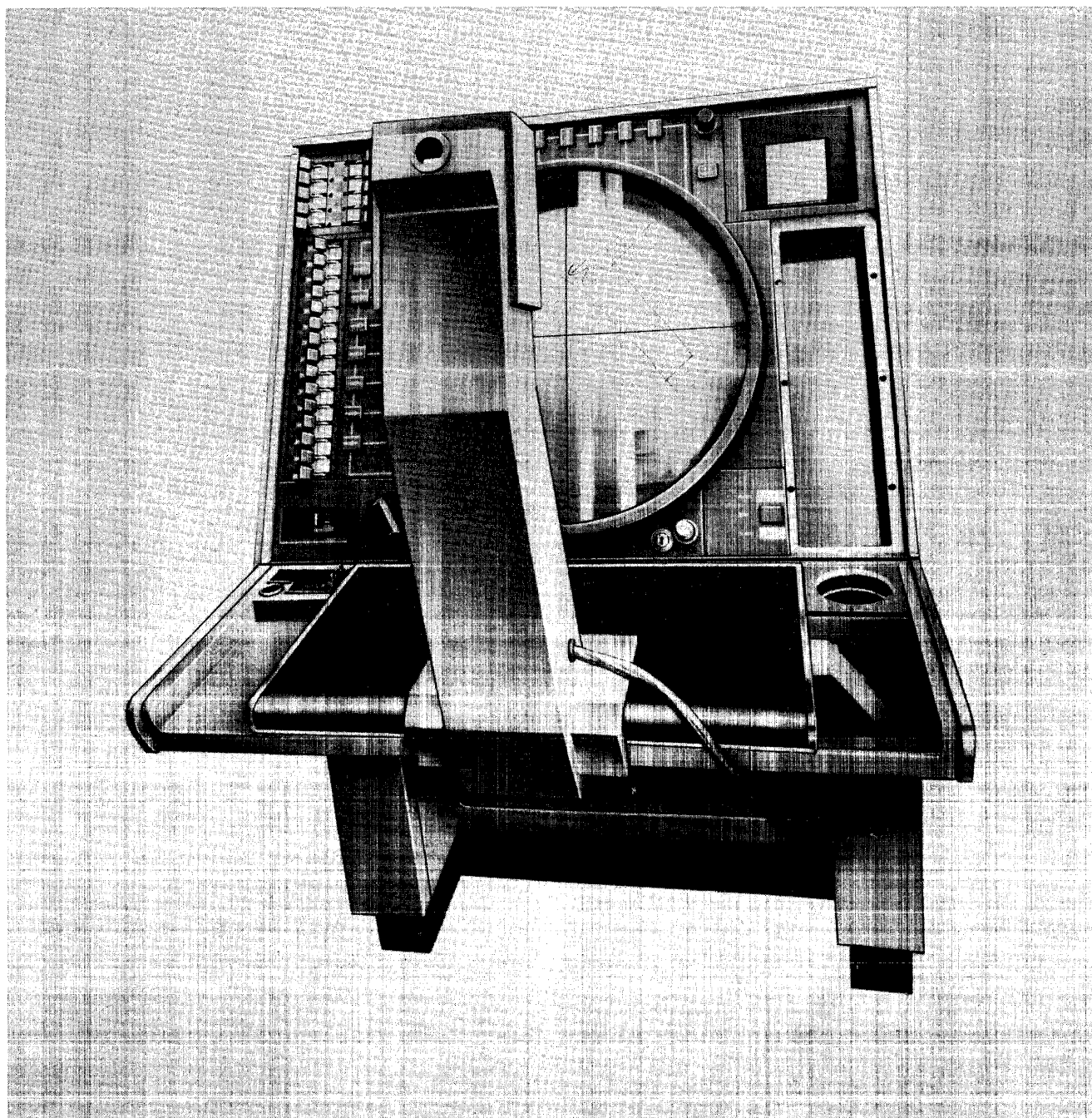


Figure 1-169. Area Discriminator Console

and efficiency of detection is high. The automatic initiation area discriminator is set to display uncorrelated present radar data. Those areas in which automatic initiation is not to be performed are masked out. When the *Operate* (area discriminator) instruction, PER (20)₈, is executed, the area discriminator transfers reports on all unmasked targets displayed during one situation display cycle. Transfer then ends until another PER (20)₈ instruction is executed.

4.2.2 Digital Display Subsystem

The only relation that exists between the manual data input element and the digital display subsystem is the use of the manual data input element by operators in requesting new or different information on their digital display tubes. An operator may send such a request via the manual input keyboard and ACTION push-button. When the request is received and processed by the Central Computer System, that system will initiate a new digital display cycle.

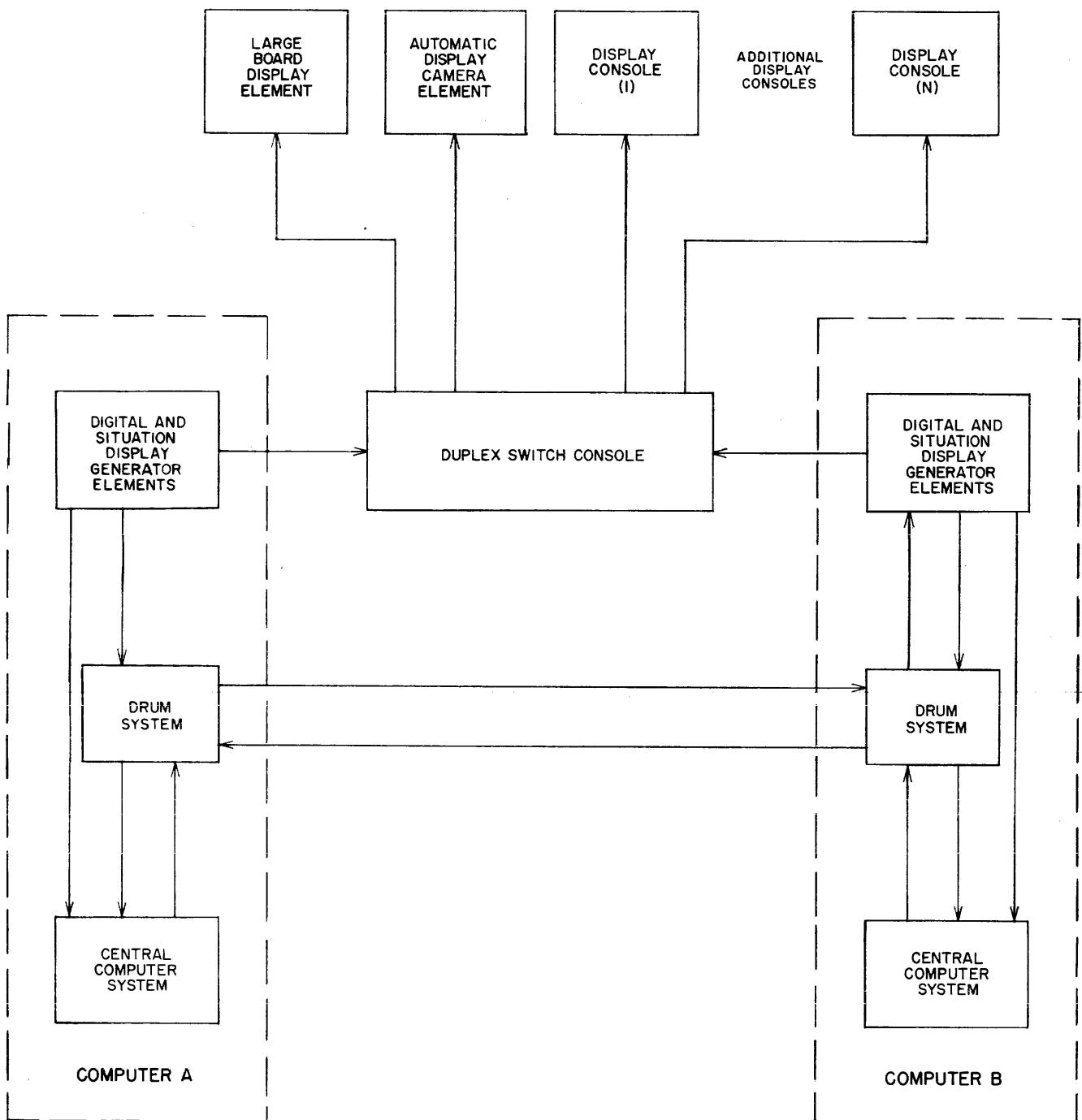


Figure 1-170. Display System Duplexed Information Flow

One further control available to the operator at a console appears to the left and on a line with the center of the digital display tube shown in figure 1-157. This control is an ERASE pushbutton which allows the operator to erase and blank out his digital display tube if the information on the displays is outdated or unnecessary. The tube will remain blank until rewritten during a digital display cycle. Thus, the operator can receive new information and then blank out the tube.

4.3 DUPLEXED INFORMATION FLOW

The fact that only the generator elements of the Display System (SDGE and DDGE) are duplexed necessitates some special switching arrangement between the generator elements and the display consoles. (See fig. 1-170.) These switching facilities provide for the distribution of signals from the pair of generator elements in the active computer (whether A or B) only to those display consoles whose unit status switches are in the

ACTIVE position. Similarly, the consoles whose unit status switches are in the STANDBY position receive signals from the generator elements in the standby computer (whether A or B). When computers A and B interchange their statuses in a switchover operation, the active consoles receive signals from the newly active computer and the standby consoles receive signals from the now standby computer; the console operators need not take any special action during switchover.

If circuit malfunctions are detected in an active generator element, a switchover operation will allow normal operations to continue while the defective generator element is being serviced. When a display console in active status shows symptoms of failure, it can be

switched to standby while servicing takes place. The operational function performed at that console can be taken over at another console similarly equipped and now switched to active. In general, a group of consoles will be provided to perform a particular operational function. Further, the group will include one console more than the minimum number of consoles required for that function. Normally, one console will be in standby status for maintenance operation or will be used as a spare for the group. The spare of standby status can be rotated among the consoles within the group to allow periodic maintenance operations on each console and to provide for approximately equal active running time on each display console.

CHAPTER 7

OUTPUT SYSTEM

7.1 SYSTEM DISCUSSION

7.1.1 System Function

The operation of AN/FSQ-7 Combat Direction Central in the processing of air defense data requires the transmission of processed information to such sites as airbases, airborne interceptors, antiaircraft artillery units, radar sites, and other centrals. The Central Computer System prepares this information and delivers it in parallel to the Drum System. The Output System obtains this information from the Drum System in parallel form and converts the information into serial form for transmission via telephone facilities to its ultimate destination. (See fig. 1-171.) The Central Computer System places output information on the output buffer (OB) fields of the LOG drum by status. The Output System reads this information from the OB fields by status identification.

Three types of output information are handled by the Output System: ground-to-air (G/A) information for transmission to airborne interceptors, ground-to-ground (G/G) information for transmission to other centrals or to height-finder radar sites, and teletype (TTY) information for transmission to nonautomatized centers. Ground-to-air information is transmitted via telephone facilities to radio transmission equipment for relay to airborne receivers; G/G information, by telephone facilities to receiving points; TTY information, as conventional teletype signals via telephone facilities. The Output System also supplies control information to the Central Computer System, allowing that system to preassign times of transmission of specific items of information.

7.1.2 System Logic

The Output System comprises two elements, the output control element and the output storage element.

(See fig. 1-172.) The output control element reads output drum words from the OB fields, performs certain checks on each word accepted, and directs the output information contained in each word to the appropriate section of the output storage element. When requested by the Central Computer System, the output control element also supplies control information to the Central Computer System.

The output storage element, since it handles three different types of information, is divided into three sections: G/A, G/G, and TTY. Each section receives output information presented to it by the output control element, performs further checks on the information, and converts the information into serial form for telephone transmission.

Since failure of any element or section of the Output System would seriously impair its functioning, the Output System is in duplex, one Output System within computer A and the other within computer B. (See fig. 1-173.) The Output System of the active computer feeds the telephone transmission equipment. The Output System of the standby computer is not used operationally, but may be used in certain test operations.

7.1.3 Information Forms

7.1.3.1 Bursts

Each of the three types of information processed by the Output System, G/A, G/G, and TTY, is transferred out of the Output System in blocks of fixed duration and content known as bursts. Within the interval known as a burst period, a single block of information is prepared and transmitted by each section. Bursts from each section are numbered in sequence. These burst numbers are used to control the sequence of transmission of information items and, in certain cases, can also be used to

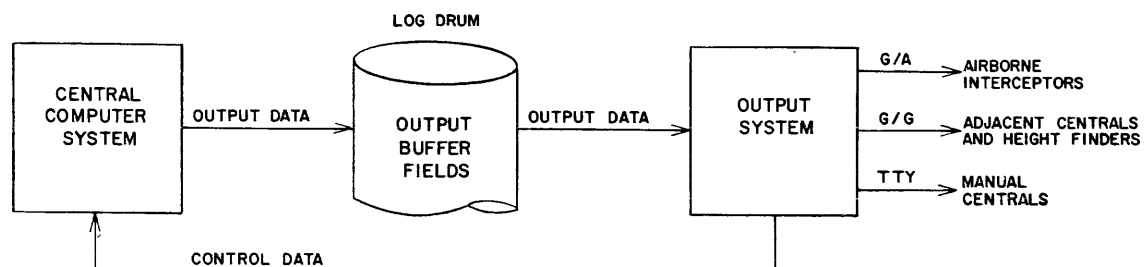


Figure 1-171. Output System Information Flow

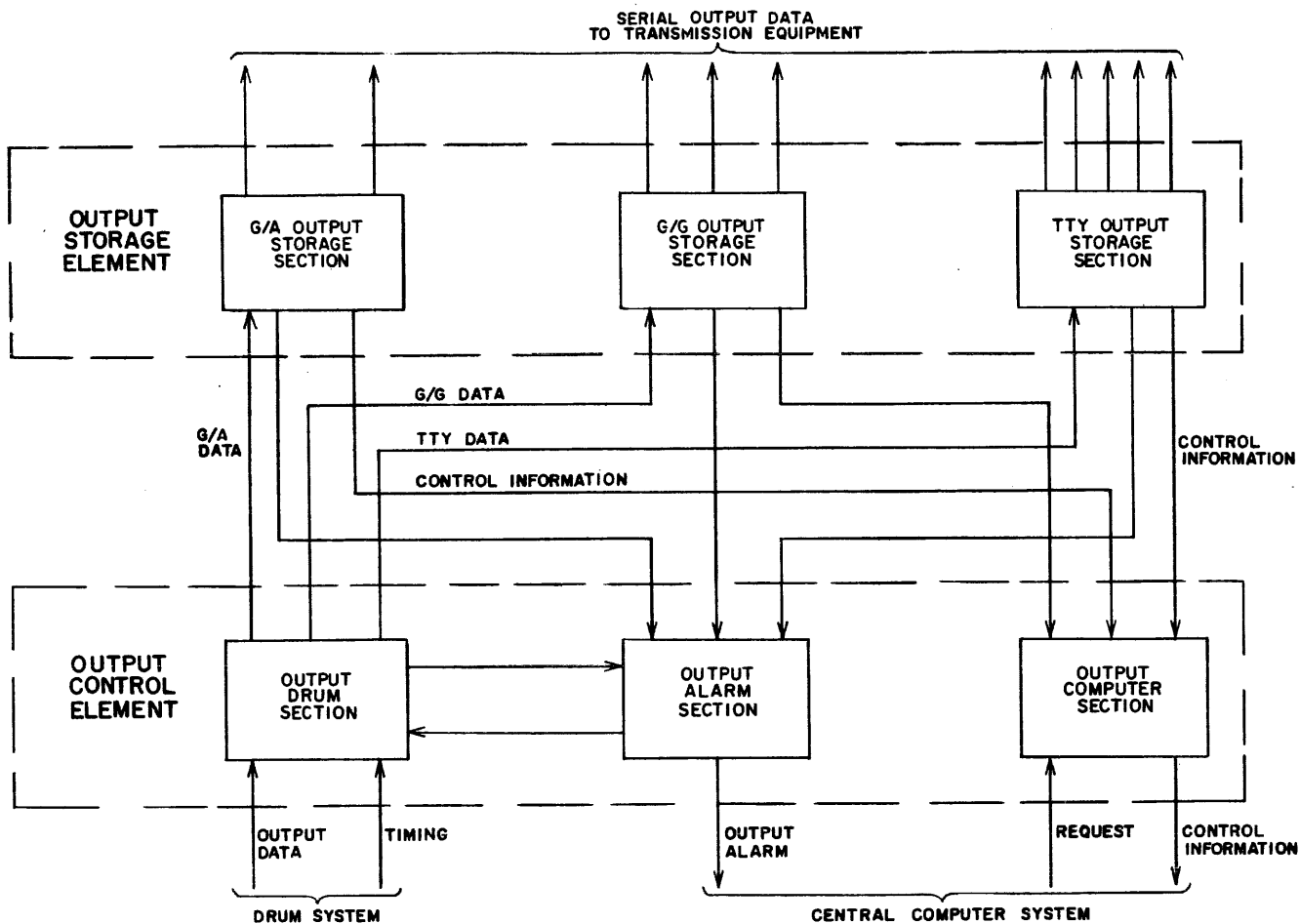


Figure 1-172. Output System, Element Block Diagram

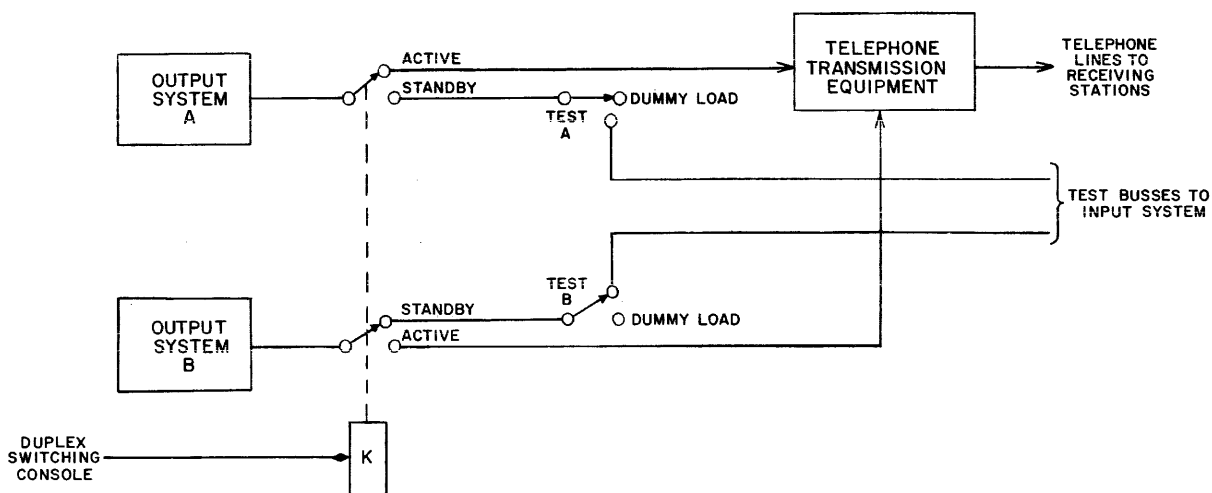


Figure 1-173. Output System Duplex Switching

control the exact time of transmission of a specific item. Each word prepared by the Central Computer System for delivery to the Output System contains an assigned burst number. These words are read into the Output System

by status identification, using the burst number as the identity code. Each storage section accepts only those words whose assigned burst number corresponds to the number of the burst being prepared by that section at

that time as indicated by a counter in the section (the burst counter). Thus, the assignment of burst numbers to the output drum words can control the sequence and time of transmission of information from each section.

The burst period for each output storage section is defined as the interval between successive stepping of its burst counter. Two intervals are included within a burst period, search time and read-out time. Search time is the interval within which words are accepted by a storage section for transmission during the following read-out time. Read-out time is the interval during which words previously assembled are transmitted.

Search time for each storage section must be of sufficient length to examine each register on the output buffer fields for words to be transmitted during a given burst period. The time required is 61.80 milliseconds, 20.48 milliseconds per drum revolution plus 120 microseconds for switching between fields. Each storage section allows approximately 65 milliseconds for its search time, more than sufficient time to search all three fields.

Read-out time for each storage section is dependent upon the number of bits to be transmitted on a single transmission channel within a burst and upon the rate of transmission on that channel. Since either the number of bits or the rate of transmission varies from section to section, read-out time is different for each section. Read-out time for G/A is 185 milliseconds, for G/G 65 milliseconds, and for TTY 500 milliseconds per burst.

The burst period for G/A is 250 milliseconds, equal in duration to the sum of search time and read-out time for that section. (See fig. 1-174.) The burst period for G/G is reduced to 70 milliseconds by utilizing two storage devices, one of which receives words during a search time which is coincident with read-out time for the other. The burst period for TTY is held at 500 milliseconds by starting search time before readout of the preceding burst is completed.

7.1.3.2 Output Drum Word

Information is delivered to the Output System in the form shown in figure 1-175. The right half-word contains the information to be assembled with information from other words for transmission from the Output System in a burst. The left half-word controls the routing of the right half-word within the Output System. This left half-word can be divided into four major parts:

- Parity bit
- Output section address
- Output register address
- Assigned burst number.

The parity bit allows a check on the accuracy of transfer of the word through the Drum System to the

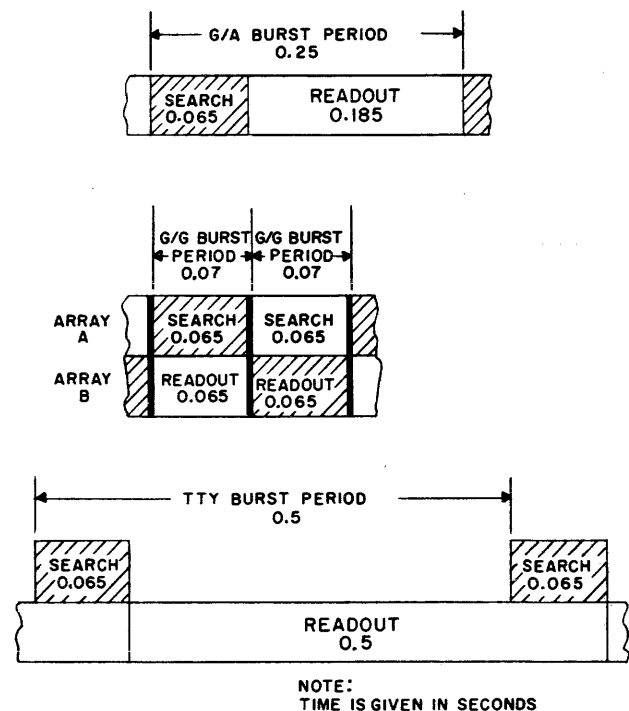


Figure 1-174. Output System Timing

Output System. The word is not accepted by the Output System unless the parity count for the entire 33-bit word is odd.

The output section address, LS through L2, designates the section of the output storage element through which the right half-word is to be transferred. Since only three output storage sections exist at present, the remaining five section addresses which can be indicated by the three binary bits are designated as illegal section addresses. (Refer to table 1-76.)

TABLE 1-76. OUTPUT SECTION ADDRESS CODES

OUTPUT SECTION	ADDRESS			OCTONARY EQUIVALENT
	LS	L1	L2	
Illegal Address	0	0	0	0.2
Ground-to-Air	0	0	1	0.4
Ground-to-Ground	0	0	0	0.0
Teletype	0	1	1	0.6
Illegal Address	1	0	0	1.0
Illegal Address	1	0	1	1.2
Illegal Address	1	1	0	1.4
Illegal Address	1	1	1	1.6

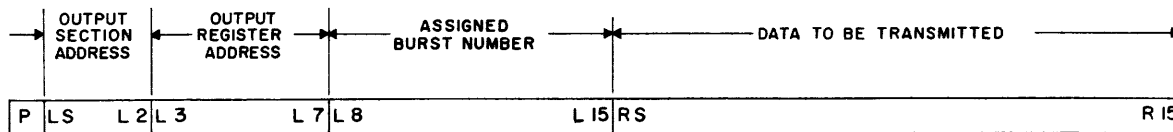


Figure 1-175. Output Drum Word Layout

The output register address, L3 through L7, designates the register within the specified storage section which is to receive the right half-word. Although 32 separate register addresses numbered from 0 through 31 may be specified by five binary bits, not all the registers are used in any section. Unused register addresses within each section are designated illegal register addresses. (Refer to table 1-77.)

TABLE 1-77. OUTPUT REGISTER ADDRESS CODES

OUTPUT REGISTER ADDRESS	CODED					OUTPUT REGISTER ADDRESS	CODED				
	REGISTER L3	ADDRESS L4	REGISTER L5	ADDRESS L6	L7		REGISTER L3	ADDRESS L4	REGISTER L5	ADDRESS L6	L7
0	0	0	0	0	0	16	1	0	0	0	0
1	0	0	0	0	1	17	1	0	0	0	1
2	0	0	0	1	0	18	1	0	0	1	0
3	0	0	0	1	1	19	1	0	0	1	1
4	0	0	1	0	0	20	1	0	1	0	0
5	0	0	1	0	1	21	1	0	1	0	1
6	0	0	1	1	0	22	1	0	1	1	0
7	0	0	1	1	1	23	1	0	1	1	1
8	0	1	0	0	0	24	1	1	0	0	0
9	0	1	0	0	1	25	1	1	0	0	1
10	0	1	0	1	0	26	1	1	0	1	0
11	0	1	0	1	1	27	1	1	0	1	1
12	0	1	1	0	0	28	1	1	1	0	0
13	0	1	1	0	1	29	1	1	1	0	1
14	0	1	1	1	0	30	1	1	1	1	0
15	0	1	1	1	1	31	1	1	1	1	1

The assigned burst number, L8 through L15, determines the order in which output drum words having different burst numbers will be read and transmitted. With 8 bits available, 256 burst numbers can be specified before the counting cycle must repeat. The assigned burst number of an output drum word must coincide with the number of the burst currently being assembled in the storage section selected by the section address of that word in order for the word to be accepted by the output control element.

7.1.3.3 Output Messages

Two blocks of G/A information are transmitted on separate channels during each G/A burst. Each block contains up to 13 right half-words plus a busy bit and an output parity bit for each original 16-bit half drum word. The 18 bits of each augmented half-word are serially interleaved with the bits of the other half-words so that successive bits of each half-word are 13 bits apart when transmitted. (See fig. 1-176.)

Three G/G messages are sent on separate channels during each G/G burst period. Each G/G message contains five right half-words and their parity bits. A G/G message is transmitted in serial interleave form so that successive bits of the same 17-bit half-word are entered on the telephone line five bits apart. (See fig. 1-177.)

The TTY storage section transmits three teletype characters per line during each burst. Figure 1-178 shows the composition of one character as generated by a teletype transmitter. Each character consists of seven consecutive intervals during which a switch in the transmitting end of the line is either open or closed to signify a mark or a space, respectively. The first and last of these seven intervals are the start and stop intervals. The five intervening intervals contain the coded letter, number, or symbol information identifying the character being sent. Since there are only two possible conditions of the teletype line, a mark or a space, the code is therefore binary in nature.

A teletype burst transmits three characters on each channel. (See fig. 1-179.) The first interval is the start bit of the first character. The next five intervals contain the character code, and the seventh interval of the first character is the stop bit. The second and third characters are sent in the same manner to complete a teletype burst.

7.2 SYSTEM OPERATION

7.2.1 Output Control Element

7.2.1.1 Introduction

All information intended for transmission via the Output System must pass through the output control element. This information is presented, one word at a time, to the output control element from the output buffer fields, is accepted or rejected on the basis of its assigned burst number, checked for errors in transfer or direction, and then placed in the appropriate location in the proper output storage section.



Figure 1-179. Transmitted TTY Burst

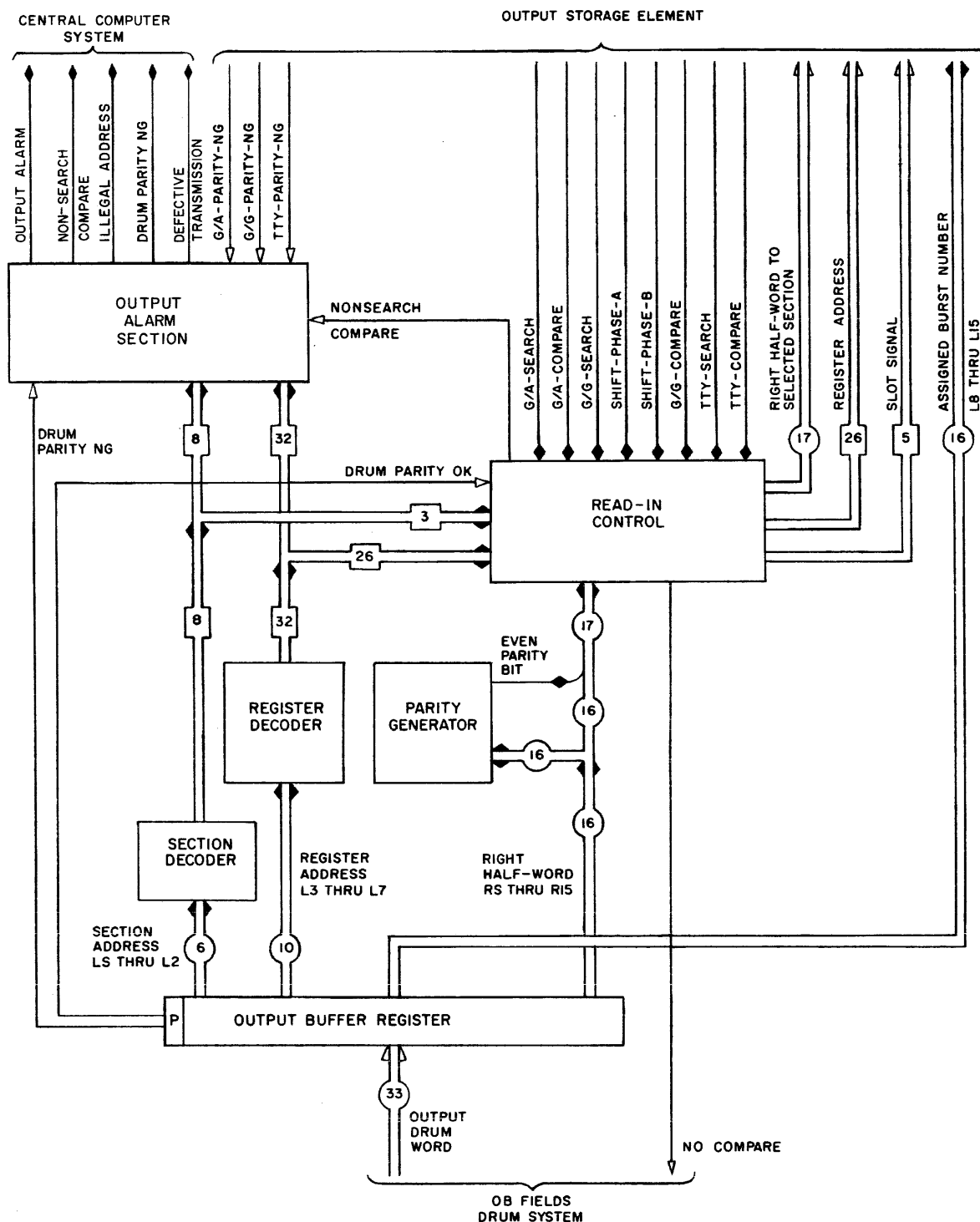


Figure 1-180. Output Control Element, Simplified

word allows recognition of only the appropriate compare level. If read-in control does not receive a G/A-, G/G-, or TTY-compare level, the output drum word is not accepted at this time. A no-compare pulse is sent by read-in control to the output buffer fields to retain the word for later acceptance. The burst counter for each output storage section is stepped by 1 just before that section begins assembling information for a new burst.

Thus, the G/A and TTY burst counters allow acceptance, during a given burst period, of words containing the assigned burst number for that burst period. The G/G burst count is one larger than the assigned burst number for the burst being transmitted during a given burst period.

The section address decoder indicates to read-in

control whether the right half-word is intended for the G/A, G/G, or TTY storage section. If the section address selects a section other than the three existing sections, the address is illegal, causing rejection of the word and generation of an output alarm. The register address decoder indicates, to read-in control, the register within the selected storage section in which the right half-word will be stored if accepted. Although there are 32 separate register addresses, the G/A section utilizes 26 registers, numbered 0 through 25; G/G utilizes 15, numbered 0 through 14; and TTY only ~~25~~ ²⁵, numbered 0 through ~~24~~. Addresses specifying unused registers within each section are illegal addresses. Their attempted usage will cause rejection of the word in which they are found and generate an output alarm.

Each output storage section applies a search level to read-in control when that storage section can accept information from the output buffer fields. The G/G storage section has two storage matrices that receive information alternately from the output control element. The G/G section must also indicate to read-in control which array is to receive information during the interval that the G/G-search level is up. The levels, shift-phase-A and shift-phase-B, perform this function.

The following information is thus applied to read-in control.

- a. Right half-word plus even parity
- b. Compare levels from one or more burst counters
- c. Legal section address
- d. Legal register address
- e. Parity-OK signal
- f. One or more search levels
- g. Shift-phase-A or shift-phase-B.

No transfer to the output storage element can occur unless the drum word parity check is successful. In addition, the right half-word is transferred to a storage section only if that section is selected by the section address, if the search level of that section is applied, and if the compare level from the burst counter of that section is applied. Finally, the specified register address must be legal. For the G/G storage section, the shift-phase-A or shift-phase-B level simply directs the transferred information to the correct storage array. If the drum word meets all these requirements, it is effectively removed from the OB fields by changing the status of the drum register containing it. Read-in control also supplies a slot signal to the G/A and G/G sections as each right half-word is transferred. In the G/A storage section, the slot signals are used to determine whether the correct number of right half-words is transmitted during a burst. In the G/G section, they are used to determine whether the correct number of right half-words has been supplied for transmission.

7.2.1.3 Output Alarms

The output alarm generator receives indications of seven types of output alarm conditions and forwards indication of an alarm condition to the Central Computer System. (See fig. 1-180.)

There are seven possible conditions that will cause the output alarm generator to generate an output alarm:

- a. Drum parity error
- b. Illegal section address
- c. Illegal register address
- d. Nonsearch compare
- e. G/A parity NG
- f. G/G parity NG
- g. TTY parity NG.

The drum parity NG alarm indicates detection of a parity error in a word received from the OB fields (usually caused by an equipment malfunction). The illegal section address alarm indicates selection of a section other than G/A, G/G, or TTY, while the illegal register address alarm indicates selection of an unused register in one of these sections. Either of the illegal address alarm conditions can be caused by a program error or by an equipment malfunction. The nonsearch compare alarm indicates the finding of a word on the output buffer fields which has the same burst number as the burst currently being transmitted, while no search level is applied by the selected storage section. This indicates an error: either the word was not erased from the output buffer fields after it had been accepted by the output control element or it was written on the OB fields too late for acceptance. When the output storage element supplies the G/A, G/G, or TTY output words to the telephone transmission equipment, a transmission and parity check is made. (An even-parity check is made as opposed to the odd-parity check made on output drum words.) If an error is found, a G/A-, G/G-, or TTY-parity-NG signal indicating defective transmission is sent to the alarm generator, and an output alarm is generated.

These seven alarm conditions are grouped into four types by the output alarm generator:

- a. Drum parity NG
- b. Illegal address (combines the illegal section address and the illegal register address)
- c. Nonsearch compare
- d. Defective transmission (combines G/A parity NG, G/G parity NG, and TTY parity NG).

Three of the four types of alarms (drum parity NG illegal address, and non-search compare) will cause read-in control to suppress the no-compare pulse to the Drum System, thereby having the output drum word erased from the output buffer fields. A defective transmission alarm cannot suppress the no-compare pulse, since the word involved in the alarm condition has al-

ready been accepted and removed from the OB fields. The Central Computer System is notified by the output alarm generator of the existence of an alarm (by the output alarm) and of its type (by the drum-parity-NG, illegal-address, nonsearch-compare, or defective-transmission signals), thus allowing the program to take account of the existence of the alarm. Each of these signal lines to the Central Computer System sets a sense unit. The Central Computer System can examine each of these sense units and produce a branch in the program if the selected sense unit is on.

The output alarm generator also supplies indications of all Output System alarm conditions to the duplex maintenance console associated with that Output System. These alarm condition signals turn on indicators on the duplex maintenance console. When the Output System is in test mode (as determined by the Output System TEST-OPERATE switch on the duplex maintenance console), the alarm indications may be used to stop the Output System and retain the conditions in that system at the instant of error detection. Specifically, detection of a drum parity error on an illegal section or register address will stop the transfer of words from the Drum System to the Output System and retain the drum word which contained the error in the OB register of the output control element. Normal information transfer from the OB fields to the output control element is resumed after the OB register is cleared manually by a control on the duplex maintenance console.

7.2.1.4 Output Computer Section

The AN/FSQ-7 Combat Direction Central must, in certain cases, present information on the time of occurrence of some event via the Output System to various distant receivers. In order to eliminate the need for synchronized clocks at the transmitter and receiver, the information is sent as a measurement of the interval between the presentation of the information to the receiver and the occurrence of the event. The presented measurement of this interval must be valid at the time it is delivered. It must therefore be possible to take into account the delay between calculation and delivery of the time information. The delays involved are:

- Delay within the Output System and from it to the receiver
- Delay between the calculation of time information by the Central Computer System and its delivery on the output buffer fields
- Delay due to storage on the output buffer fields.

The delay in transmission within the Output System and from it to the receiver can be assigned a fixed value, since this delay is constant for each storage section. The delay within the Central Computer System is negligible in comparison with the needed accuracy of time information and can be ignored if there are a num-

ber of bursts already on the output buffer fields. Since the storage time on the output buffer fields is variable, it must be calculated by the Central Computer System. In order to calculate this delay, the Central Computer System must retain a record of the burst number being assigned by the Central Computer System and must obtain the number of the burst being currently assembled or transmitted. The latter is obtained by reading the contents of the burst counters via the output computer section.

The Central Computer System calculates delay due to storage by comparing the burst number being assigned with the present burst number to determine the number of bursts stored on the output buffer fields. This number, when multiplied by the known duration of each burst period is the time delay due to storage on the output buffer fields. The calculated storage delay is erroneous by an amount equal to the time remaining within the present burst period after the reading of the burst counter. This error is, on the average, equal to $\frac{1}{2}$ burst period and may be added to the calculated storage delay.

The time accuracy required is different for different types of output information, the most stringent being for a G/A time-to-go message. Time-to-go information describes the interval of time between reception of time-to-go message by an airborne interceptor and the execution of the previously sent command. A G/A time-

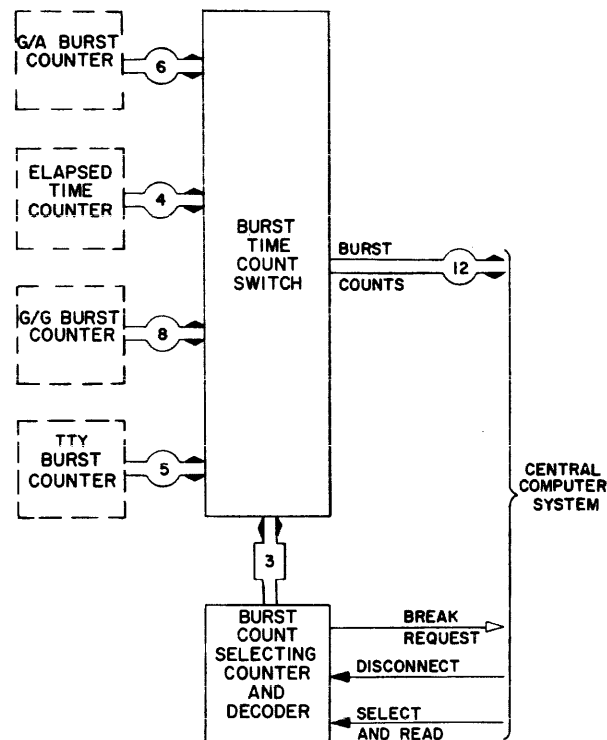


Figure 1-181. Output Computer Section, Simplified

to-go message requires accuracy greater than the $\pm 1/2$ G/A burst period ($1/2 \times 1/4 = 1/8$ second) obtained by calculation with G/A burst numbers alone. Therefore, the contents of the elapsed time counter are read by the Central Computer System to reduce the error in calculated delay to $\pm 1/2 \times 1/32$ second ($1/64$ second). The elapsed time counter is stepped by the 32-pulse-per-second time pulses from the real time clock in the Central Computer System and cleared when the G/A burst counter is stepped. Thus, the elapsed time counter measures time within a G/A burst period to an accuracy of $1/32$ second.

The contents of the G/A, G/G, and TTY burst counters and the elapsed time counter are transferred to the Central Computer System through the output computer section. (See fig. 1-181.) The output computer section consists of the burst count selecting counter and decoder and the burst time counter switch. The burst count selecting counter and decoder deter-

mines which burst count is sent to the Central Computer System. The Central Computer System, by means of the program shown in table 1-78, effects transfer of the contents of the burst counters and elapsed time counter to core memory.

The RDS instruction initiates the sequential transfer of the contents of the burst counters to the Central Computer System with the G/A burst count and elapsed time count sent first, then the G/G burst count, and finally the TTY burst count. (See fig. 1-182.) The three counts supplied by the output computer section to the Central Computer System are correct at the time of reading, since any changes in G/A, G/G, or TTY burst count or the elapsed time count is reflected instantaneously at the output computer section.

The number of significant bits presented to the Central Computer System by each counter is dependent upon the relation of the burst period of each section to the intervals between preparation of output messages in the Central Computer System. The 6 bits read from the G/A burst counter can count up to 63, covering an interval of 64 G/A burst periods (16 seconds). The 4 bits read from the elapsed time counter can count up to 15 intervals of $1/32$ second, covering a total interval of $1/2$ second or two G/A burst periods. The 8 bits read from the G/G burst counter can count up to 255, covering an interval of 256 G/G burst periods (almost 18 seconds). The 5 bits read from the TTY burst counter can count up to 31, covering an interval of 32 TTY burst periods (16 seconds).

7.2.2 Output Storage Element

7.2.2.1 Ground-to-Air Output Storage Section

At the beginning of each burst period, the G/A storage section steps its burst counter and supplies a search level to the output control element to indicate

TABLE 1-78. BURST COUNT READ PROGRAM

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
1	LDC	x	Places the core memory address in IO address counter for storage of G/A burst count.
2	SEL (21)	—	Selects burst counter for use in subsequent IO operations until another IO unit is selected.
3	RDS	3	Reads 3 words from burst counters.
4			Next program.

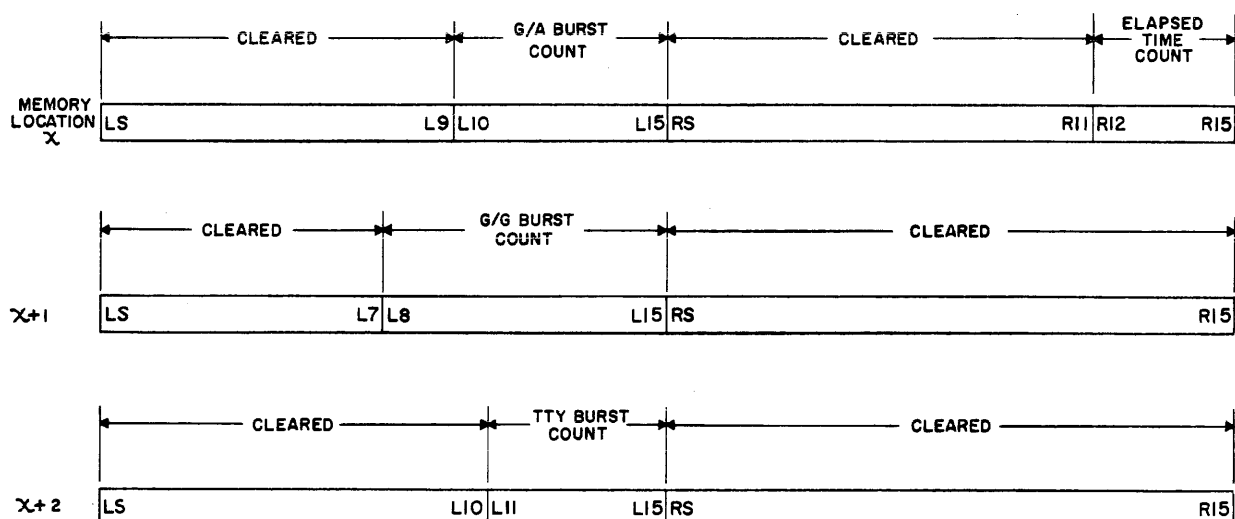
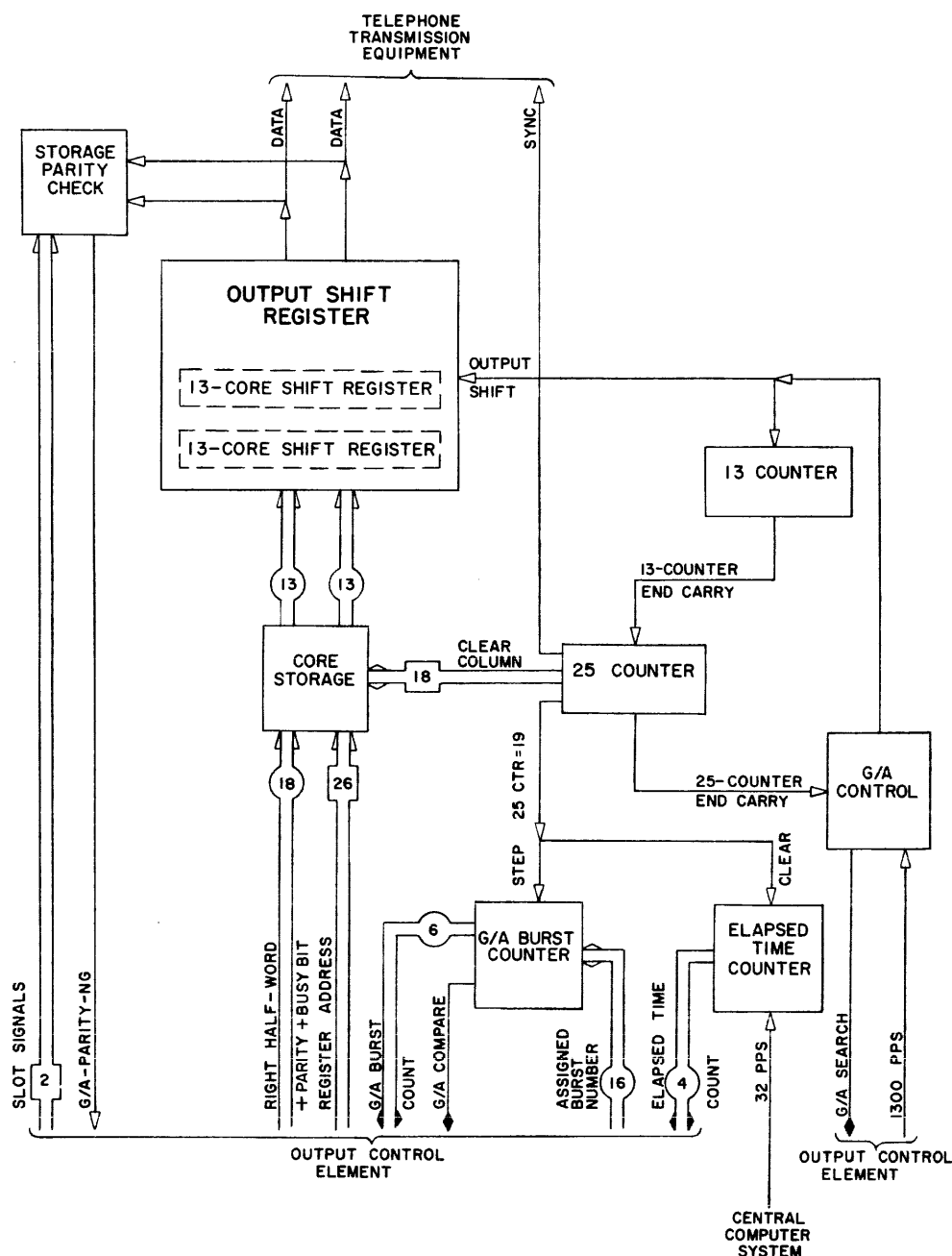


Figure 1-182. Burst Counts As Read by Central Computer System



that it is ready to receive right half-words for a new burst. The right half-words received during the search period are assembled in two groups called slots. Information contained in the two slots is subsequently converted from parallel to serial form, made compatible with telephone transmission requirements, and checked for defects in transmission. For convenience of discussion, the operation of the G/A storage section within each burst period is divided into two phases: read-in and readout.

Each right half-word accepted for the G/A storage section by the output control element is transferred in

parallel into a specified register of the G/A core storage array. (See fig. 1—183.) Up to 26 right half-words can be accepted during a search period, one word (consisting of a 16-bit right half-word plus an even-parity bit) per register. As each right half-word is placed in its register of the core storage array, a busy bit is placed in the bit position preceding the bits of the output word. Thus, there are as many busy bits placed in the array as there are words accepted during the G/A search period. It should be noted that the search period of 0.065 second is sufficient to examine all the registers in the OB fields. Therefore, output drum words that are due to be trans-

mitted by the G/A section during a given burst period will be obtained if they are placed on the OB fields before the start of the search period for that burst period.

Readout commences (shortly after the search level is removed from the output control element) with the transfer of information from the core storage array to the output shift register. This transfer occurs column by column, each column containing the corresponding bits of each word. The columns are pulsed in sequence, starting with column 1 (busy bits) and continuing through column 18 (parity bits), by clear-column pulses applied from the 25-counter. The output shift register receives 26 bits each time a column of the core storage array is cleared, placing 13 bits in each of the two 13-core shift registers making up the output shift register. Between each clear-column pulse which loads the output shift register, G/A control applies output shift pulses at a 1,300-pulse-per-second rate simultaneously to both core shift registers. Each output shift pulse shifts the contents of each core shift register one core to the right, thus supplying one data bit per channel to G/A conversion for serial transmission at a 1,300-bit-per-second rate.

After 13 output shift pulses have been applied to the output shift register, the contents of the 13-core shift registers within the output shift register have been transferred onto the telephone lines, and the output shift register is cleared. Since these output shift pulses are also applied to the 13-counter, that counter generates an end-carry as the 13th shift pulse is supplied to it. This end-carry steps the 25-counter, generating another clear-column pulse to reload the output shift register. Another 13 shift pulses move the data bits serially to G/A conversion and cause the generation of another clear-column pulse. After the 18th clear-column pulse, the G/A core storage array is emptied, completing the burst period. The new burst period begins at 25-counter-equals-19 when the G/A burst counter is stepped and the elapsed time counter is cleared. G/A conversion receives data from the output shift register, a sync pulse from the 25-counter at the beginning of readout, and timing pulses from the output control element for transfer to telephone transmission equipment.

During its transfer out of the output shift register, the information in each G/A channel is examined for errors in transmission. The examination consists of two checks, the busy bit check and the output parity check, both performed by the parity check circuit. If an error is found, a G/A-parity-NG pulse is sent to the alarm generator in the output control element.

The busy bit check is made to determine whether or not the same number of right half-words have been transferred out of the output shift register as were placed in the core storage array by the output control element. As each right half-word is placed in the core storage array, the output control elements send a slot signal

to the parity check circuit. The first column of the core storage array contains the busy bits generated as each word is placed in the core storage array. When this column is read out, the busy bits are sent to the parity check circuit. Unless the total count of slot signals and busy bits for each slot is even, a G/A-parity-NG pulse will be sent to the alarm generator at the end of read-out time. As the remaining columns of the core storage array are read out, an even-parity check is made to determine the output message parity count for each slot at the time of transmission. An even-parity check is made as opposed to the odd-parity check made on the output drum word by the output control element. Either error, in busy bit count or in parity count, sends a G/A-parity-NG signal to the output control element. It should be noted that the G/A-parity-NG signal has no effect on Output System operation but given an indication of its malfunction to the Central Computer System.

7.2.2.2 Ground-to-Ground Output Storage Element

At the beginning of each burst period, the G/G storage section steps its burst counter and supplies a search level together with a shift-phase-A or shift-phase-B level to the output control element to indicate that it is ready to receive right half-words for a new burst. The right half-words received during the search period are assembled in three groups, called slots. Information contained in the three slots is subsequently converted from parallel to serial form and checked for defects in transmission as it is supplied in compatible form to telephone transmission equipment. For convenience of discussion, the operation of the G/G storage section is divided into two phases: read-in and read-out.

Each right half-word accepted for the G/G storage section is transferred in parallel into a specified register of a G/G core storage array. (See fig. 1-184.) Up to 15 right half-words can be accepted during search time, one word (consisting of a 16-bit right half-word plus an even-parity bit) per register. The 15 registers within a G/G core storage array are divided into three groups, called slots. These slots are filled during the search period of 0.065 second, sufficient time to examine all the registers in the OB fields for output drum words due for transmission during a given burst period.

Since search time for the G/G storage section (0.065 second) is almost as long as read-out time (0.070 second), two core shift arrays are used to provide more efficient use of G/G telephone channels. If only one array were used, the telephone equipment would be idle for approximately 0.065 second (search time) between each G/G burst. With two arrays, one burst is read out of one array while the information for the next burst period is being assembled in the other array.

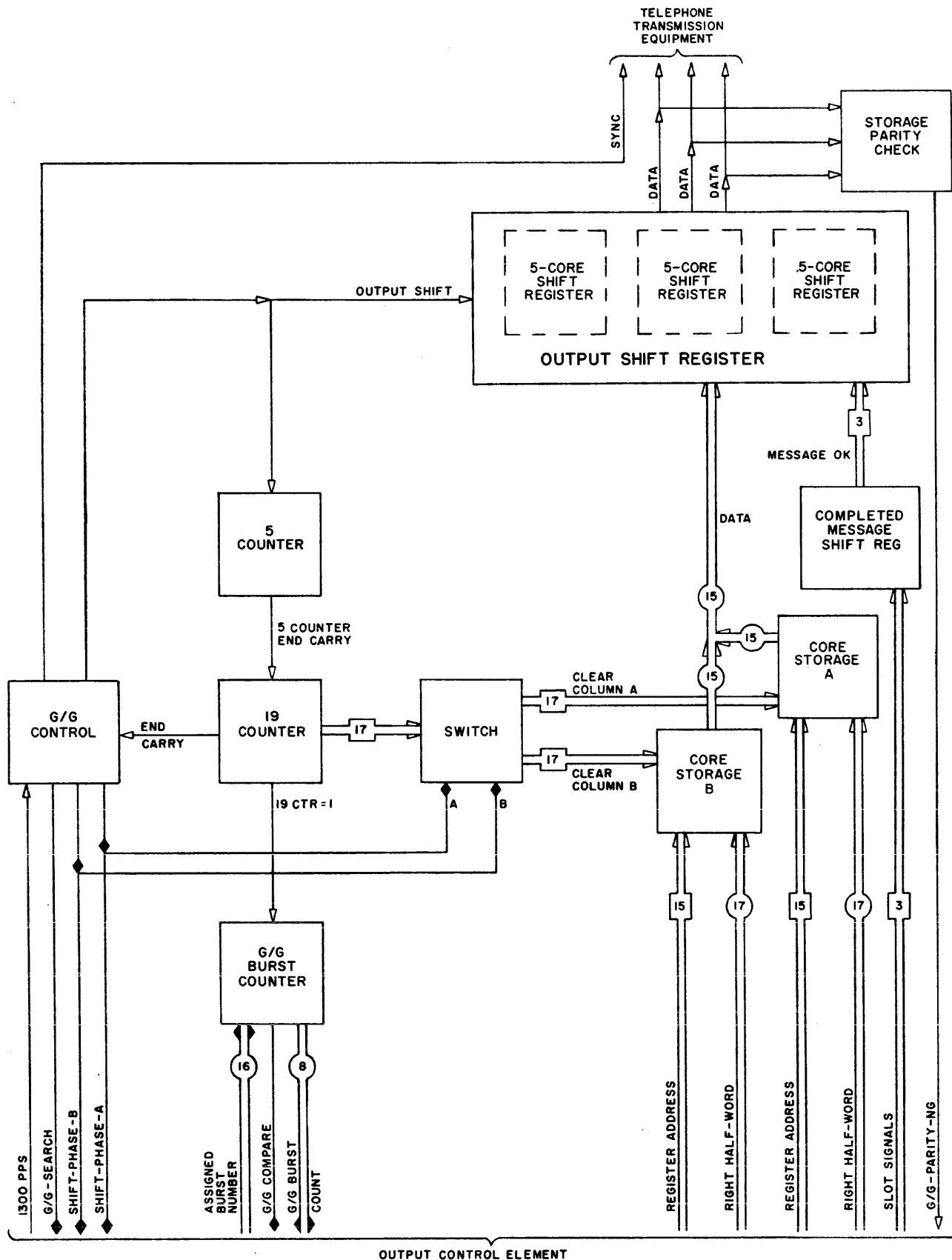


Figure 1-184. G/G Output Storage Section, Simplified

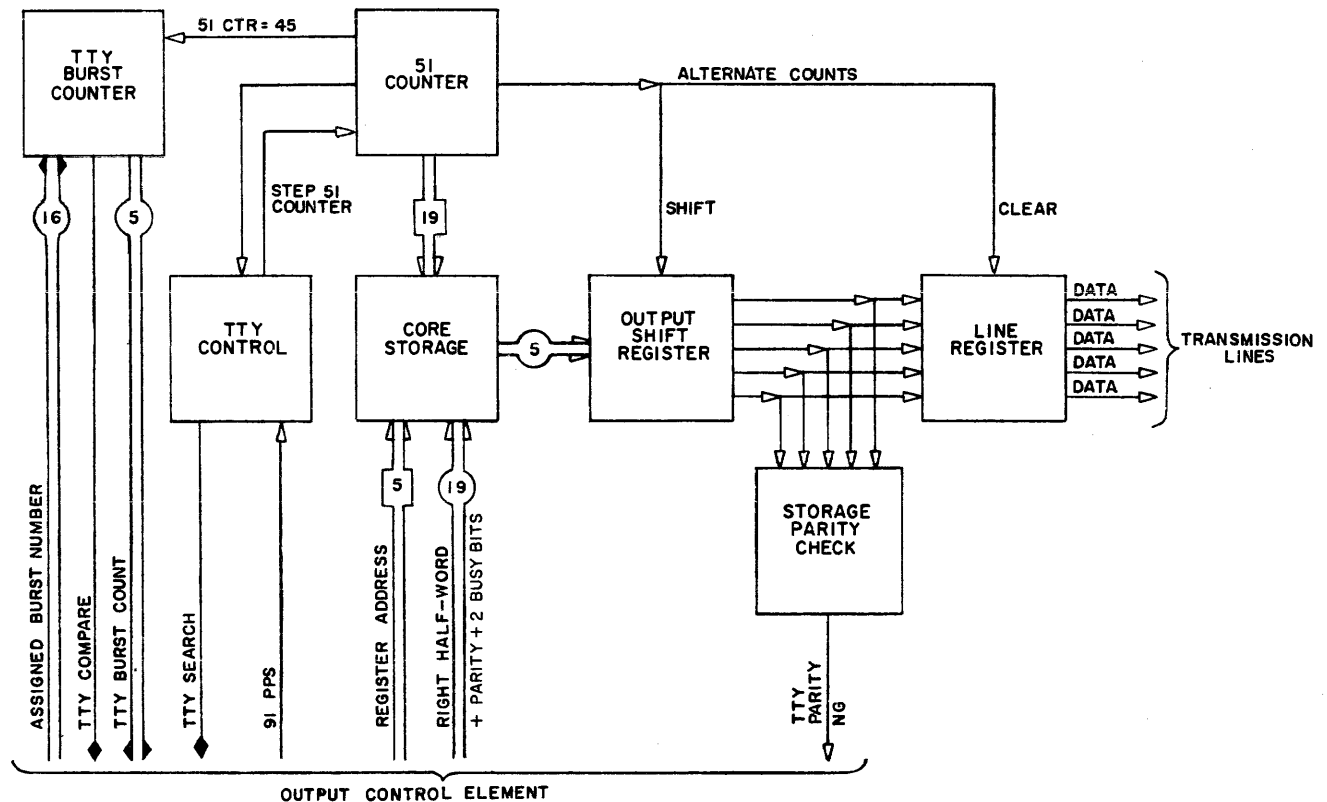


Figure 1-185. TTY Output Storage Section, Simplified

Readout commences with the transfer of information, column by column, from core storage array A or B to the output shift register. These columns are pulsed in sequence, starting with column 1 and continuing through column 17, by clear-column pulses applied by the 19-counter. Each column contains corresponding bits of the five right half-words in each slot. The output shift register receives 15 bits each time a column of the output storage array is cleared, loading them into the three 5-core shift registers making up the output shift register. Between each clear-column pulse, the contents of the output shift register are transferred serially to G/G conversion by shift pulses applied by the G/G control at a rate of 1,300 pulses per second. Each shift pulse is also applied to the 5-counter to produce an end-carry as the output shift register is cleared of the data loaded by the last clear-column pulse. The 5-counter end-carry steps the 19-counter, generating another clear-column pulse which reloads the output shift register. Data is thus transferred from the three slots of the core storage matrix through the output shift register to G/G conversion at the rate of 1,300 bits per second per channel. After the 17th clear-column pulse, the G/G storage array is empty, completing the burst period. The new burst period begins at 19-counter-equals-1 when the G/G burst counter is stepped and the two arrays reverse their functions; the array just read out is reloaded by the out-

put control element while the array that had been loaded is now read out.

Before any bits can be transferred from the output shift register to G/G conversion, the completed message shift register must send a message-OK pulse to the output shift register for each slot. The completed message shift register counts the slot signals from the output control element, indicating the transfer of right half-words into a slot in the core storage array. If a slot is filled during search period, a message-OK signal for that slot is supplied to the output shift register, allowing the message in that slot to be transferred out. If the slot is not filled, the output shift register will not receive a message-OK signal for that slot from the completed message shift register. Hence, the information in that slot is not transferred to G/G conversion but is cleared instead. G/G conversion receives data from the output shift register, a sync pulse from G/G control, and timing pulses from the output control element, and prepares this information for transfer to the telephone transmission equipment.

The information in each G/G channel is examined for parity error as it is transferred out of the output shift register to determine whether the output message parity count is correct at the time of transmission. If an error is found, a G/G-parity-NG pulse is sent to the output control element. An even-parity check is made as

opposed to the odd-parity check made on the output drum word by the output control element. It should be noted that the G/G-parity-NG signal has no effect on the Output System operation but gives an indication of a malfunction of the Central Computer System.

7.2.2.3 Teletype Output Storage Section

At the beginning of each burst period, the TTY storage section steps its burst counter and supplies a search level to the output control element to indicate that it is ready to receive right half-words for a new burst. The right half-words received in parallel during the search period are subsequently converted into serial form and checked for defects in transmission as they are supplied to the transmission quipment. For convenience of discussion, the operation of the TTY storage section is divided into two phases: read-in and readout.

Each right half-word accepted for the TTY storage section by the output control element is transferred in parallel into a specified register of the TTY core storage array. (See fig. 1-185.) Up to five right half-words can be accepted during a search period for storage in the core storage array, one word per register. Each right half-word contains code bits identifying three teletype characters. Before they are transmitted, a start bit and a stop bit must be added to each character. (Refer to 7.1.3.3.) The three start bits are the RS bit and two busy bits that are generated in the read circuit of the output control element and added as the half-word is written into the array. There are then 19 bits in each register of the TTY core storage array: the three start bits, bits R1 through R15, containing the three characters, and the even-parity bit. In order to generate all three start bits, bit RS must be a 1.

Readout commences with the transfer of information from the core storage array to the output shift register. This transfer occurs column by column, initiated by clear-column pulses which are alternate pulses from the 51-counter. The columns are read out so that the parity bits of all five words are read out first, then the start bits of the first characters, then, in sequence, the five data bits comprising the first character. The stop bits are generated by a pause before readout of the start bits of the second characters. The cycle continues until after the pause which indicates the end of transmission of the third character, completing readout of the core storage array.

The output shift register receives five bits each time a column of the TTY storage array is cleared, loading one bit in each of the five 3-core shift registers making up the output shift register. Each 3-core shift register feeds a separate teletype channel. Between each clear-column pulse, TTY control applies an output-shift pulse (a 51-counter pulse not used as a clear-column pulse) to both the five 3-core shift registers and the line register. Each shift pulse shifts the contents of each 3-core shift regis-

ter one core to the right, thus supplying one data bit per channel to the line register. The line register, which is cleared by the shift pulse just before the data bits are supplied, prolongs the application of each data bit on its transmission channel for a time equal to that of a mark generated by a teletype transmitter. After three output shift pulses have been applied by TTY control, the original contents of the output shift have been transferred to the line register. The time taken to shift the last bits of the last characters of the burst through the output shift register is $3 \times 2/91$ seconds, approximately the length needed for search time (0.066 second). Thus, by using this interval to search the OB fields and by refilling the output shift register within the time of transmission of the stop bits for the last characters, TTY transmission from the line register can be made continuous.

The last bit of the last character is read out from the core storage array to the output shift register at 51-counter-equals-43, leaving the core storage array empty and the last three bits of the present burst in the output shift register. The TTY burst counter is stepped, starting search time at 51-counter-equals-45, while these last three bits are shifted through the output shift register. Search time ends at 51-counter-equals-51 ($6 \times 1/91$, or 0.066 second); the core storage array is now filled. At 51-counter-equals-50, the stop bit of the last character is shifted into the line register, and the output shift register is empty. The stop bit stays in the line register for the required interval of three timing pulses until 51-counter-equals-2. At 51-counter-equals-1, the first bit (parity) is shifted from core storage into the output shift register. At 51-counter-equals-2, when the stop bit of the last character of the preceding message is cleared from the line register, a high-speed shift starts at a 50,000-pulse-per-second rate. The 51-counter, being stepped at a 50-kilocycle rate (one pulse every 20 microseconds), shifts the parity bit and three succeeding bits through the output shift register in 140 microseconds. At 51-counter-equals-8, the parity bit is cleared out of the line register, and the start bit of the first character is read into it, ending the high-speed shift. The parity bit, although applied to the parity circuit, is not applied to the line register long enough to cause transmission of a pulse. Thus, the high-speed shift prevents transmission of the parity bit and, at the same time, reduces the interval between the stop bit that ends the last burst and the start bit of the next burst to 0.12 millisecond. This interval is negligible in comparison with the 22-millisecond duration of each TTY bit. Therefore, transmission is made continuous by starting search time for a new burst before readout of the preceding burst from the output shift register is completed.

The information in each TTY channel is examined for parity error as it is transferred out of the output shift register to determine whether the output message parity

count is correct at the time of transmission. If an error is found, a TTY-parity-NG pulse is sent to the output control element. It should be noted that the TTY-parity-NG signal has no effect on Output System operation but gives an indication of its malfunction to the Central Computer System. An even-parity check is made as opposed to the odd-parity check made on the output drum word by the output control element. The even-parity bit supplied with each right half-word covers the parity of bits RS through R15. Two start bits (both 1's) are added although they were not included within the original even-parity count. If the RS bit is 0, only two start bits are produced, causing improper TTY transmission and generation of the TTY-parity-NG alarm. Therefore, the programmer must make the RS bit a 1 for all TTY output drum words to avoid generating an invalid TTY-parity-NG signal.

7.2.3 Output Test Section

There are three types of test facilities built into the Output System to be used for checking its operation: the unit loop test, the computer loop test, and the teletype unit test.

The unit loop is a means of checking the operation of the Output System independent of other systems of AN/FSQ-7 Combat Direction Central. In the unit loop test, a test word is manually inserted into the output buffer register of the output control element by toggle switches located on the duplex maintenance console. This test word is then processed through the output control element to either the G/A or the G/G storage section. Output data from the output shift register in the selected storage section is looped back into the output buffer register. If the transfer of the information is successful, the output buffer register is cleared. A failure can be detected by observing the neons on the Output system test panels which indicate the contents of the output buffer register.

The computer loop test is a programmed test conducted in conjunction with the Central Computer System. In the computer loop test, messages travel from the Central Computer System through the Drum System into the Output System. The messages are looped from the Output System back through the Input System to the Central Computer System for comparison with the original messages. There are two alternate paths for test messages leading back to the Central Computer System. If the test message is a G/G message, it will pass through the G/G storage section into the crosstell (XTL) input element and back to the Central Computer System. However, if the message is a G/A message, it is routed through the G/A storage section to the long-range radar input (LRI) element and back to the Central Computer System for comparison with the original message. Thus, the computer loop test can serve as a check on the operation of

the Output System and also on the operation of the Input System and the Drum System.

A third Output System check is the teletype unit test. In the TTY unit test, a test word composed of three known characters is manually set into the OB register of the output control element. This test word is then processed through the output control element and TTY storage section to a teletype printer used as a TTY monitor. (See fig. 1-186.) The data printed on this monitor is then visually compared with the known message to detect any error.

These tests are normally performed only in the Output System of the standby computer. Thus, while computer B is active, testing may be done on the Output System of computer A.

7.3 INFORMATION FLOW

7.3.1 Burst Preparation

7.3.1.1 Ground-to-Air Bursts

The Central Computer System program must prepare G/A information for transmission in the form of bursts. During each burst, up to 26 right half-words can be transmitted, 25 of which can contain a G/A message intended for a particular airborne receiver. In order to prepare a burst, the burst preparation program must be provided with the following information for each message:

- a. Message destination
- b. Information type (whether time-to-go or not)
- c. Information to be transmitted.

In order to deliver a specific G/A message to a specified receiver, the program must be written with a knowledge of the means available for directing that message to its destination. Each burst, although sent over two telephone channels from the G/A storage section, is supplied to a single G/A data-link transmitter. This G/A data-link transmitter receives all the messages within a G/A burst and uses each one to modulate a separate subcarrier. Up to 25 of these subcarriers are frequency-multiplexed on a single ultra-high frequency (uhf) carrier. Each subcarrier is subsequently received by as many as 16 aircraft, allowing transmission to as many as 400 aircraft. Only one airborne receiver accepts the message on a given subcarrier during one burst, since each message contains call letters identifying the one receiver within the group of 16 for which the message is intended.

A specific subcarrier can be selected by the burst preparation program, since each subcarrier receives information from a specific register of the G/A core storage array; the busy bit generated within the core register is used to turn on the appropriate subcarrier modulator. In effect, the assignment of a register address to a G/A output drum word addresses the message within that word to one group of 16 aircraft. Bits RS through

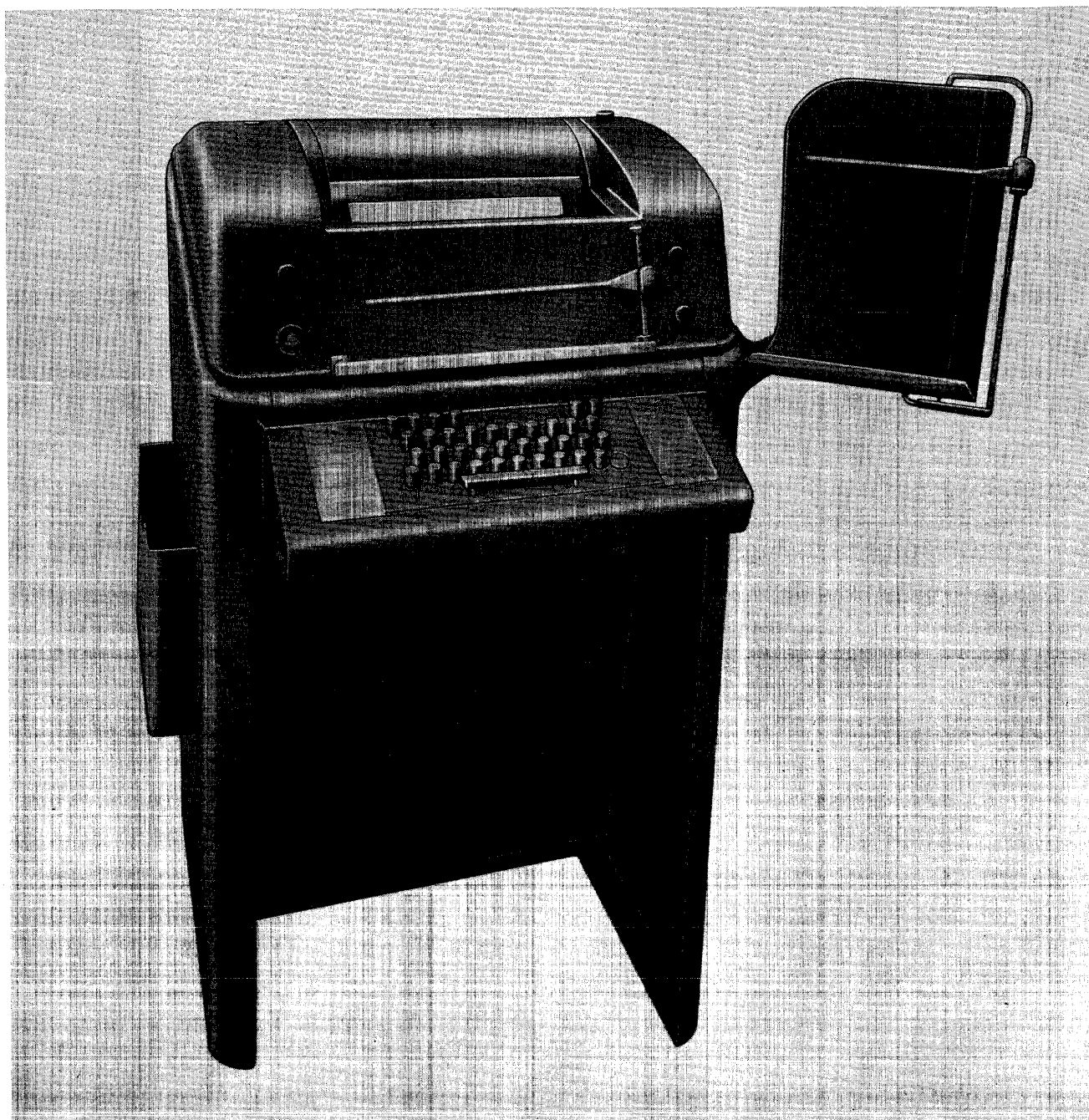


Figure 1-186. Teletype Printer with Keyboard Used As TTY Monitor

R3 are the call letters which designate one of the 16 aircraft as the message destination.

In addition to message destination, the burst preparation program must be informed of the nature of the message to be prepared. If the message is to contain time-to-go information, this information (prepared by another program) must be corrected by the burst preparation program by the time delay between preparation of the message and its delivery to its destination. The time-to-go information transmitted within a G/A message can be calculated as shown in the following equation:

$$T = T_c - [T_k + T_p(BC_a - BC_p) - E]$$

where T = Time-to-go transmitted

T_c = Time-to-go calculated by prior program

T_k = Time delay between Output System and receiver

T_p = Time of one burst period

BC_a = Burst count assigned to time-to-go message being prepared

BC_p = Present G/A burst counter contents

E = Present elapsed time count.

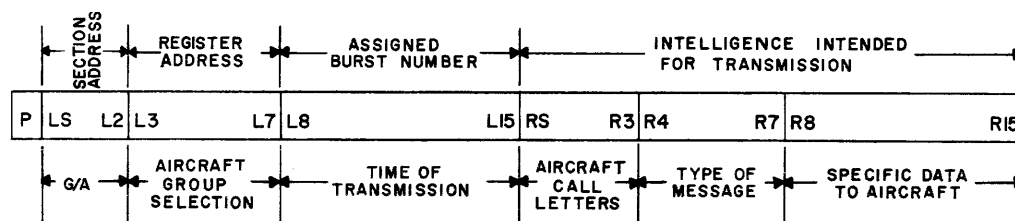


Figure 1-187. G/A Drum Word Layout

A general discussion of the means available for correcting time information within an Output System message is given in 7.2.1.4.

In addition to the first two requirements, the burst preparation program may be required to perform translation if the form in which the information is presented is different from that which must be received by the aircraft. Each message, when prepared in the Central Computer System, must be of the form shown in figure 1-187. The G/A section is specified by address bits LS through L2. Register address bits L3 through L7 indicate the aircraft group. Bits L8 through L15, the assigned burst number, determine the time of transmission. Bits RS through R15, the right half-word, are the intelligence intended for transmission. The first 4 bits of the 16-bit message, RS through R3, address the aircraft itself. The next four bits, R4 through R7, identify the type of message. Bits R8 through R15 contain the specific data for the aircraft. Twenty-five such words, each with the same burst number, may be prepared for transmission to airborne receivers within each burst. A 26th message in each burst is used for testing and controlling the G/A data-link transmitter but is not intended for transmission to an aircraft.

If a program or equipment error schedules an extra word for transmission within a G/A burst, some

alarm indication is generated. The output drum word containing the extra message may specify an illegal register address, causing the generation of an illegal address alarm. If the register address is a legal one, the extra message will be written over another message in the G/A core storage matrix. The resulting message in the matrix register will no longer have correct parity (except in a very unusual case) and will not be accepted by the G/A data-link transmitter. In addition, the busy bit check during G/A read-out will generate a G/A-parity-NG alarm. (Refer to 7.2.1.3.) Further, if the extra message is placed on the OB fields after the search time for its burst is over, a nonsearch-compare alarm is generated.

When the words for a G/A burst are prepared, they are written on the OB fields by status. The process of writing on these fields is discussed in 7.3.2.

7.3.1.2 Ground-to-Ground Bursts

The G/G information must be prepared, under program control, for transmission in the form of bursts. Three G/G messages may be transmitted within each burst, each message (containing five right half-words) being transmitted over a separate telephone channel to a specific destination. (See fig. 1-188.) The three types of messages include:

- Forward-telling messages to higher headquarters

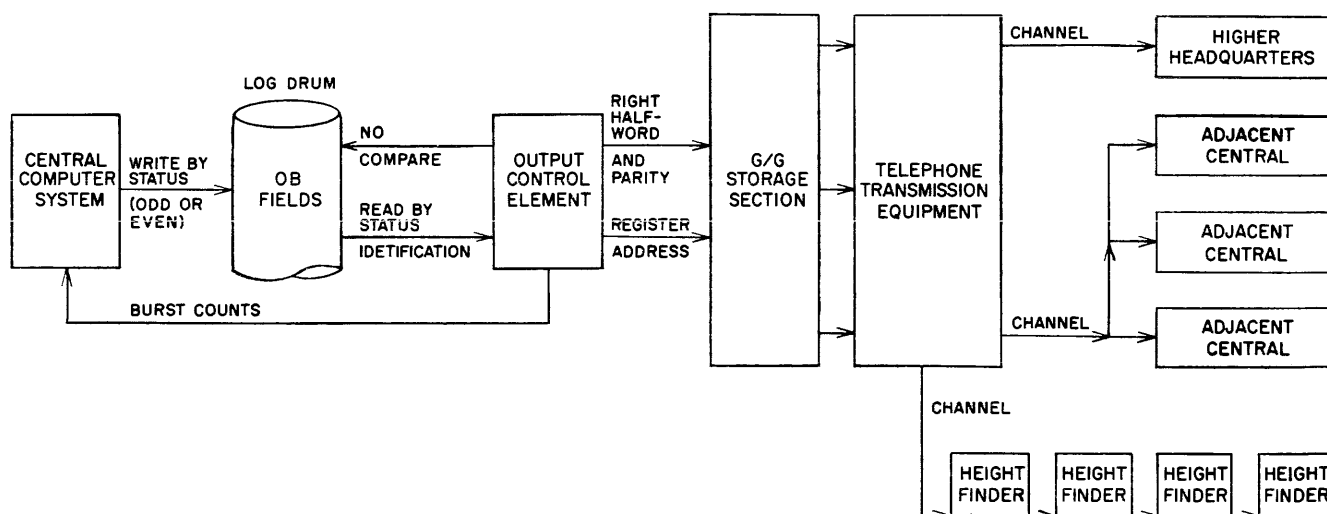


Figure 1-188. G/G Information Flow

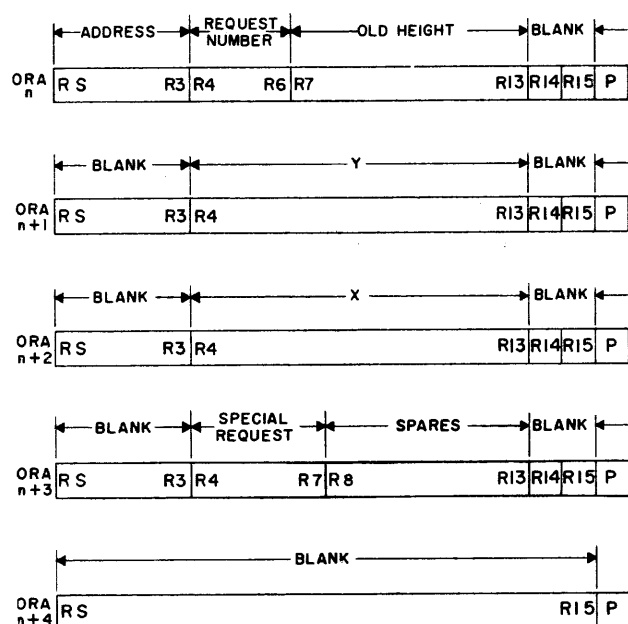


Figure 1-189. Height-Request Message Layout

- b. Crosstelling messages to adjacent centrals
- c. Height-request messages to height-finder radars at P sites.

It is assumed in this discussion that each of the three channels through the G/G output storage section is reserved for transmission of one type of message. Therefore, the type of message indicates directly to the burst preparation program the slot of the G/G core storage array in which it must be stored; i.e., message type indicates register addressing. In the case of forward-telling messages, no further addressing is required, since only one destination is possible.

Since crosstelling messages are transmitted via a party line connected to all adjacent centrals, the addressing of a specific message to a specific central must be accomplished within the message. This addressing is done in bits R11 through R15 in the fourth output drum word of a crosstelling message; bits R11 through R14 specify a single central whereas R15, if it contains a 1, specifies that all centrals should accept the message. The burst preparation program must therefore assign the correct register addresses to select the slot for a crosstelling message and include the specific address bits within the fourth message word.

Height requests are also transmitted over a party line connected to all P sites. The address of a specific height-finder radar is given in bits RS through R3 of the first output drum word of the message. The burst preparation program must again assign the proper register addresses to select the slot for a height request.

The layout of a forward-telling or a crosstelling message depends so much upon the nature of the information being transmitted that no typical layout can be given. Transmission of time information within a forward-telling or crosstelling message requires correction of the time information by the burst preparation program. This correction can be accomplished according to the equation:

$$T = T_c - [T_k + T_p (BC_n - BC_p) + 1/2 T_p]$$

where T = Time information as transmitted

T_c = Time information as supplied to burst preparation program

T_k = Constant delay between Output System and receiver

T_p = Duration of one G/G burst period

BC_p = Present G/G burst counter contents

BC_a = Burst number assigned to burst being prepared.

The term $1/2 T_p$ is included to approximate the elapsed time interval within readout of the present burst and is roughly equivalent to the elapsed time count used in G/A message preparation.

The layout of a height request message is given in figure 1-189. Bits RS through R3 of the first message word contain the address of the specific height-finder radar. Bits R4 through R6 contain the request number (used to identify the reply). Bits R7 through R13 contain the old height of the target on which a new height determination is to be made. Bits R4 through R13 of the second and third message words contain the Y and X co-ordinates of the target with respect to the height-finder radar. Bits R4 through R7 of the fourth word indicate a special request, if any. Bits R8 through R13 of this word are reserved for possible expansion of the number of possible special requests. The fifth message word, although it contains no information, must be prepared by the burst preparation program to satisfy the completed message requirement of the G/G output storage section. If this word is not prepared, the height request will not be transmitted.

In addition to directing each message to its appropriate destination, the burst preparation program may be required to translate information into the form necessary for transmission. Another possible requirement is the conversion of the reference of a target co-ordinate to the location of a height-finder radar. When all requirements are satisfied, the burst preparation program causes the transfer of the words for a G/G burst onto the OB fields by status.

TABLE 1-79. TELETYPE CODE

CHARACTERS		
LETTERS (DOWN)	FIGURES (UP)	CODE SIGNALS
A	-	00111
B	?	01100
C	:	10001
D	\$	01101
E	3	01111
F	!	01001
G	&	10100
H	#	11010
I	8	10011
J	,	00101
K	(00001
L)	10110
M	.	11000
N	,	11001
O	9	11100
P	0	10010
Q	1	00010
R	4	10101
S	Bell and Line Feed	01011
T	5	11110
U	7	00011
V	;	10000
W	2	00110
X	/	01000
Y	6	01010
Z	"	01110
	Space and Letters Shift	11011
	Carriage Return and Line Feed	11101
	Line Feed Only	10111
	Figures Shift	00100
	Letters Shift	00000
	Blank (no action)	11111

7.3.1.3 Teletype Bursts

Teletype (TTY) information, like G/A and G/G, must be prepared for transmission in bursts. There are five TTY channels fed by the TTY output storage section. During each burst, three TTY characters can be transmitted on each channel. Since a TTY message is always longer than three characters, it follows that several bursts are required to transmit a single message on each channel.

Each teletype character is represented by 5 binary bits. Thirty-two different characters can be transmitted: 26 represent letters or symbols; the other 6 convey machine commands such as blank, space and letters shift, line feed, carriage return and line feed, letters (down) shift, and figures (up) shift. The teletype code is given in table 1-79.

The 32 teletype code signals are used to transmit 51 symbols and 7 machine commands by using 2 commands to shift the teletype receiver carriage up or down. When the carriage is down, the code signals cause the printing of the symbols shown in the Letters column of table 1-79. When the carriage is up, the code signals cause the printing of the symbols shown in the Figures column of table 1-79. The letters-shift signal and the space signal shift the carriage down; the figures-shift signal shifts the carriage up. These machine command signals, as well as the carriage-return and the line-feed signals, are not dependent upon the position of the carriage.

Two precautions must be observed in teletype operation. First, a blank, a letters-shift, or a figures-shift signal must be sent immediately following the transmission of a *carriage return* command to allow time for the carriage to return. If this is not done, the first symbol printed after the *carriage return* command is printed while the carriage is still moving; this may prevent the carriage from returning all the way to the left side, resulting in double printing when the end of that line is reached. Second, the combination of figures shift followed by # (sometimes abbreviated as FIG H) must be avoided within a message. This combination ends a message by turning off the teletype receiving the message.

A teletype receiver is turned on when it receives a character or pair of characters transmitted by the TTY output storage section just before transmission of the message proper. These characters, known as selective addresses, are never printed by a teletype receiver; they serve to turn on the receiver or receivers which are to receive the following message. A particular address may designate a single receiver, a group of receivers, or all the receivers on a particular channel. (See fig. 1-190.) Some receivers may be assigned a group address in addition to their unique addresses. In addition, certain stations may have a second unique address which makes

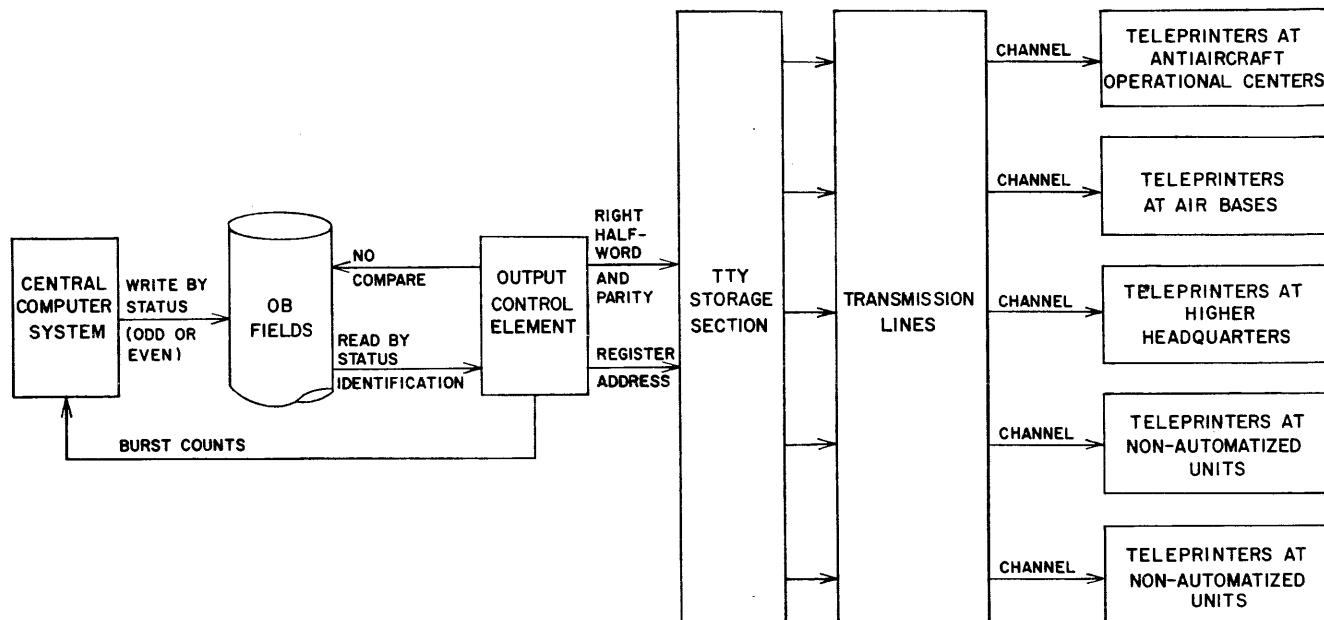


Figure 1-190. TTY Information Flow

the printer ready and also sounds an alarm at the receiving station. The universal address selecting all receivers on a channel is the letter U.

The program used to prepare teletype messages for transmission directs a message to the correct destination by first selecting the channel connecting that destination to the central and then transmitting the selective address which turns on the receiver at that destination. When addressing is completed, a carriage-return-and-line-feed signal is sent. The program then causes transmission of the message in bursts, each burst containing three characters. The program may be required to translate the information to be transmitted into teletype code as well as to prepare it in the form of bursts.

Since teletype receivers can print no more than 35 characters (counting spaces) per line, the program must break up long messages into lines by inserting a carriage-return-and-line-feed signal (followed immediately by a nonprinting signal) after each line of a long message. When transmission of the message is completed, attention may be called to the message by having the bell at the teletype receiver rung several times. Transmission is ended by sending the sequence of figures-shift signal, # signal, and letters-shift signal, which turns off the receivers.

The form in which a sample teletype message is sent is shown in table 1-80. The message is transmitted on channel 2 as indicated by the output register address. Address characters DD and DE, each followed by a letters-shift signal to indicate the completion of an address, are transmitted by the TTY output storage section to turn on the selected receivers on channel 2. The car-

riage-return-and-line-feed signal indicates the start of message and readies the receivers for the message. The letters-shift signal sent next provides the time necessary for completing the carriage return.

The message transmitted by the series of bursts shown in table 1-80 would be printed out at the receivers in this form:

B36 AB 00

B244 V 330

At the end of the message, the bell on the receiver is rung six times. The receivers are then turned off by the figures-shift, #, letters-shift signal sequence. The final character in the last burst is made a blank to prevent its possible interpretation as a selective address. A succeeding message must start with the transmission of selective address characters and proceed as described for this sample message.

7.3.2 Delivery to Drum System

All words to be delivered to the Output System must be placed on the output buffer (OB) fields of the LOG drum by the Central Computer System. Although there are three OB fields, they are considered on the CD side as two fields of 3,072 registers each. One field, OB even, comprises the even-number registers on the three fields; the other field, OB odd, contains the odd-numbered registers on the three fields. The Central Computer System can select either OB even or OB odd for writing by status. With either one selected, the maximum transfer rate between the Central Computer System and OB fields is one word every 20 microseconds.

TABLE 1-80. SAMPLE TELETYPE MESSAGE

DRUM WORD	OSA* LS-L2	ORA** L3-L7	BURST NUMBER L8-L15	START BIT RS***	CHARACTER NO. 1 R1-R5	CHARACTER NO. 2 R6-R10	CHARACTER NO. 3 R11-R15
1	011	00010	00000000	1	01101 D	01101 D	00000 LTRS
2	011	00010	00000001	1	01101 D	01111 E	00000 LTRS
3	011	00010	00000010	1	11101 CR + LF	00000 LTRS	01100 B
4	011	00010	00000011	1	00100 FIGS	01111 3	01010 6
5	011	00010	00000100	1	11011 SP+LTRS	00111 A	01100 B
6	011	00010	00000101	1	11011 SP+LTRS	00100 FIGS	10010 0
7	011	00010	00000110	1	11100 9	10010 0	01010 6
8	011	00010	00000111	1	11101 CR + LF	00000 LTRS	01100 B
9	011	00010	00001000	1	00100 FIGS	00110 2	10101 4
10	011	00010	00001001	1	10101 4	11011 SP+LTRS	10000 V
11	011	00010	00001010	1	00100 FIGS	01111 3	01111 3
12	011	00010	00001011	1	10010 0	00100 FIGS	01011 BELL+LF
13	011	00010	00001100	1	01011 BELL+LF	01011 BELL+LF	01011 BELL+LF
14	011	00010	00001101	1	01011 BELL+LF	01011 BELL+LF	00100 FIGS
15	011	00010	00001110	1	11010 #	00000 LTRS	11111 BLANK

* Output section address (OSA) must be 011 for TTY.

** Output register address (ORA) selects TTY channel.

*** RS must always be 1.

Character codes and meanings are given in table 1-79.

On the OD side, the OB fields are read as if they were one large triple-sized field. Thus, words can be delivered to the output control element at the maximum rate of one every 10 microseconds for checking and sorting. However, words can be accepted by a given output storage section at a maximum rate of one every 20 microseconds. (The maximum reliable switching rate for the cores used in the storage arrays is once every 20 microseconds.) Since the words intended for transmission within a single burst by one storage section are placed on the OB fields in alternate registers, the maximum rate at which they can be delivered to a section is one word every 20 microseconds. Thus, the division of the OB fields on the CD side into OB odd and OB even reduces

the information delivery to a rate compatible with the acceptance rate of a core storage array.

The actual rate of information transfer through the OB fields is dependent upon the amount of information which must be transmitted by the three output storage sections. The maximum rate at which information may be supplied to the OB fields without filling them is determined by the rate at which words are accepted by the output control element, which is in turn controlled by the transmission rate of each output storage section. A maximum of 328 words can be transmitted per second. Therefore, if the average rate of word transfer onto the OB fields does not exceed this value, no difficulties due to overfilling the OB fields are encountered.

CHAPTER 8

WARNING LIGHT SYSTEM

8.1 SYSTEM DESCRIPTION

8.1.1 System Function

The Warning Light System in AN/FSQ-7 Combat Direction Central provides program-controlled indicator devices which notify personnel at display consoles and GFI mapper consoles of special computer actions. Two types of indicator devices are used, neon lights and audible alarms which are paralleled by filament type lights.

Each indicator has a predetermined function. For example, activation of a neon light on a display console may signify to the operator that a new track has been initiated by the Central Computer System. The neon light will be extinguished after the operator has taken the proper action. In some cases, an audible alarm will be energized in conjunction with the neon light. A filament type lamp which blinks on and off is associated with each audible alarm. Both the audible alarm and the filament light may be manually de-energized by the operator at the console. However, the neon light will remain energized until extinguished by the Central Computer System.

8.1.2 System Logic

The Warning Light System is logically divided into three elements: storage, interconnection and indicator, and control. The storage and control elements are in duplex, while the interconnection and indicator element is in simplex.

The storage element contains a bank of eight 32-bit flip-flop registers (called warning light registers) which activate the various indicator devices. These registers are loaded with eight words from core memory under Central Computer System program control. The words control the status of the flip-flops, which in turn control the neon lights and the audible alarms. When a particular flip-flop is set to 1, the indicator devices controlled by the flip-flop are energized; when the flip-flop is cleared, these devices are de-energized. If the flip-flop controls an audible alarm also, the alarm is turned on when the flip-flop is set but may be turned off manually by depressing a momentary contact pushbutton. The alarm then remains off until the controlling flip-flop is cleared and reset by the program. The storage element also contains all the relays necessary for switching the simplex indicators between the two storage elements. Synchronization of the storage element with Central Computer System operation is provided by the control element,

which chooses each of the eight warning light registers sequentially for loading of the eight words from core memory.

The interconnection and indicator element consists of a patchboard interconnection unit and various indicators located at the consoles. The interconnection unit consists of two sections. One section, consisting of four panels of 32 by 34 hubs, receives all the warning light wires from the flip-flops in the duplex storage element. The other section, consisting of four panels of 35 by 48 hubs, receives all the wires from the indicator devices at the consoles and the unit status control lines. By proper patching, the indicator at any console may be connected to any flip-flop in the storage element. This arrangement provides flexibility in indicator assignments. The unit status control lines from a console will cause the outputs from the storage element whose status (active or standby) is the same as that of the console to be switched to that console.

8.2 SYSTEM OPERATION

8.2.1 General

In order to change the condition of one or more warning lights, the Central Computer System must institute an input-output (IO) operation, during which words are transferred from core memory to the warning light registers. The control element synchronizes the action of the storage element with the IO word transfers. In accordance with this function, the control element receives break-out and timing pulses (BO and TP) from the Central Computer System. It also generates break-request, clear-IO-interlock, and other control signals, which are sent to the selection control element in the Central Computer System, where they assist in the initiation, synchronization, and termination of the IO word transfers.

Each word from core memory is placed in a particular warning light register in the storage element. The warning light registers are numbered from 1 to 8 and are chosen sequentially by the warning light register counter (a counter of eight) in the control element. No more than eight words may be transferred to the warning light registers in any one operation, although transfer of fewer than eight is possible.

Each flip-flop in the warning light registers has two inputs, complement and clear. Each word transferred to the IO register of the Central Computer System is ap-

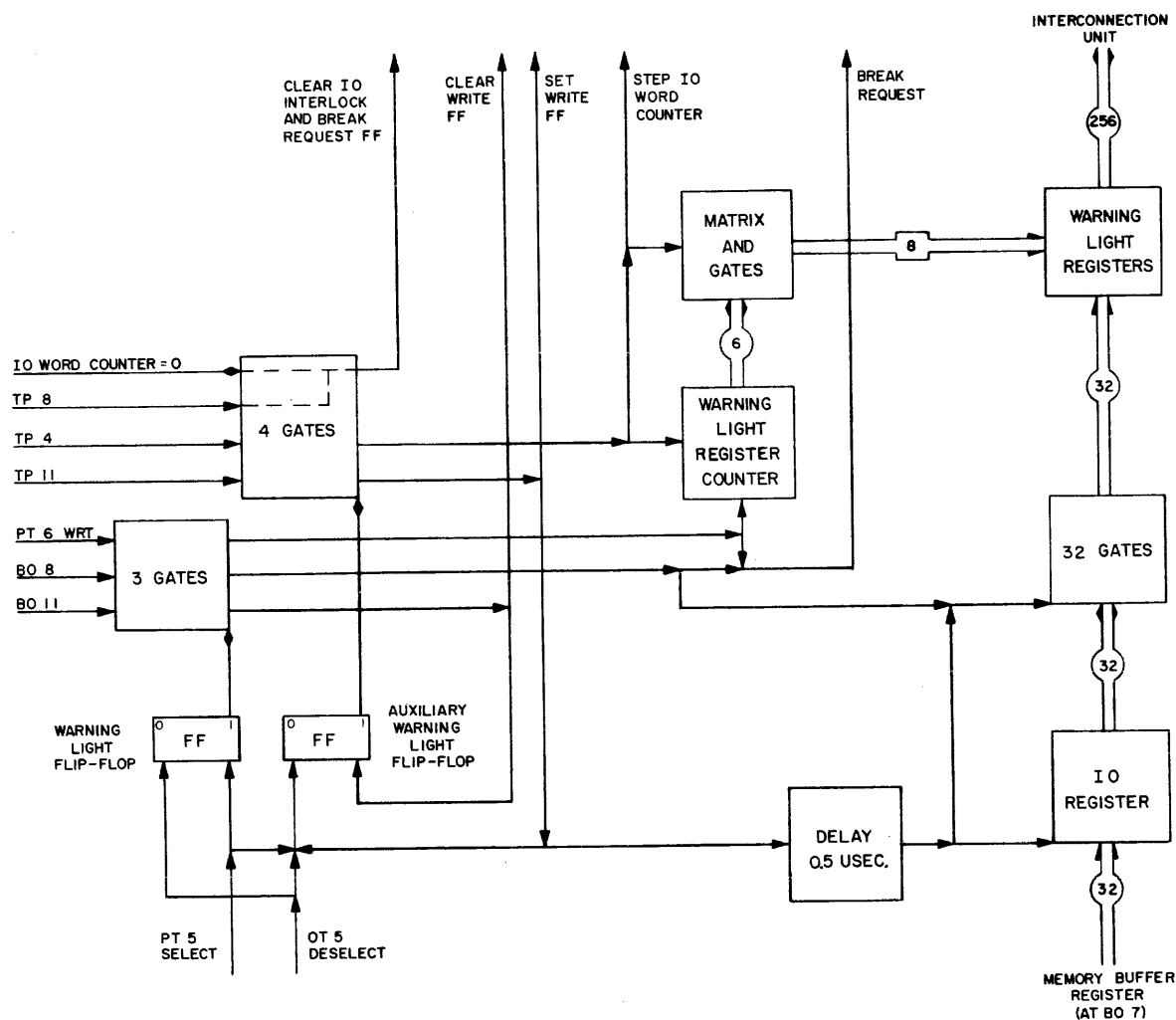


Figure 1-191. Warning Light Control and Storage Elements, Simplified Block Diagram

plied to the complement inputs of all eight warning light registers. Initially, this modifies the content of all eight registers without loading the word into the appropriate register. The register into which the word is to be loaded is then selected and cleared by the warning light register counter. The word in the IO register is again applied to the complement inputs of all eight registers. The contents of the cleared register will then correspond to the word in the IO register while the other seven registers are returned to their original condition. The same process is repeated for each word transferred to the warning light registers.

8.2.2 Analysis

The loading of the first information word into a warning light register is started with the receipt of the select pulse at PT 5. The select pulse sets the warning light flip-flop, which conditions a bank of three gate tubes. (See fig. 1-191.) At PT 6 of the *WRT* instruction, the warning light register counter is set to 001,

selecting the first warning light register (the eighth register is chosen when the counter returns to 000), and a break-request pulse is generated. The Central Computer System then proceeds to execute a memory break cycle, and, at BO 7, the selected word is transferred from the memory buffer register to the IO register. At BO 8, the contents of the IO register are transferred to the warning light registers. Thus, the IO register bits which contain a 1 complement the eight flip-flops in the corresponding column of the warning light registers. Since there is not enough time between BO 8 and BO 11 to complete the write operation, it is necessary to generate a dummy break cycle. To ensure completion of the write operation and to prevent the generation of break-out pulses, BO 8 requests the dummy break cycle, and BO 11 sets the auxiliary warning light flip-flop and clears the write flip-flop. The setting of the auxiliary warning light flip-flop conditions a bank of four gate tubes, which are sensed by timing pulses from the Central Computer System.

TABLE 1-81. WARNING LIGHT INFORMATION TRANSFER

TIME	IO REGISTER	WARNING LIGHT REGISTER 001	WARNING LIGHT REGISTER 010
Original Contents	0000000	1100010	0101011
First Break Cycle			
BO 7	1001101		
BO 8 (complement)		0101111	1100110
Dummy Break Cycle			
TP 4 (clear)		0000000	1100110
TP 11 (complement)	0000000	1001101	0101011

The matrix and gates section decodes the information in the warning light register counter and conditions one of eight gate tubes, each of which is connected to the clear inputs of a specific warning light register. The gate tubes are sensed at TP 4, and the selected register is cleared. The warning light register counter and the IO word counter are stepped at the same time.

At TP 11, the word in the IO register is again applied to the complement inputs of the eight warning light registers. This cancels the previous complementing action in seven of the eight registers. However, the contents of the IO register are duplicated in the first register, which had been cleared at TP 4. The auxiliary warning light flip-flop is cleared at TP 11, and the write flip-flop is set. This action ensures that the next memory cycle will be a break-out cycle. At TP 8 of each dummy break cycle, a gate tube conditioned when the IO word counter is 0 is examined. If the gate tube is conditioned, TP 8 clears the IO interlock and the break request flip-flop, thus ending the IO operation.

Table 1-81 illustrates the operations needed to change the contents of the first warning light register (001) to correspond to the contents of the IO register. The second warning light register (010) is representative of the seven remaining registers and shows the effect of the complement-recomplement action. The original contents of warning light register 001 are immaterial. However, the word to be written should contain the 1 bits corresponding to the warning lights that are to be turned on and the 1 bits corresponding to the warning lights that were on and which should remain unchanged.

The activation of an audible alarm and its associated filament light is subject to intervention by the op-

erator at the console where the audible alarm is located and to certain timing restrictions. The sounding of an audible alarm occurs only when its associated flip-flop is shifted from 0 to 1. Once the alarm has sounded, the operator may turn it off by depressing a momentary contact pushbutton at the console. Although this action turns off the audible alarm, the status of the controlling flip-flop remains unchanged. In order to resound an audible alarm, it is necessary to clear the flip-flop for a period of 2 milliseconds and then to reset it.

8.3 INFORMATION FLOW

Eight registers in core memory contain the image of the contents of the warning light registers. If the condition of a warning light must be changed, the bit in core memory representing the warning light is changed. The Central Computer System program periodically transfers the contents of the core memory registers to the warning light registers.

In order to effect the transfer of information, three IO class instructions must be executed: *LDC x*, *SEL (u)*, and *WRT n*. The first two instructions may be executed in any order, but must be followed by the *WRT n* instruction. The *LDC x* instruction specifies the address in core memory of the word to be placed in the first warning light register. The *SEL (u)* instruction designates the IO unit involved in the information transfer, which, in this case, is the Warning Light System. The *WRT n* instruction specifies the number of words to be transferred and initiates the first break-out cycle, during which the first word is transferred to the IO register. Two machine cycles are required in order to write each word in a warning light register. Thus, 96 microseconds are required to write 8 words into the warning light registers after the execution of the *WRT n* instruction.

CHAPTER 9

DUPLEX FACILITIES

9.1 EQUIPMENT DESCRIPTION

9.1.1 Introduction

The bulk of the equipment in AN/FSQ-7 Combat Direction Central has been described elsewhere in this part. The distinction between duplex and simplex equipment in each system has been presented in the chapter relating to each system. This chapter summarizes the duplex and simplex portions of the equipment; it describes the status in which each unit can operate and the facilities which control the status of the units.

9.1.2 Duplex Philosophy

The portions of AN/FSQ-7 Combat Direction Central whose failure would render the entire central inoperative and which are therefore duplex include the Central Computer System, the Drum System, the Output System, and portions of the Input, Display, and Warning Light Systems. (See fig. 1-10.) To simplify the switching operations required in the utilization of duplex equipment, all duplex units are assembled into two

complete groups, computer A and computer B. If a unit within one computer fails, it is not replaced by another unit; instead, that entire computer is replaced by the other computer. Switching an entire computer rather than units within it requires less interunit cabling and equalizes running time on the units within each computer. (See fig. 1-192.)

The units within the central which are not duplex fall into three groups: the channel input sections of the Input System, the situation display and auxiliary consoles of the Display System, and the interconnection unit and indicators of the Warning Light System. These units are simplex, since the failure of any one unit will not seriously disrupt the operation of their element. Instead of having duplicates of these units, a number of spares are provided in each group either to replace other defective units or to remove other units from operational use for maintenance.

Table 1-82 summarizes the division of equipment into duplex and simplex units.

TABLE 1-82. DUPLEX AND SIMPLEX EQUIPMENT

SYSTEM	DUPLEX	SIMPLEX
Central Computer System	All	
Drum System	All	
Input System		
LRI	Common input section, monitor control	Channel input sections, monitors
GFI	Common input section, monitor control	Data conversion receivers, mapper sections, counter sections, monitors
XTL	Common input section	Channel input sections
MDI	Direct IO buffer entry section matrix and read-out control	Direct IO buffer entry section keyboards and light guns
	Drum entry section controls	Drum entry section, computer entry punches, light guns, area discriminators
Display System	Situation and digital display generators, display tester, SD camera control	Situation and digital display indicator sections
Output System	All	
Warning Light System	Control and storage elements	Interconnection and indicator elements

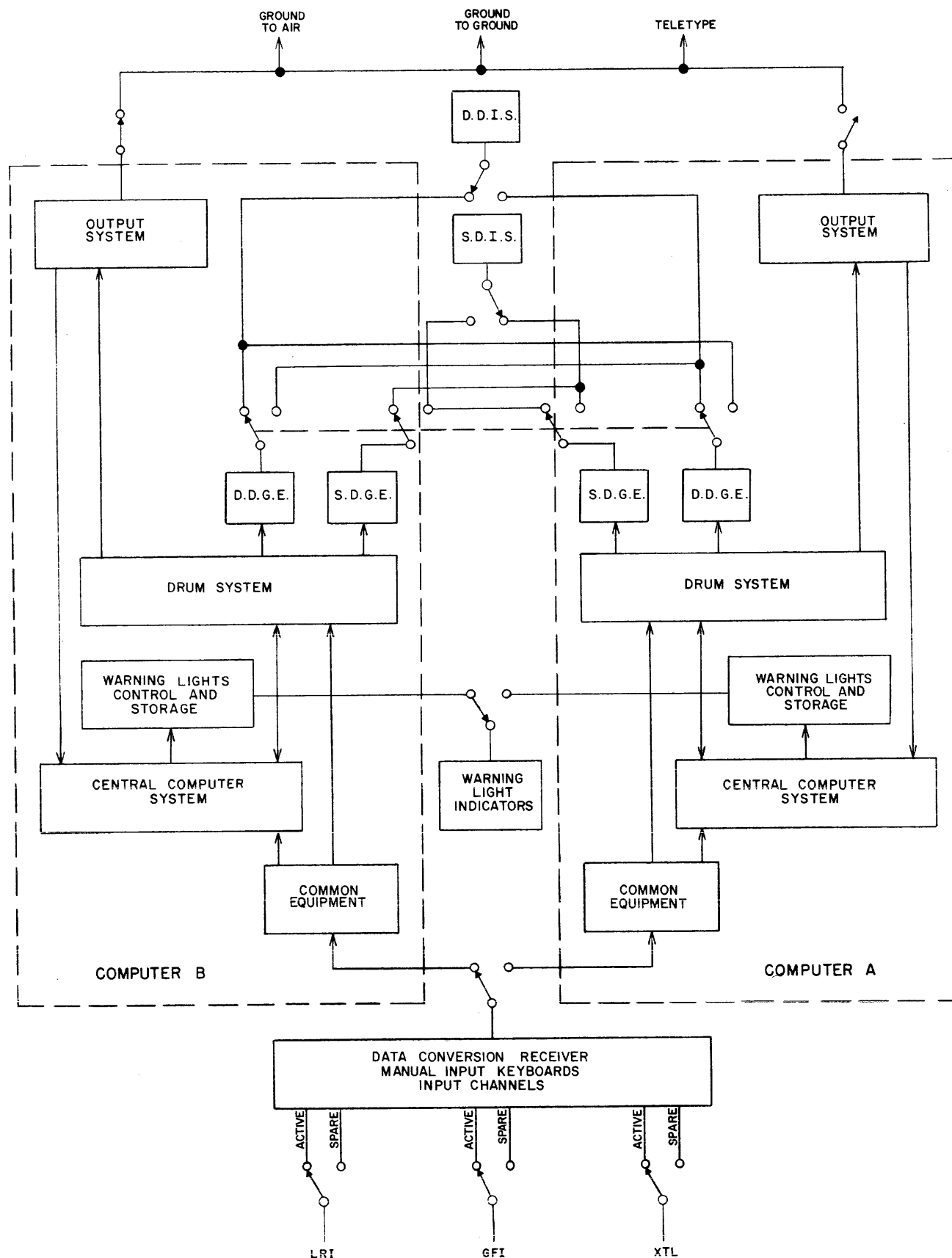


Figure 1-192. Duplex and Simplex Switching Facilities

9.2 STATUS

9.2.1 Active Status

Those units in AN/FSQ-7 Combat Direction Central which are actively processing air defense data are said to be in active status. Of the two duplex computers, only one is active at any one time. Those simplex units supplying information to, or receiving information from, the active computer must also be in active status. For example, the channel input sections of the Input System which receive data from radar sites and supply converted data to the active computer are themselves active. Those channel input sections which are undergoing maintenance or are simply turned off are not in active status.

9.2.2 Standby Status

The computer which is not in active status but is capable of taking over the operations necessary for air defense is said to be in standby status. Those simplex units which supply information to, or receive information from, the standby computer must be in standby status. For example, if a display console is to receive display data from the standby computer for test or training purposes, that console must be in standby status.

9.2.3 Other

Although the computer which is not active is usually in standby status, it can also be down for maintenance or completely off because of some equipment failure. Strictly speaking, in the two latter cases the computer is not in standby status. A more complete discussion of divisions within the standby status for a computer is given in Part 2, Chapter 1.

Simplex units can be placed in other than standby status. They may be placed in a status with only power on, with only a-c power on, or with no power applied. These conditions are useful in maintenance of these units.

9.2.4 Status Control

The status of units within AN/FSQ-7 Combat Direction Central is controlled in one of two ways. The status of the duplex computers is controlled at a duplex switching console. The status of simplex units is controlled by unit status switches located on the units or (in the case of the channel input sections of the Input System) on the simplex maintenance console. Changing the status of a computer requires prior information about the condition of each computer at the time of the contemplated switchover; the controls and indicators on the duplex switching console provide much of this information. Switchover of simplex units, which is much less involved than the corresponding operation for a computer, is controlled either at the unit whose operation can be evaluated directly by personnel there or at the simplex maintenance console, whose controls and indicators provide sufficient information for person-

nel performing switchover operations. The duplex switching console and the unit status switches are described in this chapter. The controls and indicators on the simplex maintenance console are described in Chapter 10, Section 3.

9.3 DUPLEX SWITCHING CONSOLES

9.3.1 Introduction

The controls and indicators necessary for controlling the status of the two computers in AN/FSQ-7 Combat Direction Central are contained on the duplex switching console. (See fig. 1-193.) One duplex switching console is provided in the maintenance area of the central.

In addition to controlling the status of the duplex computers, the duplex switching console exercises control over power supplies C and D, the duplex power supplies for the simplex units. One power unit supplies power to simplex units in active status; the other power unit supplies power to simplex units in any status other than active.

The controls and indicators on the duplex switching console are arranged on four panels called A, B, C, and D, respectively. Panel A contains all controls and indicators associated with computer A; panel B contains the corresponding controls and indicators for computer B. Panels C and D contain the controls and indicators associated with power supplies C and D, respectively.

INFORMATION NOT AVAILABLE

Figure 1-193. Duplex Switching Console

9.3.2 Duplex Switching Controls

9.3.2.1 General

Panel A on the duplex switching console contains the following:

- All summary indicators associated with computer A
- The duplex ACTIVE pushbutton A and associated relays switch
- The INTERLOCK BYPASS switch
- The marginal checking OPERATE TEST OVERRIDE BYPASS switch.

The listed controls are for the operation of computer A only. For the operation of computer B, panel B on the duplex switching console duplicates the controls mounted on panel A and performs the same functions for computer B.

9.3.2.2 Duplex ACTIVE Switch

The duplex switch consists of two pushbutton switches, one located on duplex switching panel A and the other on duplex switching panel B. Only one of these switches may be placed in active status at any one time. The switch that is activated causes the computer it controls to operate in active status. For this reason, it is called the ACTIVE switch.

When the ACTIVE switch is depressed on panel A of the duplex switching console, computer A is switched to active status and computer B to standby status. Depressing the ACTIVE switch on panel B of the duplex switching console switches computer B to active status and computer A to standby status.

In switching a computer into active status, the ACTIVE switch performs the following functions:

- It sends a signal (duplex switch alarm) to the computer with which it is associated to indicate it has been placed on active status.
- It sends a signal to place the other computer in standby status.
- It sets certain power units either to active or to standby status depending upon which duplex switch panel, A or B, is being operated. These units supply signals which, in turn, will operate relays at the display consoles, determining which computer information will be displayed at the consoles.
- It sends an output-active signal to each output storage element, causing the active element to send information out over telephone lines and allowing the standby element output to be used for testing purposes.

As a result of the duplex switching operation, a switching-complete signal is sent to each computer in the central from the duplex maintenance console associ-

ated with the computer which has been just switched to active status. This signal may be used to initiate the air defense program in the active computer.

9.3.2.3 INTERLOCK BYPASS Switch

An INTERLOCK BYPASS switch is provided on both panel A and panel B of the duplex switching console. Its function is to bypass open interlocks caused by existing conditions in the computer it is associated with and to allow that computer to be switched to its active status.

Under normal operating conditions, the standby computer may be switched to active status by the duplex switching console. However, the duplex switching action is prevented by open interlocks when the following conditions exist:

- Drum erase operation or writing timing track on a drum
- Drum motors off or not up to speed
- Power off at any of the logical units of the computer in standby condition
- Wrong test memory control panel plugged in
- Tests involving tape drive hazards.

To bypass these interlocks, the INTERLOCK BYPASS switch is used.

The INTERLOCK BYPASS switch completes a control signal path to the ACTIVE switch which is normally completed by the closed interlocks. This, in turn, permits the ACTIVE switch to operate the relays, causing the standby computer to be switched to active status although the interlocks are open.

9.3.2.4 TEST OPERATE OVERRIDE BYPASS Switch

The TEST OPERATE OVERRIDE BYPASS key-operated switch bypasses overriding of the TEST-OPERATE switches within a computer usually required to make that computer active. This action allows the active computer to be placed into a test mode while processing air defense data.

When either computer in AN/FSQ-7 Combat Direction Central is switched to an active status by the duplex switch, the TEST-OPERATE switches on the duplex maintenance console must be in the OPERATE position or must be overridden by use of the INTERLOCK BYPASS switch. Consequently, a system within the computer being switched to active status is also switched to operate status, regardless of the position of its TEST-OPERATE switch. If, within the computer in active status, a need arises to test one of its systems, the TEST OPERATE OVERRIDE BYPASS switch is used. Depressing this switch causes each system to operate as designated by its TEST-OPERATE switch on the duplex maintenance console.

9.3.3 Duplex Switching Indicators

Certain indicators have been provided on each duplex switching panel to indicate to those concerned with the air defense problem the state of each switch.

The indicators on each panel illuminate when each of the switches is actuated and the operation it is to perform has been completed. Thus, there are four lights which summarize the state of each computer in AN/FSQ-7 Combat Direction Central.

When the computer is in active status, as determined by the associated duplex switch, the ACTIVE light is illuminated. Its illumination starts when the switching-complete signal has been received at the duplex switching console. When the computer is in standby condition, the STANDBY light is illuminated. Illumination of this light indicates that its associated computer may be tested, checked, or utilized for functions not involved in the air defense problem. The OPERATE light, when illuminated, indicates that no system within the computer is on test or disabled. The TEST light illuminates to indicate that a system within the associated computer is in test mode. Normally, the TEST light is on only when the computer is in standby. However, upon activation of the TEST OPERATE OVERRIDE BYPASS switch, the TEST light can be illuminated with the computer in active status. The DISABLED light, when illuminated, indicates that one or more of the interlocks preventing switchover is open and that, consequently, the computer is disabled. (Refer to 9.3.2.3.)

In addition to the status lights on the duplex switching consoles, there are certain lights which warn the console operator of alarm conditions. These alarm indicators are summaries of the alarms that are pre-

sented on the duplex maintenance consoles. There are four alarm lights on each duplex switching panel. The NEON ALARM light, when illuminated, indicates that an alarm condition exists as indicated on the duplex maintenance console. The AIR CONDITIONING light is associated with the air conditioning of the computer; it will illuminate when the temperature or pressure is off limits in any system of the computer. The VOLTAGE OFF LIMITS light is illuminated when the regulated supply voltages have deviated more than ± 5 percent from their correct values. In the case of a deviation of ± 10 percent, an automatic shutdown of power occurs. The OUTPUTS SWITCHING light is illuminated when information from the Output System just switched to active does not reach the telephone lines.

A summary of the lights on the duplex switching panels, of their colors when illuminated, and of their functions is given in table 1-83.

9.3.4 Power Switching Controls

9.3.4.1 General

The simplex power switching operation is controlled by switches located on panels C and D of the duplex switching console. As in the duplex switching operation, the switching functions are performed by relays which are controlled by these switches. The panel controlling simplex Power System C is called panel C; the panel controlling simplex Power System D is called panel D. Panel C includes the following:

- a. All summary indicators associated with simplex Power System C
- b. The simplex power ACTIVE pushbutton

TABLE 1-83. DUPLEX SWITCH PANEL INDICATORS

INDICATOR	COLOR	INDICATION
ACTIVE	Red	Computer is in active status
STANDBY	Green	Computer is in standby status
TEST	Amber	Summary of the following test status lights on the duplex maintenance console: Central Computer System drums, tapes, and outputs
OPERATE	Red	Computer Systems are not on test or disabled
DISABLED	Blue	Any unit interlock which prevents duplex switching is open
ALARM	Red	Summary of all neon alarm lights on duplex maintenance console
AIR CONDITIONING	Red	Temperature or pressure is off limits at any unit
VOLTAGE OFF LIMITS	Red	Any voltage off limits, above or below a designated percent
OUTPUTS SWITCHING	Red	Output unit just switched to active not completely connected to telephone lines

- c. The INTERLOCK BYPASS switch used to override all interlocks which prevent simplex Power System C from being switched to active status.

Simplex power switching panel D includes the same equipment for simplex Power System D as switching panel C includes for simplex Power System C.

9.3.4.2 Simplex Power ACTIVE Switch

The simplex power ACTIVE switch consists of two pushbutton switches, one located on panel C and the other on panel D. The simplex power ACTIVE switch on panel C of either of the two duplex switching consoles is used to switch simplex Power System C to active status and simplex Power System D to standby status. The simplex power ACTIVE switch on panel D is used to reverse the condition selected by the simplex power ACTIVE switch on panel C.

Depressing the ACTIVE switch places the Power System with which it is associated in active status and automatically places the other Power System in standby status. Operation of the ACTIVE switch activates banks of relays in the simplex maintenance console as dictated by the operator of the duplex switching console.

Upon completion of switching, the duplex switching console sends a signal to both computers to indicate that the switching operation has been accomplished. Such a signal may be used to restart the air defense program.

In order to switch a Power System from standby to active status, all power interlocks must be closed. The Power System is rendered inoperative by open interlocks if the following conditions exist:

- a. The standby relay power is off.
- b. Any standby power supply service voltage is off.
- c. Regulated ac from standby power supply is off.
- d. Any standby power supply service voltage is off limits from normal by more than 10 percent.

Indications for these conditions are given on the duplex maintenance console.

9.3.4.3 INTERLOCK BYPASS Switch

At times, it may become necessary to switch the Power System that is in standby status to active status with some power interlocks open. The duplex switching console operator may do this by depressing the INTERLOCK BYPASS switch. The INTERLOCK BYPASS switch bypasses the power interlocks for the purpose of switching the standby Power System to active status.

9.3.5 Simplex Power Switching Indicators

The simplex power switching operation at the duplex switching console is facilitated by the illumination of certain indicators. These indicators provide a summary of the conditions existing in the simplex Power

System and furnish the operator with information essential to a switching operation. Each panel includes three status lights and one alarm light.

The status lights are marked ON, ACTIVE, and STANDBY. The ON light, when illuminated, indicates that the Power System with which it is associated is on. The on-off status of this Power System is controlled by the simplex maintenance console operator. The ACTIVE light is illuminated when the Power System with which it is associated is on and active. In this case, the active status is determined by the ACTIVE switch on the same panel. The STANDBY light is illuminated when the Power System with which it is associated is on and in standby status. The standby status of one Power System is an automatic result of the opposite Power System's being put on active status.

The POWER ALARM light is illuminated when the regulated voltage deviates more than ± 5 percent from its normal value. This light remains on when there is a complete shutdown of power.

A summary of these lights and of their functions is given in table 1-84.

TABLE 1-84. SIMPLEX POWER SWITCH
PANEL INDICATORS

INDICATOR	COLOR	INDICATION
ON	White	Power supply is on
ACTIVE	Green	Power is on and active
STANDBY	Amber	Power is on and standby
POWER ALARM	Red	Power supply voltages off limits

9.3.6 Other Facilities

In addition to the duplex and simplex power switching facilities, the duplex switching consoles include facilities for digital displays, manual data input keyboards, and telephone communication.

The duplex switching console is equipped with two digital display sections: one section receives information from computer A; the other, from computer B. Two data selection control panels input keyboards are also provided. Each keyboard, part of the manual data input element, is permanently connected to cores in one computer and is used to insert information into that computer.

The telephone facilities available at the duplex switching console are used by operating personnel for coordination of efforts in different areas. Such action ensures smooth operation and minimizes errors.

9.4 UNIT STATUS SWITCHES

The AN/FSQ-7 Combat Direction Central is equipped with facilities for utilizing 192 unit status switches. The sections which use these switches are sections which can make use of standby computer information or sections which supply this computer with information. Such sections are the situation display consoles, the auxiliary consoles, the computer entry punches, and

the input channels. The unit status switches for the input channels are mounted on the simplex maintenance console; those for the first three groups are mounted on the individual consoles and computer entry punches. Regardless of their location, the unit status switches operate on the same principle. A typical unit status switch operation is shown in figure 1-138. Each unit status switch has five positions: ACTIVE, OFF, STANDBY, POWER ON, and AC ON.

CHAPTER 10

MAINTENANCE FACILITIES

SECTION 1

INTRODUCTION

1.1 GENERAL

The use for which AN/FSQ-7 Combat Direction Central is designed requires a high degree of reliability (freedom from failures). The design criteria for the circuits used in the equipment place high priority on the reliability of these circuits. On a larger scale, the inclusion of duplexing in the design of the equipment increases the degree of reliability attainable by the equipment. To maintain equipment reliability, convenient and effective maintenance facilities are necessary. These maintenance facilities (and housing for operating controls for most of the equipment) are provided by two duplex maintenance consoles, one for each computer, and one simplex maintenance console for the simplex equipment (not including display consoles). These consoles, together with the duplex switching consoles described in Chapter 9 and the card machines and tape units described in Chapter 3, are located in a centralized maintenance area from which the operation of the bulk of AN/FSQ-7 Combat Direction Central can be controlled, tested, and monitored. (See fig. 1-194.)

Maintenance of the equipment is aided by the existence of duplex equipment. The computer in standby status normally runs reliability checks on itself in addition to supporting the active computer. These reliability checks are program controlled, thus using the high-speed characteristics of the equipment to exercise itself and check its own operation. Maintenance programs are written on one of two levels:

- a. Reliability programs which operate and check for operational errors in large portions of a system
- b. Diagnostic programs which operate and check for errors in a portion of the equipment found to be unreliable by a reliability program.

These programmed maintenance operations can localize causes of equipment malfunction closely enough in most cases to allow replacement of a small number of suspected pluggable units with known good units, thus restoring the equipment to normal operation in minimum time.

The reliability of the equipment is further improved by the use of facilities which allow detection of potential failure conditions; i.e., circuits or components which are likely to fail during the next active operation period. These potential failure conditions can be detected through the use of marginal checking, operating and testing these circuits while an abnormal supply voltage is applied to them.

All of these facilities (the duplex switching consoles, the duplex and simplex maintenance consoles, and the marginal checking facilities on the maintenance consoles) are contained in the central maintenance area. This chapter describes the controls and indicators on the duplex and simplex maintenance consoles and the marginal checking controls on these consoles. The other units located in the central maintenance area and those system maintenance facilities located at individual systems have been described earlier in this part.

1.2 CENTRAL MAINTENANCE FACILITIES

1.2.1 Duplex Maintenance Consoles

The duplex maintenance consoles each contain controls and indicators for initiating, monitoring, and maintaining the operation of the duplex equipment associated with each console. The two duplex maintenance consoles, one used with computer A and the other with computer B, are identical; hence, only one console is described.

The controls located on the duplex maintenance console include controls for applying power to the duplex equipment, loading programs into the Central Computer System, performing marginal checking of duplex equipment, and running various maintenance sequences on other systems contained in the computer controlled by that duplex maintenance console. The indicators housed on the duplex maintenance console provide visual indications of equipment operation, alarm conditions, and other pertinent conditions.

1.2.2 Simplex Maintenance Console

The simplex maintenance console provides a central housing for the controls and indicators used with

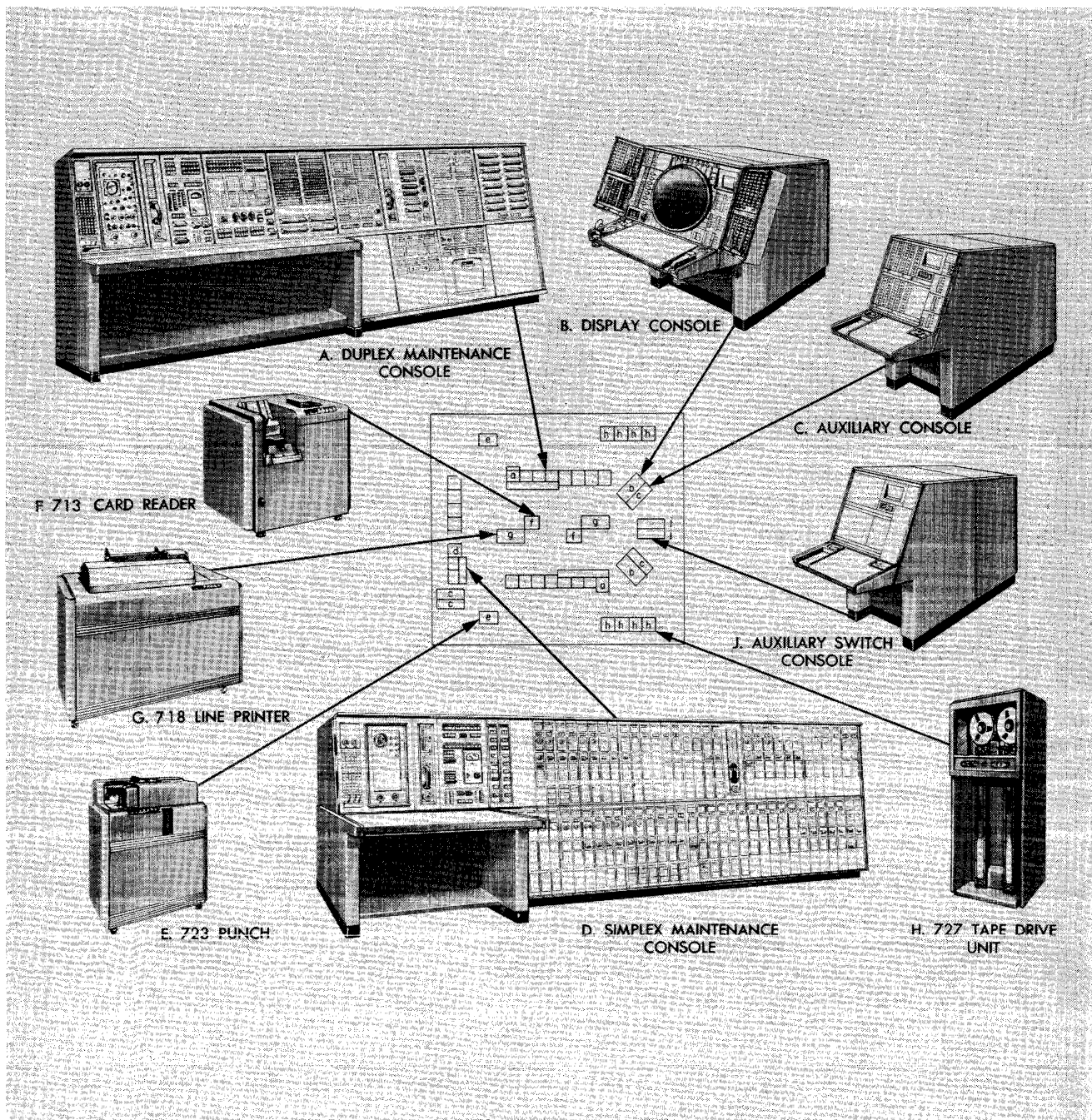


Figure 1-194. Central Maintenance Area Facilities

the simplex portion of the equipment. Only the display consoles are not monitored from the simplex maintenance console, since they have built-in monitoring and testing facilities. The simplex maintenance console does provide central control of the status of the Input System channel equipment, determining which channels are active and which are in the standby status. In addition, the simplex maintenance console contains marginal checking and power controls for input channel equipment, as well as equipment to generate test signals for

this equipment. The simplex maintenance console is described in detail in Section 3.

1.2.3 Marginal Checking

Since marginal checking involves the same principles of operation whether it is used to test duplex or simplex equipment, both duplex and simplex marginal checking controls are described in Section 4, together with a more thorough discussion of marginal checking philosophy.

SECTION 2

DUPLEX MAINTENANCE CONSOLE

2.1 GENERAL

The duplex maintenance console is divided physi-

INFORMATION NOT AVAILABLE

cally into nine modules, five of which mount single panels and four of which mount two panels. (See fig. 1-195.) These panels include:

- Central scope and probes panel
- Marginal checking control panel
- Central Computer System control panel
- Register neons panel
- Main and auxiliary drum group indicator panels
- Output test control and output and common Input System indicator panels
- Situation display and warning lights register neons and tape element and manual data input indicator panels
- Display tester control panel.

The function and use of each control and indicator is described separately in this section.

2.2 CENTRAL SCOPE AND PROBES PANEL

Information not available.

2.3 MARGINAL CHECKING CONTROL PANEL

2.3.1 General

In addition to the controls and indicators on the marginal checking control panel specifically connected with marginal checking, there are two other groups of controls and indicators on the upper section of this panel: the POWER GROUP controls and indicators and the NOT READY indicators. (See fig. 1-196.)

Figure 1-195. Duplex Maintenance Console

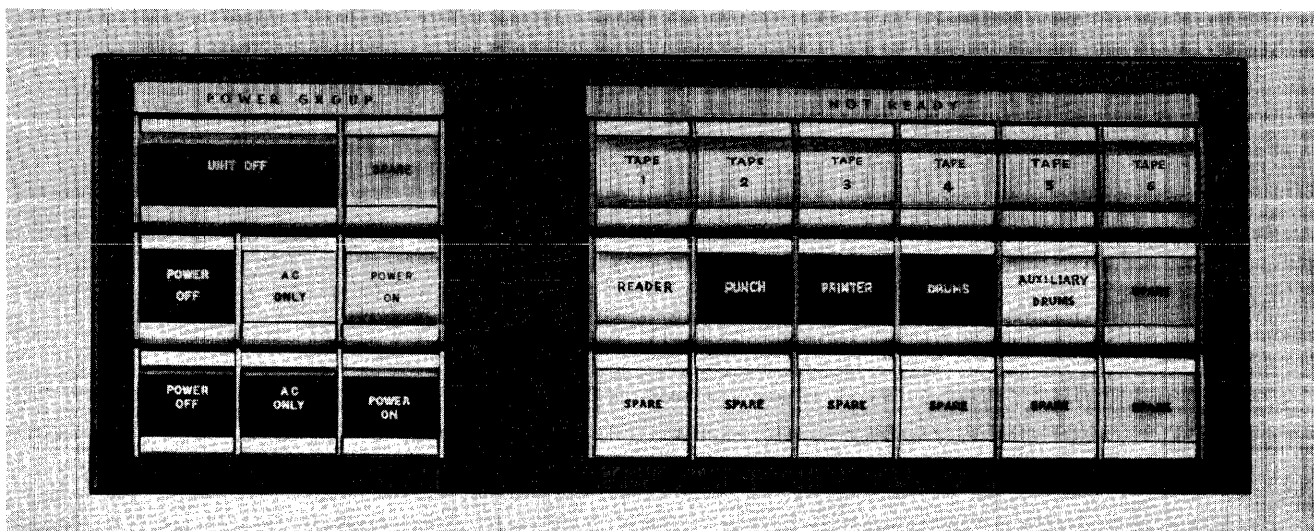


Figure 1-196. POWER GROUP and NOT READY Controls and Indicators

2.3.2 POWER GROUP Controls and Indicators

The POWER GROUP controls and indicators control the application and removal of power for the du-

plex equipment as well as give visible indication of the type of power applied to the equipment. (Refer to table 1-85.)

TABLE 1-85. POWER GROUP CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
POWER ON pushbutton		Initiates power application sequence (if motor-generator sets are running) for all load units not individually turned off.
	POWER ON light	Indicates power applied to all units as initiated by POWER ON pushbutton.
AC ONLY pushbutton		Allows application of only a-c filament voltages to load units, either initiating application sequence through a-c voltages and no further or, if d-c voltages are already applied, initiating power off sequence down to a-c voltages.
	AC ONLY light	Indicates only a-c filament power applied to all load units; off if POWER ON light is on.
POWER OFF pushbutton		Initiates normal sequence for removal of power from all load units; motor generators continue running.
	POWER OFF light	Indicates that all power supplies are turned off, no power applied to equipment.
UNIT OFF pushbutton		Removes all power from equipment without regard to sequence of removal; used for emergency power removal.

2.3.3 NOT READY Indicators

The NOT READY group of indicators provides a visible indication of input-output (IO) units not capable of carrying on IO operations with the Central Computer System. In general, a not-ready condition results from

power failures or mechanical failures of the unit involved. An audible alarm sounds if any of these lights, other than the three lights for the card machines, are on. This group of indicators is listed together with the function of each light in table 1-86.

TABLE 1-86. NOT READY INDICATORS

INDICATOR	AUDIBLE ALARM	INDICATION
TAPE 1 through TAPE 6	Yes	Indicates that tape drive unit is not ready; tape is broken, power is off, or unit is in manual control.
READER	No	Indicates card reader is not ready; power is off, fuse blown, cards not in hopper, stacker full, or reader under manual control.
PUNCH	No	Indicates card punch is not ready; power is off, fuse blown, blank cards not in hopper, stacker full, or punch under manual control.
PRINTER	No	Indicates line printer is not ready; power is off, fuse blown, paper not on platen, control panel not in, or printer under manual control.
DRUMS	Yes	Indicates that main drum group is not ready; drum motors are off, drum control dc off, or drum housing dc off.
AUXILIARY DRUMS	Yes	Indicates that auxiliary drum group is not ready; either drum motors are off or d-c power not applied.

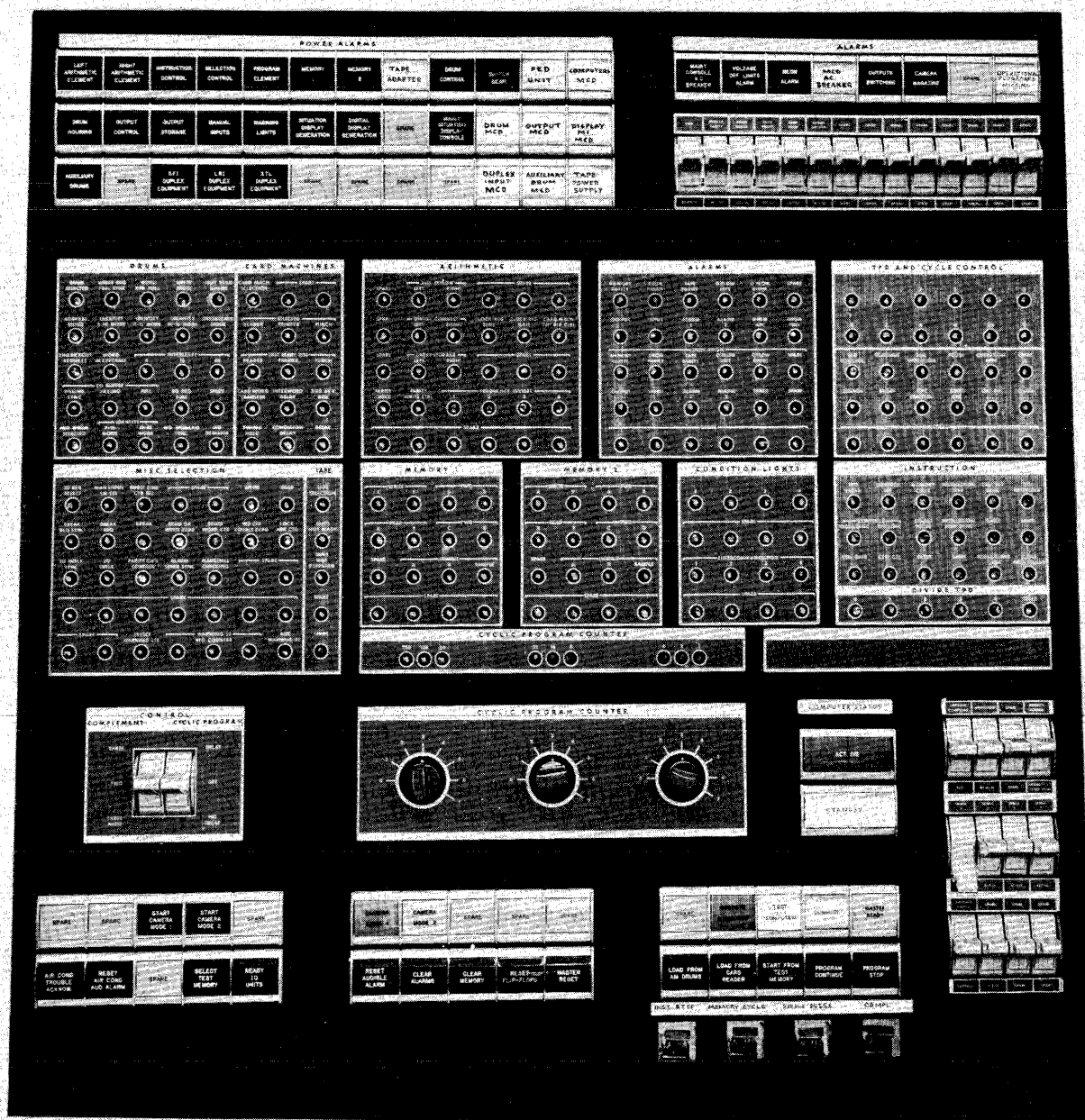


Figure 1-197. Central Computer System Control Panel

2.4 CENTRAL COMPUTER SYSTEM CONTROL PANEL

2.4.1 General

The Central Computer System control panel houses the majority of operating controls for the entire duplex equipment associated with the duplex maintenance console. In addition, indicators and manual test facilities for the Central Computer System are housed on this panel. (See fig. 1-197.) For convenience of discussion,

the controls and indicators on this panel are considered in groups, each discussed separately.

2.4.2 Power Alarms

The POWER ALARMS (a group of indicators in the upper left corner of the Central Computer System control panel) give visible indications of power faults at various units within the equipment. (See fig. 1-198.) The power fault may be lack of power, a tripped circuit breaker at a power distribution unit, or voltage off limits.

POWER ALARMS											
LEFT ARITHMETIC ELEMENT	RIGHT ARITHMETIC ELEMENT	INSTRUCTION CONTROL	SELECTION CONTROL	PROGRAM ELEMENT	MEMORY 1	MEMORY 2	TAPE ADAPTER	DRUM CONTROL	AC SWITCH GEAR	PCD UNIT	COMPUTER MCD
DRUM HOUSING	OUTPUT CONTROL	OUTPUT STORAGE	MANUAL INPUTS	WARNING LIGHTS	SITUATION DISPLAY GENERATION	DIGITAL DISPLAY GENERATION		MAINT SITUATION DISPLAY CONSOLE	DRUM MCD	OUTPUT MCD	DISPLAY MI MCD
AUXILIARY DRUMS		GFI DUPLEX EQUIPMENT	LRI DUPLEX EQUIPMENT	XTL DUPLEX EQUIPMENT					DUPLEX INPUT MCD	AUXILIARY DRUM MCD	TAPE POWER SUPPLY

Figure 1-198. POWER ALARMS Indicators

TABLE 1-87. POWER ALARMS INDICATORS

INDICATOR	AUDIBLE ALARM	INDICATION
LEFT ARITHMETIC ELEMENT	Yes	Loss of d-c power at left arithmetic element.
RIGHT ARITHMETIC ELEMENT	Yes	Loss of d-c power at right arithmetic element.
INSTRUCTION CONTROL	Yes	Loss of d-c power at instruction control element.
SELECTION CONTROL	Yes	Loss of d-c power at selection control element.
PROGRAM ELEMENT	Yes	Loss of d-c power at program element.
MEMORY 1	Yes	Loss of d-c power at memory 1 unit.
MEMORY 2	Yes	Loss of d-c power at memory 2 unit.
TAPE ADAPTER	Yes	Loss of d-c power at tape adapter unit.
DRUM CONTROL	Yes	Loss of d-c power at drum control unit.
AC SWITCHGEAR	Yes	Overload circuit breaker tripped in switchgear, removing all power; light is battery-operated.
PCD FRAME	Yes	Circuit breaker tripped, knife switch open, or a voltage is more than 5 percent off limits in the power control and distribution unit; light is battery-operated.
COMPUTERS MCD	Yes	Circuit breaker tripped in Central Computer System marginal checking distribution unit.
DRUM HOUSING	Yes	Loss of d-c power at drum housing unit.
OUTPUT CONTROL	Yes	Loss of d-c power at output control unit.
MANUAL INPUTS	Yes	Loss of d-c power at manual inputs unit.

TABLE 1-87. POWER ALARMS INDICATORS (cont'd)

INDICATOR	AUDIBLE ALARM	INDICATION
WARNING LIGHTS	Yes	Loss of d-c power at warning lights unit.
SITUATION DISPLAY GENERATOR	Yes	Loss of d-c power at situation display generator unit.
DIGITAL DISPLAY GENERATOR	Yes	Loss of d-c power at digital display generator unit.
MAINT SITUATION DISPLAY CONSOLES	Yes	Loss of d-c power at maintenance display consoles.
DRUM MCD	Yes	Circuit breaker tripped on drum MCD unit.
OUTPUT MCD	Yes	Circuit breaker tripped on output MCD unit.
DISPLAY MI MCD	Yes	Circuit breaker tripped at display and MI MCD.
AUXILIARY DRUMS	Yes	Loss of d-c power at auxiliary drum unit.
GFI DUPLEX EQUIPMENT	Yes	Loss of d-c power at GFI duplex equipment unit.
LRI DUPLEX EQUIPMENT	Yes	Loss of d-c power at LRI duplex equipment unit.
XTL DUPLEX EQUIPMENT	Yes	Loss of d-c power at XTL duplex equipment unit.
DUPLEX INPUT MCD	Yes	Circuit breaker tripped on duplex input MCD unit.
AUXILIARY DRUM MCD	Yes	Circuit breaker tripped on auxiliary drum MCD unit.
TAPE POWER SUPPLY	Yes	Tape power supply not operative.

In most cases, an audible alarm sounds if one of these indicators is on. Their exact functions are listed in table 1-87.

2.4.3 ALARMS Indicators

The indicators in the ALARMS group (located on the upper right corner of the Central Computer System control panel) give indications of various special alarm conditions. (See fig. 1-199.) The function of each of these indicators is given in table 1-88.

2.4.4 Alarm Control Switches

2.4.4.1 General

A number of program errors whose effects can be detected by checking circuits and certain types of equipment malfunctions which make reliable operation of a computer difficult or impossible are reported to the duplex maintenance console as alarms. These alarms include memory parity, drum parity, tape parity, left over-

flow, right overflow, Output System (which, in turn, includes output drum parity, defective transmission, illegal section or register address, or nonsearch comparison), other computer alarm 1 (memory parity or overflow alarm in other computer), other computer alarm 2 (drum or tape parity alarm in other computer), and duplex switch alarm, which indicate that this computer has just been switched to active status.

All of these alarms set sense units whose statuses can be used to control the execution of programs through *BSN (u) x* instructions. In addition, five of these alarms can be used directly (without a *Sense* instruction) to control the course of execution of a program. The exact action taken upon detection of one of these five alarm conditions is determined by a group of alarm control switches located on the Central Computer System control panel directly beneath the ALARMS indicators. (See fig. 1-200.)

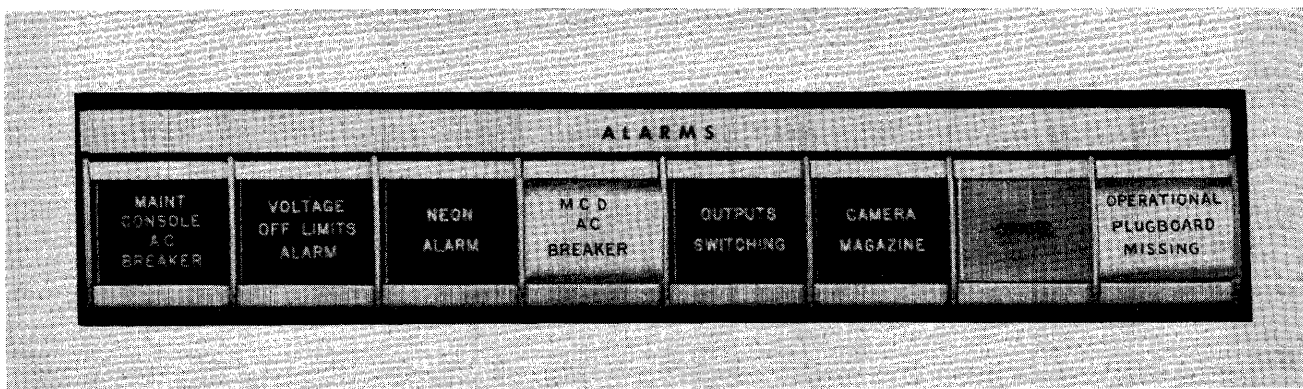


Figure 1-199. ALARMS Indicators

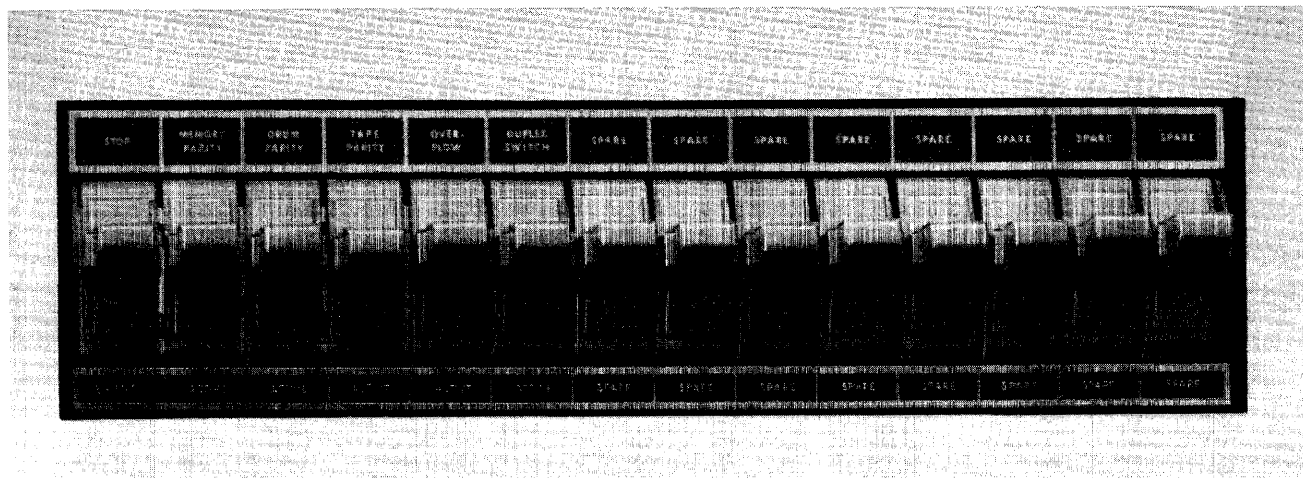


Figure 1-200. Alarm Activation Switches

TABLE 1-88. ALARMS INDICATORS

INDICATOR	AUDIBLE ALARM	INDICATION
MAINT CONSOLE AC BREAKER	No	A-c circuit breaker on marginal checking control panel of duplex maintenance console is open.
VOLTAGE OFF LIMITS ALARM	Yes	One of d-c voltages in PCD is off limits by more than 5 percent.
NEON ALARM	Yes	Alarm flip-flop is on in one of these units: selection control, drums, outputs or inputs. (Refer to 2.4.5.)
MCD AC BREAKER	No	Circuit breaker on MCD supplying voltages to duplex maintenance console is tripped.
OUTPUTS SWITCHING	No	Telephone output channels have not switched over to receive signals from active Output System.
CAMERA MAGAZINE	Yes	Less than 5 feet of film is left in magazine of automatic camera described in Chapter 6.
OPERATIONAL PLUGBOARD MISSING	No	Plugboard with special interlock wiring is not in test memory jack field. (Refer to 2.6.)

TABLE 1-89. ALARMS NEONS

INDICATOR NEON	INDICATION
MEMORY PARITY	Error has occurred in transferring a word from core memory to the memory buffer register.
DRUM PARITY	Error has occurred in transferring a word from drums to memory buffer register.
TAPE PARITY	Error has occurred in transferring a word from tape to memory buffer register.
FLOW LEFT	Overflow condition has occurred in left accumulator register.
FLOW RIGHT	Overflow condition has occurred in right accumulator register.
OUTPUT	Error has occurred in Output System.
INPUT	Information not available.
ALARM 1	Drum or tape parity alarm in the other computer.
ALARM 2	Memory parity or right or left overflow condition exists in the other computer.
DRUM APC	Drum APC out of sync.
DRUM FULL	One of six drum fields in main drums full.
MEMORY PARITY (INDICATION)	Memory parity alarm has been given; remains on until cleared by CLEAR ALARM pushbutton.
DRUM PARITY (INDICATION)	Drum parity alarm has been given; remains on until cleared by CLEAR ALARM pushbutton.
TAPE PARITY (INDICATION)	Tape parity alarm has been given; remains on until cleared by CLEAR ALARMS pushbutton.
O'FLOW LEFT (INDICATION)	Overflow in left accumulator has occurred; remains on until cleared by CLEAR ALARM pushbutton.
O'FLOW RIGHT (INDICATION)	Overflow in right accumulator has occurred; remains on until cleared by the CLEAR ALARMS pushbutton.
OUTPUT (INDICATION)	Output System alarm has occurred; remains on until cleared by CLEAR ALARMS pushbutton.
INPUT (INDICATION)	Information not available.
ALARM 1 (INDICATION)	Drum parity or tape parity alarm has been given in the other computer; remains on until cleared by CLEAR ALARMS pushbutton.
ALARM 2 (INDICATION)	Memory parity or right or left overflow alarm has been given in the other computer; remains on until cleared by CLEAR ALARMS pushbutton.

2.4.4.2 Alarm Activation Switches

Excluding the leftmost switch in the row, the five switches which direct that some action be taken in response to specific alarms are:

- MEMORY PARITY (inactive)—ACTIVE switch
- DRUM PARITY (inactive)—ACTIVE switch
- TAPE PARITY (inactive)—ACTIVE switch
- OVERFLOW (inactive)—ACTIVE switch

e. DUPLEX SWITCH (inactive)—ACTIVE switch.

If one of these alarms is generated and the corresponding switch is in the inactive position, the existence of that alarm is ignored (unless the program being executed includes provisions for sensing the appropriate sense unit). However, if the alarm is generated when the corresponding switch is in the ACTIVE position, the Central Computer System either stops or branches to location 20,010₈, depending upon the position of the first switch in the row, the STOP-BRANCH switch.

2.4.4.3 STOP-BRANCH Switch

When this switch is in the STOP position and an alarm signal is applied to it through an alarm activation switch, the Central Computer System program is stopped and any IO transfer in progress is interrupted. Usually, the IO unit attempts to complete the transfer and thus steps the IO word counter normally; however, no break-in cycles are executed, thus discarding any words read from the IO unit. If the IO unit is a drum field read by status, the words are lost permanently. If the IO operation is a writing operation on a drum field, the registers written after the alarm interruption contain positive 0. If a tape reading or writing operation is interrupted by a memory or tape parity alarm, transfer stops immediately and the tape unit is deselected. Further, if this occurs while a tape unit is reading, the tape unit stops immediately, not at the end of a record.

If the STOP-BRANCH switch is in the BRANCH position and an alarm signal is applied to it, the program is stopped immediately whether or not the execution of the current instruction is completed. Any IO operation in progress is stopped immediately by deselecting the IO unit. The contents of the program counter are transferred to the right A register, and all registers and controls are cleared except the right A register, the IO word counter, and alarm indicators and sense units. Finally, the program counter is loaded with 20,010₈, causing program execution to continue with the routine stored in test memory.

2.4.4.4 Applications

The alarm control switches provide versatility in response to alarm conditions, since they augment the use of alarm sensing by *Sense* instructions. The alarm control switches can be used to give immediate response to alarms whereas the *Sense* instruction can only indicate the generation of a particular alarm at some instant between the sensing which detects the prior existence of that alarm and a previous sensing at which time no alarm had been indicated. For example, either *BSN* (30) 0.20010 or placing the DUPLEX SWITCH alarm control switch in the ACTIVE position along with the STOP-BRANCH switch in the BRANCH position accomplishes the same result, branching to test memory when the computer is switched to active status. Test memory may contain a start-over routine to load the air defense program. If the program being executed by the computer while in standby is critical enough to delay switchover until that program reaches a convenient stopping place, the *Sense* instruction can be used to detect switchover and initiate the air defense program. If the program being run by the standby computer is not critical, the switch settings can be used to detect switchover and immediately initiate the air defense program.

2.4.5 ALARMS Neon Group

The neons in the ALARMS group (located in the first row of neon blocks, second block from the right) give indications of various special alarm conditions that may occur during the operation of the Central Computer System. (See fig. 1-201.) These alarm conditions may include an error in the transfer of a word from core memory, the Drum System, or the tapes to the memory buffer register. Other causes for an alarm condition are an overflow condition in the left or right accumulator or an error detected in the Output System. In most cases, if an alarm occurs, a neon indicator remains illuminated until cleared by the CLEAR ALARMS pushbutton. The function of each neon is given in table 1-89.

2.4.6 CONDITION LIGHT Neon Group

The neons in the CONDITION LIGHT group are located in the second row of neon blocks, second block from the right. (See fig. 1-201.) Each CONDITION LIGHT neon, 1, 2, 3, or 4, is turned on by a separate *Operate (PER)* instruction selection code. If the status of a CONDITION LIGHT is examined by a specific *Sense (BSN)* instruction while the light is on, the CONDITION LIGHT is extinguished and the program branches. If the light is off, no branch occurs. An illuminated INTERCOMMUNICATION neon (either 1, 2, 3, or 4) indicates that its respective flip-flop in this computer (the one associated with this duplex maintenance console) has been set by the other computer. If sensed by this computer, a branch operation is performed.

2.4.7 TPD AND CYCLE CONTROL Neon Group

The neons in the TPD AND CYCLE CONTROL group (located in the first row of neon blocks, last block on the right) indicate various operations performed by the TPD AND CYCLE CONTROL. (See fig. 1-202.) The neons are used primarily during testing since, during operations, their indications change too rapidly to be perceived visually.

The 12 time pulse distributor (TPD) neons, located on the upper portion of the TPD AND CYCLE CONTROL neon group, indicate which time pulse (TP) or instruction pulse (IP) is to be propagated by the TPD. One neon illuminates at a time in the sequence 0 through 11. Because of the speed of the equipment, these neons cannot be seen changing their status. The CYCLE B, OPERATION TIME, PAUSE, and 2-MC OPERATION, located under the 12 TPD neons, indicate the type of operation taking place during the machine cycle. The function of each neon is given in table 1-90.

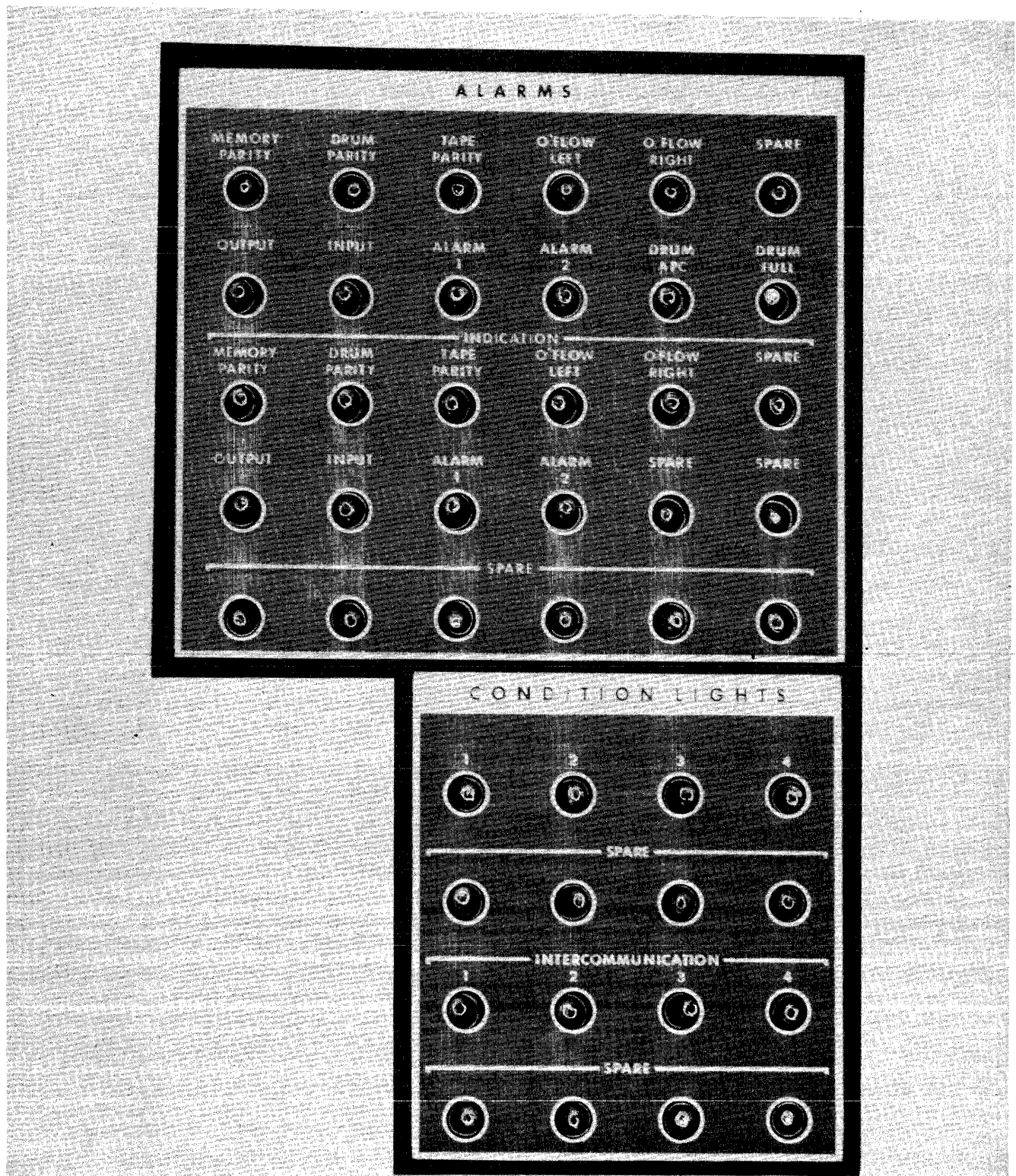


Figure 1-201. ALARMS and CONDITION LIGHTS Neons

2.4.8 INSTRUCTION Neon Group

The neons in the INSTRUCTION group (located in the second row of neon blocks, last block on the right) indicate various operations performed by the Cen-

tral Computer System during the execution of an instruction. (See fig. 1-202.) These neons are used primarily during testing since, during operations, their indications change too rapidly to be perceived visually. The function of each neon is given in table 1-91.

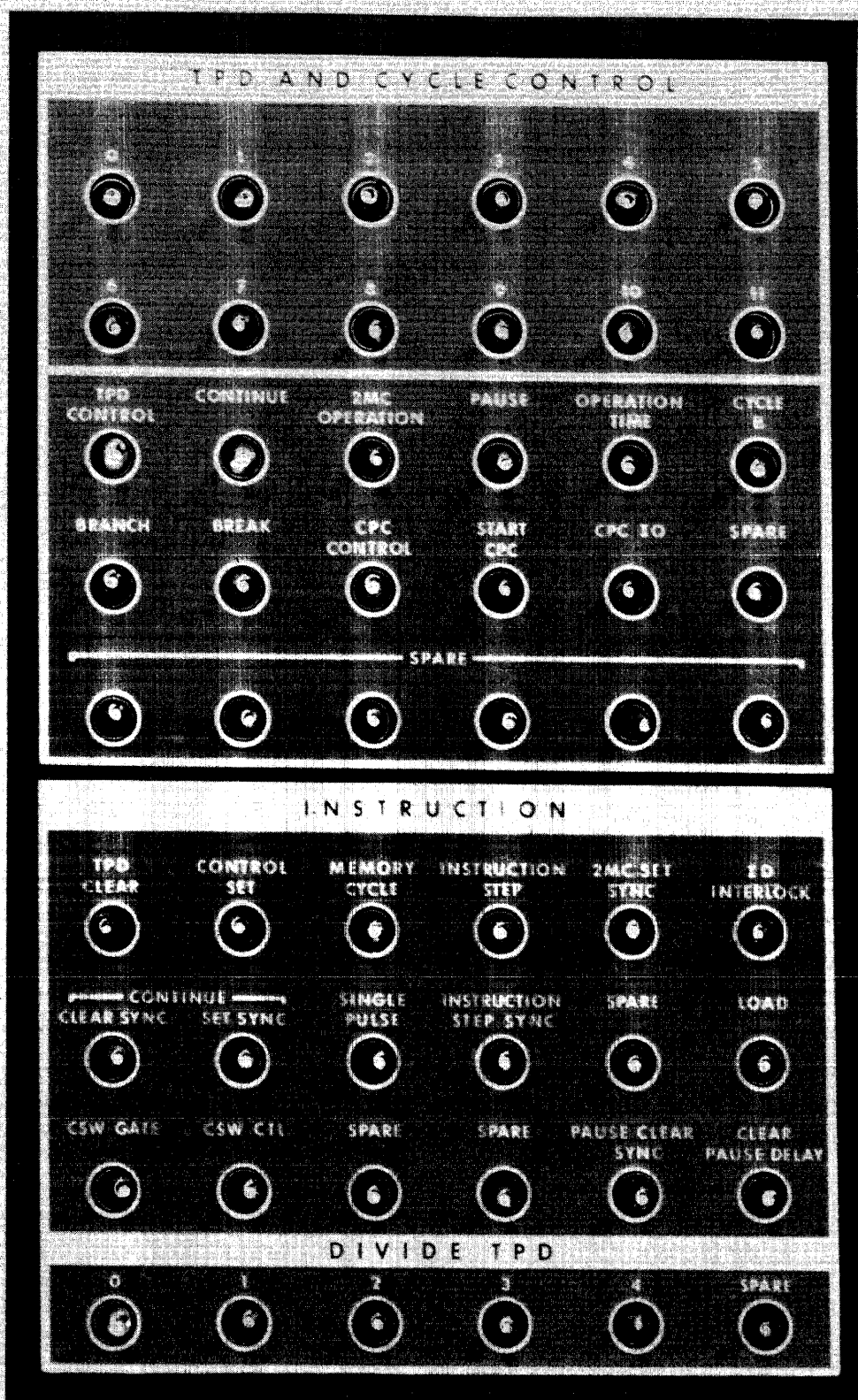


Figure 1-202. TPD AND CYCLE CONTROL and INSTRUCTION Neons

TABLE 1-90. TPD AND CYCLE CONTROL NEONS

INDICATOR NEON	INDICATION
0 through 11	TPD is propagating TP 0 through 11 or IP 0 through 11.
TPD CONTROL	Normally on during automatic operation; indicates that 2-mc pulses are available.
CONTINUE	When computer is started, indicates that TPD shall continue to develop time pulses; off only during an alarm condition or program stop.
2-MC OPERATION	Indicates 2-mc pulses are being supplied to the command generators during a pause.
PAUSE	A pause is in progress; the TPD is off unless a break is in progress.
OPERATION TIME	The Central Computer System is in an operate time cycle; when off, the Central Computer System is in program time.
CYCLE B	The Central Computer System is in an operate time B cycle; off during operation time A; meaningful only during the operate time cycle.
BRANCH	Conditions necessary to perform a branch instruction have been met.
BREAK	Break is being performed; the TPD has been started; instruction pulses are suppressed.
CPC CONTROL	The CPC control flip-flop has been set by means of the CPC-DELAY-NO DELAY switch and remains on during one cycle of the CPC; turned off by the end-carry pulse of the CPC.
START CPC	Start CPC flip-flop (FF) has been set by the CPC-DELAY-NO DELAY switch in conjunction with 2-mc pulses or the clock register; this flip-flop (FF) conditions a gate tube which passes the next 2-mc pulse or clock register pulse which starts the cyclic program.
CPC-IO	When off, CPC-IO flip-flop (FF) has been cleared to 0 by PT 2-halt, thus conditioning gate tubes to pass pulses to step the CPC; when on, IO operation is in progress during a cyclic program; the CPC is not being stepped during this IO operation.

TABLE 1-91. INSTRUCTION NEONS

INDICATOR NEON	INDICATION
TPD CLEAR	The TPD control clear flip-flop is in the 1 condition and is ready to clear the TPD control flip-flop.
CONTROL SET	The TPD control set FF is in the 1 condition and is ready to set the TPD control FF.
MEMORY CYCLE	Memory cycle FF has been set by the memory cycle switch; FF conditions circuits which at IP 5 set the TPD CTL CLR sync FF, clear the continue FF, and clears itself.
INSTRUCTION STEP	Instruction step FF has been set by instruction step, stop, or memory cycle switches, or by PT 1 of <i>Read</i> or <i>Write</i> instruction in conjunction with a single pulse or cyclic program; FF conditions circuits which at PT 5 set the TPD CTL CLR sync FF, clear the continue FF, and, after 1.0-microsecond delay, sets the TPD set sync FF and clears itself.
2 MC SET SYNC	The 2-mc operation set sync FF is in the 1 position and ready to synchronize the 2-mc operate FF with the 2-mc pulses.
IO INTERLOCK	An IO operation is being performed; no new IO instruction can be started until the one in progress is completed.

TABLE 1-91. INSTRUCTION NEONS (cont'd)

INDICATOR NEON	INDICATION
CLEAR SYNC	Synchronizing FF for the clear continue FF is in the 1 condition and ready to sync the clear continue FF with the 2-mc pulses.
SET SYNC	Synchronizing FF for the set continue FF is in the 1 condition and ready to sync the set continue FF with the 2-mc pulses.
SINGLE PULSE	Single pulse FF has been 1 and the computer will advance the program one time pulse for each depression of the SINGLE PULSE pushbutton; operation possible only when the computer is in test status.
INSTRUCTION STEP SYNC	Instruction step sync FF has been set by instruction step, stop, or memory cycle switches and is ready to sync the instruction step FF to PT 5.
LOAD	Load FF has been set by means of the LOAD FROM CARD READER or LOAD FROM AM DRUM switches and indicates the loading of memory from the selected IO unit.
CSW GATE	At PT 9 of a CSW instruction, the step-word-control pulse can set the CSW control FF; at TP 0, neon goes off (the CSW control and CSW gate FF's are used to prevent transfer of word counter contents to the right accumulator register if the word counter is being stepped).
CSW CTL	On a CSW instruction, if a step-word-counter pulse comes in after IP 9 and before TP 0; in this case, CSW is carried out at IP 5; if the step-word-counter pulse does not come through before TP 0, neon remains off.
PAUSE CLEAR SYNC	Synchronizing FF for the pause clear FF is in the 1 condition and ready to sync the pause clear FF with 2-mc pulses.
CLEAR PAUSE DELAY	Clear pause delay FF has been set by receiving a clear-pause pulse and on the next TP generates a signal to clear the pause FF.

TABLE 1-92. ARITHMETIC NEONS

INDICATOR NEON	INDICATION
AUX O'FLOW LEFT	Overflow condition may have occurred in the left accumulator; if the overflow condition is verified, the overflow neon is turned on, otherwise it is not; in either case, this neon is immediately turned off.
AUX O'FLOW RIGHT	Overflow condition may have occurred in the right accumulator register; if the overflow condition is verified, the overflow neon is turned on, otherwise it is not; in either case, this neon is immediately turned off.
DIVIDE CONNECT LEFT	During a <i>Divide</i> instruction, indicates that the carry 1 and carry 0 lines of the left accumulator register sign bit adder have been connected to the bit 15 FF in the left B register.
DIVIDE CONNECT RIGHT	During a <i>Divide</i> instruction, indicates that the carry 1 and carry 0 lines of the right accumulator sign bit adder have been connected to the bit 15 FF in the right B register.
CLOCK REG SYNC	Turned on by the output of the frequency divider circuit 32 times per second; conditions a gate tube which allows TP 11 or 2-mc pulses to set the clock reg gate FF; the second TP 11 or second 2-mc pulse turns this light off.
CLOCK REG GATE	First TP 11 pulse or first 2-mc pulse after the clock reg sync FF has been set; the second TP 11 or second 2-mc pulse turns this light off and steps the clock register once.
CARD MACH THY BFR CTRL	Even words are being transferred to card machines during printer or punch operation.

TABLE 1-92. ARITHMETIC NEONS (cont'd)

INDICATOR NEON	INDICATION
CARRY STORAGE LEFT	In the rounding-off operation, a 1 is being added to the positive number in the left accumulator register or subtracted from the negative number in the left accumulator register.
CARRY STORAGE RIGHT	In the rounding-off operation, a 1 is being added to the positive number in the right accumulator register or subtracted from the negative number in the right accumulator register.
PARITY CHECK	On at the start of a parity count; if the count is correct, this neon goes off; if the count is incorrect, it stays on until cleared by RESET FLIP-FLOPS switch.
PARITY WRITE CTRL	Parity write CTRL FF is set on; FF gates a tube which allows a 1 to be placed in the left memory buffer register parity bit FF if the parity count is even.
FREQUENCY DIVIDER 1 through 4	Indicates the status of a 4-stage binary counter which counts the 512-cps pulses produced by the tuning fork oscillator.

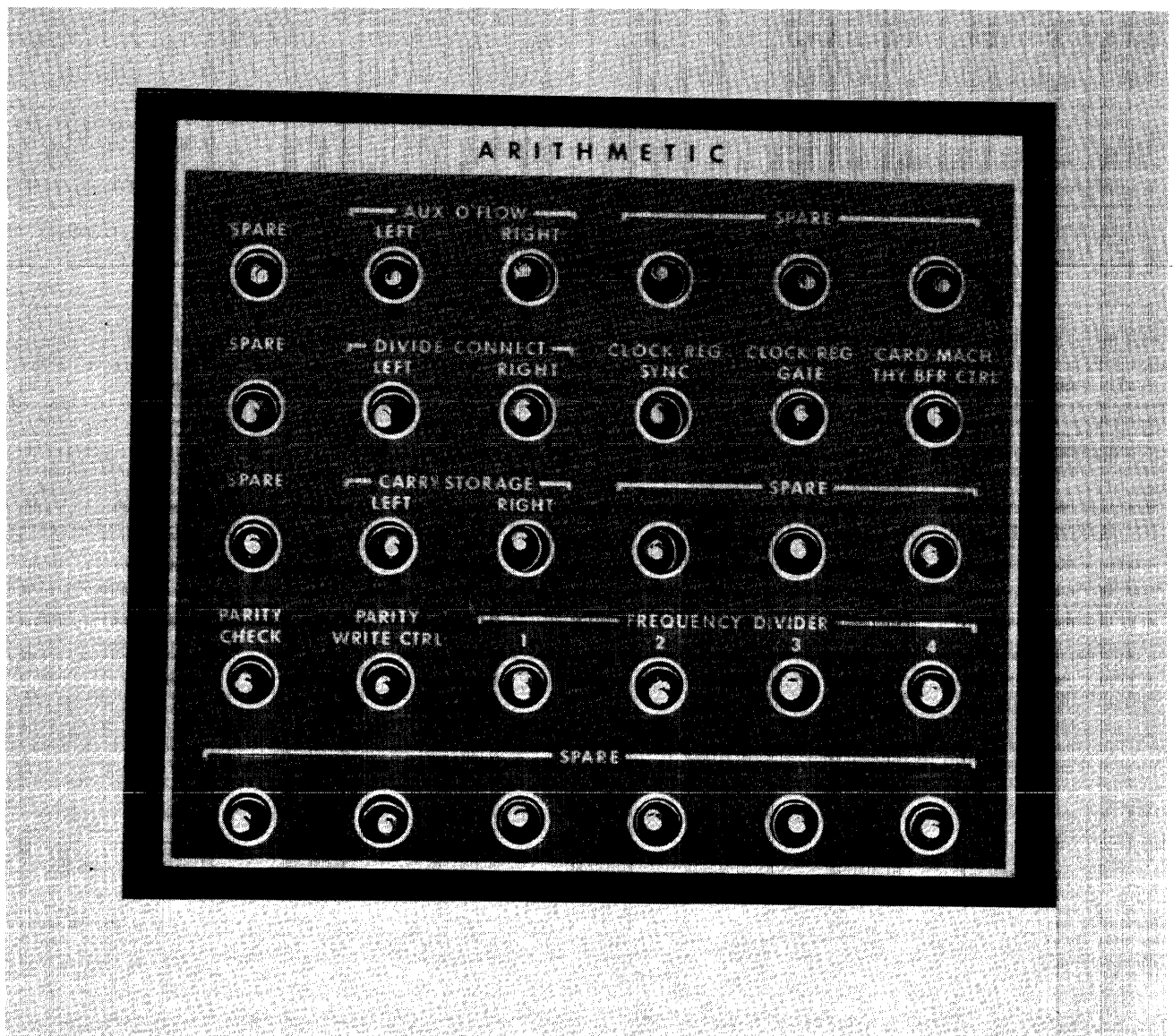


Figure 1-203. ARITHMETIC Neons

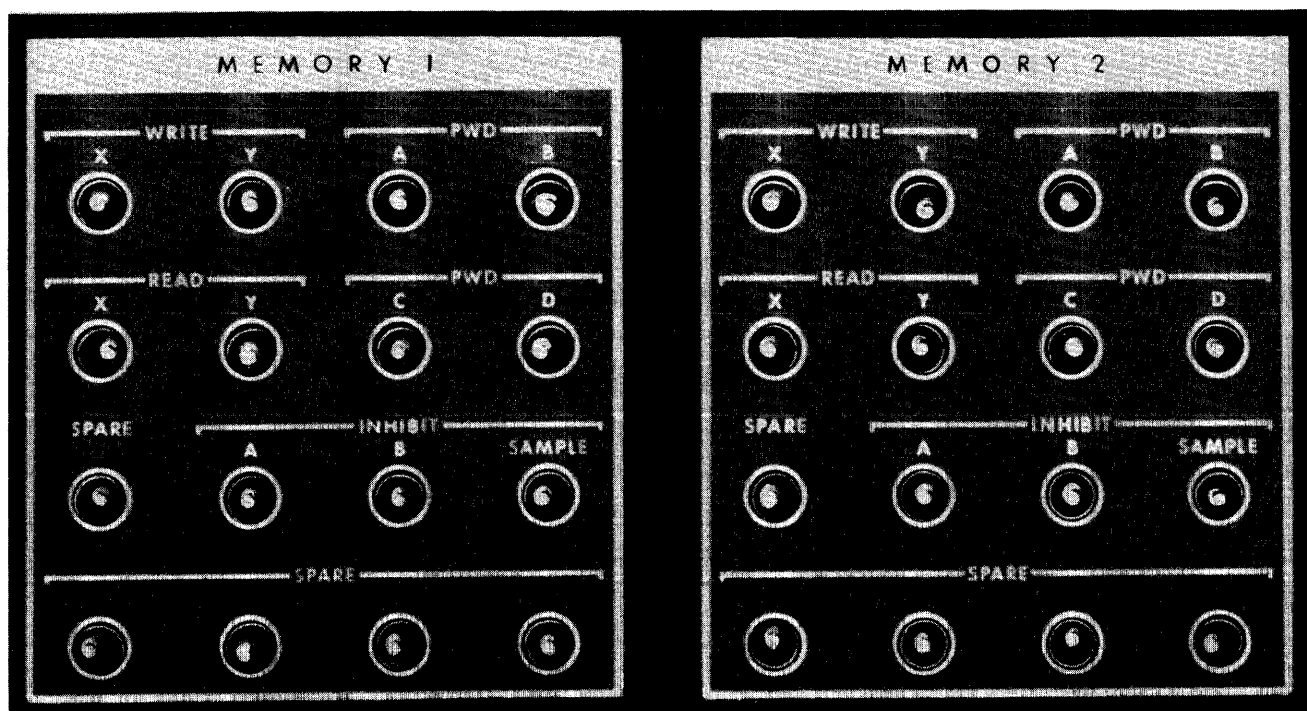


Figure 1-204. MEMORY 1 and MEMORY 2 Neons

TABLE 1-93. MEMORY 1 AND MEMORY 2 NEONS

INDICATOR NEON	INDICATION
WRITE X	Write pulse is being applied to the X direction lines of the core array in memory 1 or 2.
WRITE Y	Write pulse is being applied to the Y direction lines of the core array in memory 1 or 2.
PWD A and B	Not used at present.
READ X	Read pulse is being applied to the X direction lines of the core array in memory 1 or 2.
READ Y	Read pulse is being applied to the Y direction lines of the core array in memory 1 or 2.
PWD C and D	Not used at present.
INHIBIT A	Inhibit pulse is being applied to the digit plane driver for writing the left half-word in memory 1 or 2.
INHIBIT B	Inhibit pulse is being applied to the digit plan driver for writing the right half-word in memory 1 or 2.
INHIBIT SAMPLE	A command has been given to prevent reading out of core memory 1 or 2 when a word is in the memory buffer register waiting to be written into memory 1 or 2.

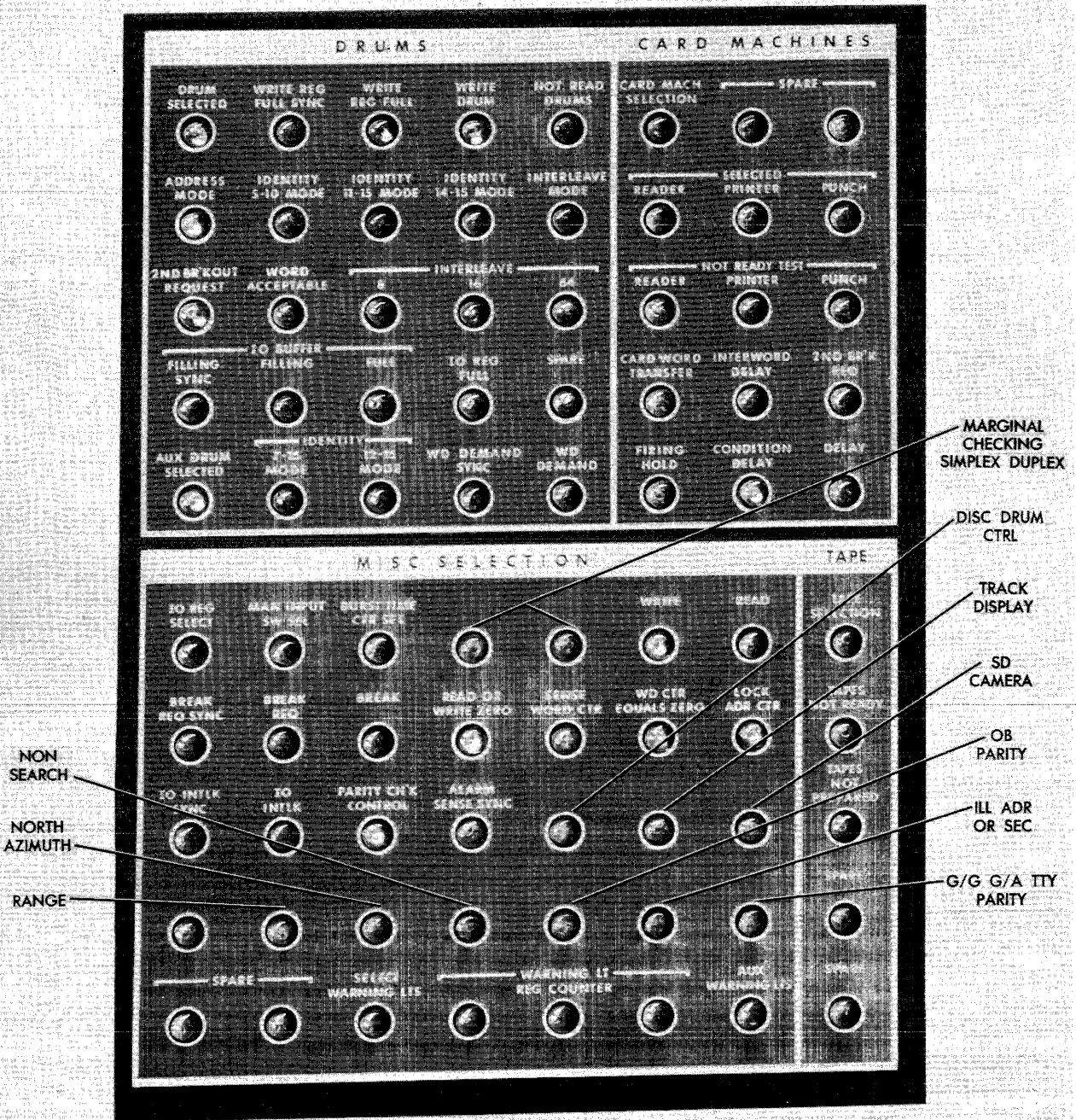


Figure 1-205. DRUMS, CARD MACHINES, MISC SELECTION and TAPES Neons

2.4.9 DIVIDE TPD Neons Group

The five neons (labelled 0 through 4) in the DIVIDE TPD group (located in the second row of neon blocks, last block on the right) indicate the status of the five flip-flops in the TPD timing ring. Each neon remains on until a *Divide* instruction is received, whereupon it cycles with the timing ring.

2.4.10 ARITHMETIC Neon Group

The neons in the ARITHMETIC group (located in the first row of neon blocks, second block from the

left) indicate the status of various circuits in the arithmetic element. (See fig. 1-203.) In addition, the CLOCK REG SYNC and CLOCK REG GATE neons and the FREQUENCY DIVIDER 1 through 4 neons indicate the status of their respective flip-flops in the real time clock. The neons in the ARITHMETIC group are used primarily during testing since, during normal operations, their indications change too rapidly to be perceived visually. The function of each neon is given in table 1-92.

TABLE 1-94. DRUMS NEONS

INDICATOR NEON	INDICATION
DRUM SELECTED	Drums are being used for IO operations; turned on by SDR instruction or by a manual load from the AM-A pushbutton.
WRITE REG FULL SYNC	Each time a word is written on drums, a word-demand-delay pulse turns this neon on; it will be turned off as soon as a word is placed in the write register.
WRITE REG FULL	Word to be written on drums has been transferred from the IO register to the write register.
WRITE DRUM	Drums are being used for output operations; turned on at PT 6 of the <i>Write</i> instruction and remains on until cleared by the word counter end carry.
NOT READ DRUMS	Drums are being used for output operations; will remain on during writing operations until IO word ctr = 0 and the write register is empty; at TP 11, IO interlock is cleared and neon goes off.
ADDRESS MODE	Drums are reading or writing by the address mode.
IDENTITY 5-10 MODE	Drums are reading or writing by identity 5-10 mode.
IDENTITY 11-15 MODE	Drums are reading or writing by identity 11-15 mode.
IDENTITY 14-15 MODE	Drums are reading or writing by identity 14-15 mode.
INTERLEAVE MODE	Addressable drums are reading or writing by interleave 8, 16, or 64.
2ND BR'KOUT REQUEST	At PT 6 of the <i>Write</i> instruction; during the first breakout cycle, the word to be written on drums is transferred from memory to the IO register; when the word is transferred from the IO register to the write register, a pulse is sent through a gate tube, gated by this FF; this initiates a 2nd BO cycle and clears this FF; henceforth, the transfer of words is continued by word-demand pulses.
WORD ACCEPTABLE	Favorable comparison has been made between the word in the drum control register and the word in the IO buffer register; indicates that the transfer of the word between computer and drums is to proceed.
INTERLEAVE 8	Interleave by 8 FF has been set by the selection and IO control and causes the drum control register to count by 8 when <i>step drum control</i> command is received; remains on until cleared by the selection control unit.
16	Interleave by 16 FF has been set by the selection and IO control unit and causes the drum control register to count by 16 when <i>step drum control</i> command is received; remains on until cleared by the selection control unit.
64	Interleave by 64 FF has been set by the selection control unit and causes the drum control register to count by 64 when <i>step drum control</i> command is received; remains on until cleared by the selection control unit.
IO BUFFER FILLING SYNC	The IO buffer load sync has been set and is ready to sync the IO buffer load FF with the 2-mc pulses.
FILLING	The IO buffer load FF has been set and the IO buffer is being loaded.
FULL	The IO buffer status FF has been set and the IO buffer is full; FF is set by a pulse from the IO buffer load and remains on until cleared by the deselect pulse.
IO REG FULL	The IO register contains a word; during a transfer between IO devices and memory, this neon and the IO buffer full neon show the complement of each other.
AUX DRUM SELECTED	Auxiliary drums have been selected for IO operations.
IDENTITY 7-15 MODE	Drums are reading by identity 7-15 mode.

TABLE 1-94. DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
IDENTITY 12-15 MODE	Drums are reading by identity 12-15 mode.
WD DEMAND SYNC	Word demand sync FF has been set by a word-demand pulse from the drums; cleared by the step-word-counter pulse.
WD DEMAND	Word demand FF has been set by the word demand sync FF, thus gating a gate tube which sends a pulse to step the word counter, to clear the word demand FF, and to clear the word demand sync FF.

TABLE 1-95. CARD MACHINES NEONS

INDICATOR NEON	INDICATION
CARD MACHINE SELECTION	One of the three card machines (reader, printer, or punch) has been selected for IO operation.
SELECTED	
READER	Reader is the card machine selected for input operations.
PRINTER	Printer is the card machine selected for output operations.
PUNCH	Punch is the card machine selected for output operations.
NOT READY TESTS	
READER	Reader is selected; remains on until cleared by the deselect pulse. (FF must be on to set the sense sync FF if the reader is not ready and a <i>Sense</i> instruction is given.)
PRINTER	Printer is selected; remains on until cleared by the deselect pulse. (FF must be on to set the sense sync FF if the printer is not ready and a <i>Sense</i> instruction is given.)
PUNCH	Punch is selected; remains on until cleared by the deselect pulse. (FF must be on to set the sense sync FF if the punc is not ready and a <i>Sense</i> instruction is given.)
CARD WORD TRANSFER	During transfer of words to printer or punch; turned off when the IO interlock is cleared.
INTERWORD DELAY	Even word is being transferred to the printer or punch; goes on when the control grid voltage for the odd word is dropped and remains on until the control grid voltage for the even word is dropped.
2ND BR'K REQUEST	Second break-request FF has been set by the request-first-break signal from the card reader and has conditioned circuits so that a card machine second break-request pulse is generated at BI 9.
FIRING HOLD	Thyratron control grids are being conditioned during printer or punch operation; neon on for only 12 microseconds for each word transferred.
CONDITION DELAY	Condition delay FF is on; FF is used to introduce a 6- to 12-microsecond delay while the thyratron shield grids are being conditioned for printer or punch operation.
DELAY	Delay FF is on; FF is used to introduce a 6-microsecond delay at various times in the process of transferring a word to the printer or punch.

2.4.11 MEMORY 1 and MEMORY 2 Neon Groups

The neons in the MEMORY 1 and MEMORY 2 groups are located in the second row of neon blocks, second and third blocks from the left. (See fig. 1–204.) These neons specifically indicate whether a read or write operation on the X or Y axis is taking place, and also if an inhibit pulse is being applied to the digit plane drivers when writing the left and/or right half-word. The function of each neon is given in table 1–93.

2.4.12 DRUMS Neon Group

The neons in the DRUMS group are located in the first row of neon blocks, last block on the left. (See fig. 1–205.) The DRUMS SELECTED, WRITE REG FULL SYNC, WRITE REG FULL, WRITE DRUM, and NOT READ DRUMS neons indicate the type of operation the drums are involved in and also the status of the drums during this operation. The ADDRESS MODE, IDENTITY 5-10 MODE, IDENTITY 11-15 MODE, IDENTITY 14-15 MODE, and the INTERLEAVE MODE neons indicate the drum reading or writing mode. The remaining neons in this group indicate circuit conditions during transfer operations between the Central Computer System and the Drum Sys-

tem. The function of each neon is given in table 1–94.

2.4.13 CARD MACHINES Neon Group

The neons in the CARD MACHINES group are located in the first row of neon blocks, to the right of the DRUMS neon group. (See fig. 1–205.) The neons in this group indicate the conditions that exist when one of the card machines is selected as an IO device. The CARD MACHINE SELECTION, READER SELECTED, PRINTER SELECTED, and PUNCH SELECTED neons specifically indicate which card machine is to be used as the IO device. The function of each neon is given in table 1–95.

2.4.14 MISC SELECTION Neon Group

The neons in the MISC SELECTION group are located in the second row of neon blocks, last block on the left. (See fig. 1–205.) These neons indicate the contents of their respective flip-flops during miscellaneous operations performed by the Central Computer System. The IO REG SELECTED, MAN INPUT SW SEL, BURST TIME CTR SEL, and the SELECT WARNING LTS neons indicate the selected IO unit. The function of each neon is given in table 1–96.

TABLE 1–96. MISC SELECTION NEONS

INDICATOR NEON	INDICATION
IO REG SELECT	The IO register has been selected as an IO unit.
MAN INPUT SW SEL	The manual input matrix has been selected as an input unit.
BURST TIME CTR SEL	Burst time counter has been selected as an input unit.
MARGINAL CHECKING	
SIMPLEX	On when Computer is performing marginal checking on simplex equipment.
DUPLEX	On when computer is performing marginal checking on duplex equipment.
WRITE	The write FF has been set; for a true break-out operation, this neon and the one associated with the break FF must both be on.
READ	The read FF has been set; for a true break-in operation, this neon and the one associated with the break FF must both be on.
BREAK REQ SYNC	Break-request sync FF has been set by a break request from an IO unit and is ready to sync the break-request FF with the 2-mc pulses.
BREAK REQ	Break-request FF has been set by a break request from any IO unit, thus conditioning a gate tube to produce a set-break-FF signal at TP 11.
BREAK	A break cycle is in progress; when accompanied by the write neon, a breakout is indicated; when accompanied by the read neon, a break-in is indicated.
READ OR WRITE ZERO	The read or write zero FF is set, conditioning a gate tube so that a word-counter-equal-to-zero signal clears the IO interlock; when off, the IO interlock is not cleared by this signal.
SENSE WORD CTR	At PT 3 when a <i>Sense</i> instruction has been programmed; indicates that the sense word counter FF has been set and is sensing the word counter for status (= 0 or \neq 0).

TABLE 1-96. MISC SELECTION NEONS (cont'd)

INDICATOR NEON	INDICATION
WD CTR EQUALS ZERO	Word counter has been stepped to 0 which, in turn, has cleared the IO interlock, indicating that a transfer operation has been completed.
LOCK ADR CTR	Address counter has been locked by a <i>PER</i> (75) ₈ instruction.
IO INTLK SYNC	The IO interlock sync FF is in the 1 position and ready to synchronize the IO interlock FF with 2-mc pulses.
IO INTLK	An IO instruction is being performed; while on, no new IO instruction can be started until the one in progress is completed.
PARITY CH'K CONTROL	The parity check control FF has been set by the deselect pulse (OT 5) and indicates that the input unit selected has a parity bit associated with each word; in this state, this FF conditions circuits which initiate a parity check; when off, the parity check control FF has been cleared by the selection of an input unit which has no parity bit associated with each word, and conditions circuits which initiate a parity count and the writing of a parity bit when required.
ALARM SENSE SYNC	One of the alarms has occurred and has been sensed at TP 7; this FF (sense sync) produces an output which may be sensed at OT 9 to set the branch FF.
DISC DRUM CTRL	On when a maximum of three disconnect pulses have been sent to drums.
TRACK DISPLAY	On when the Display System is displaying TD information; off when RD information is being displayed; sense unit checked by <i>BSN</i> (37) ₈ is associated with this neon.
SD CAMERA	On when SD camera is photographing a display; sense unit checked by <i>BSN</i> (35) ₈ is associated with this neon.
RANGE	On when pattern generator has generated a GFI range pulse; turned off if associated sense unit is checked by <i>BSN</i> (34) ₈ .
NORTH AZIMUTH	On when pattern generator has generated a north or azimuth pulse; turned off if associated sense unit is checked by <i>BSN</i> (47) ₈ .
NON SEARCH	On whenever a nonsearch comparison alarm is generated in Output System; remain on until associated sense unit is checked by <i>BSN</i> (50) ₈ or until clear-alarm pulse is generated in Output System.
OB PARITY	On when a parity alarm is generated in transferring words from the OB fields to the Output System; remains on until associated sense unit is checked by <i>BSN</i> (51) ₈ or until cleared by clear-alarm pulse.
ILL ADR OR SEC	On when illegal address alarm is generated in Output System; remains on until associated sense unit is checked by <i>BSN</i> (52) ₈ or until cleared by clear-alarm pulse.
G/G G/A TTY PARITY	On when an output parity alarm is generated during G/A, G/G, or TTY read-out; remains on until associated sense unit is checked by <i>BSN</i> (53) ₈ or until cleared by clear-alarm pulse.
SELECT WARNING LTS	The warning lights have been selected as the output unit.
WARNING LT REG COUNTER	Indicates the status of the binary counter which indicates the warning light register being selected.
AUX WARNING LTS	The auxiliary warning light FF has been set by BO 11 or by a deselect pulse; FF conditions gates, which at TP 4 reset a counter and at TP 11 transfer a word from the IO register to the warning light registers and clear the IO register; FF is cleared at TP 11.

2.4.15 TAPE Neon Group

The neons in the TAPE group (located in the second row of neon blocks, last block on the left) indicate that the tape units have either been selected for IO operations, disabled by some mechanical fault, or are performing some programmed function. (See fig. 1–205.) The function of each neon is given in table 1–97.

2.4.16 Test and SENSE Control Switches

2.4.16.1 General

The lever switches in the test and SENSE control switch group are located in the lower right-hand section of the panel. (See fig. 1–206.) As the lever switches perform various unrelated functions, each lever switch will be discussed individually.

2.4.16.2 OPERATE COMPUTER-TEST Switch

The OPERATE COMPUTER-TEST lever switch chooses the mode of operation of the Central Computer System. During calculation, the switch is in the OPERATE COMPUTER position, thereby rendering inoperative certain controls which might disrupt calculation. The controls which are operative only when the OPERATE COMPUTER-TEST lever switch is in the TEST position are:

- a. MASTER RESET
- b. RESET FLIP-FLOPS
- c. CLEAR MEMORY
- d. MEMORY CYCLE
- e. INST STEP
- f. COMPLEMENT
- g. SINGLE PULSE
- h. SELECT TEST MEMORY
- i. NORMAL-REVERSED

When the OPERATE COMPUTER-TEST switch is in the TEST position, the TEST COMPUTER indicator is on while both the MASTER READY and the COMPUTER indicators are off.

2.4.16.3 SENSE Switches

The four SENSE lever switches provide manual determination of branches of program control. Four

Sense instruction selection codes are provided, one for each switch, which cause a branch of program control if the SENSE switch examined by the *Sense* instruction is on. The SENSE switches can thus be used manually to initiate a program branch to the address specified by the *Sense* instruction. For example, if one element of the Input System is being serviced while a tactical program is being executed, the program routine which reads the messages from that element can be bypassed until servicing is completed. A *Sense* (SENSE switch No. 1) instruction, *BSN* (21)₈, can be inserted before the routine to be bypassed, thus calling for a conditional branch to the next routine. As long as SENSE switch No. 1 is left on, that routine is bypassed. When servicing of the element is completed, SENSE switch No. 1 is turned off to allow the previously bypassed routine to be executed. The function of the SENSE 2, 3, or 4 ACTIVE switch is identical to that of the SENSE No. 1 switch, except that it is associated with a *Sense* 2, 3, or 4 instruction.

2.4.16.4 UNASSIGN-ASSIGN Switch

The UNASSIGN-ASSIGN switch selects the set of addresses for the 16 registers of test memory. With this switch in the ASSIGN position, the test memory registers are addressed as shown in table 1–98. With the switch in the UNASSIGN position, addresses 20,000₈ through 20,017₈ specify control panel addresses. (Refer to 2.6.4.)

TABLE 1–98. ASSIGNED TEST MEMORY ADDRESSES

REGISTER	ADDRESS
Switch register A	20,000 ₈
Switch register B	20,001 ₈
Plugboard 13 ₍₁₀₎	20,002 ₈ through 20,016 ₈
Live (flip-flop)	20,017 ₈

TABLE 1–97. TAPE NEONS

INDICATOR NEON	INDICATION
TAPE SELECTION	Tape units have been selected for IO operations.
TAPES NOT READY	Tape unit is disabled by some mechanical fault (door open, reel removed, etc.) which must be cleared manually
TAPES NOT PREPARED	Tapes are performing some programmed function, such as rewind, which makes it impossible to read or write on them

2.4.16.5 BRANCH NORMAL – BRANCH TO ZERO MEM Switch

When the BRANCH NORMAL – BRANCH TO ZERO MEM lever switch is in the BRANCH NORMAL position, the program branches normally on all branch instructions. When in the BRANCH TO ZERO MEM position, the program branches to the first address in core memory on all branch instructions unless there is a 1 in the LS bit of the branch instruction, in which case it branches normally. This switch is interlocked so that it functions only if the Central Computer System is in the test status.

2.4.16.6 CAMERA INDEX – SUPPRESS Switch

The CAMERA INDEX – SUPPRESS lever switch, when in the SUPPRESS position, prevents the film in the situation display camera from being advanced between exposures. After the first camera cycle, illumination of the data card and counter is prevented. Thus, successive exposures of situation displays can be made on one film frame. When this switch is not in the SUPPRESS position, the camera film is advanced after each exposure.

2.4.16.7 NORMAL – REVERSED Switch

When the NORMAL – REVERSED lever switch is in the NORMAL position, the start-memory-1 or the start-memory-2 pulse starts its memory array. When in the REVERSED position, the start-memory-1 pulse starts the memory 2 array, and vice versa.

2.4.17 Loading Preparation Controls

2.4.17.1 General

The loading preparation control pushbuttons are useful in preventing incorrect carryovers from the previous program when a new program is to be inserted into the Central Computer System. For example, after preventive maintenance and before a tactical program is loaded, these controls would be used to clear the Central Computer System control circuits and registers. The controls in this group are shown in the lower middle section of figure 1-207, and will be discussed individually.

2.4.17.2 CLEAR MEMORY Pushbutton

The CLEAR MEMORY pushbutton clears all 8,192 core memory locations and the live register of test memory to positive zero. Depressing this pushbutton is equivalent to executing the short program listed in table 1-99.

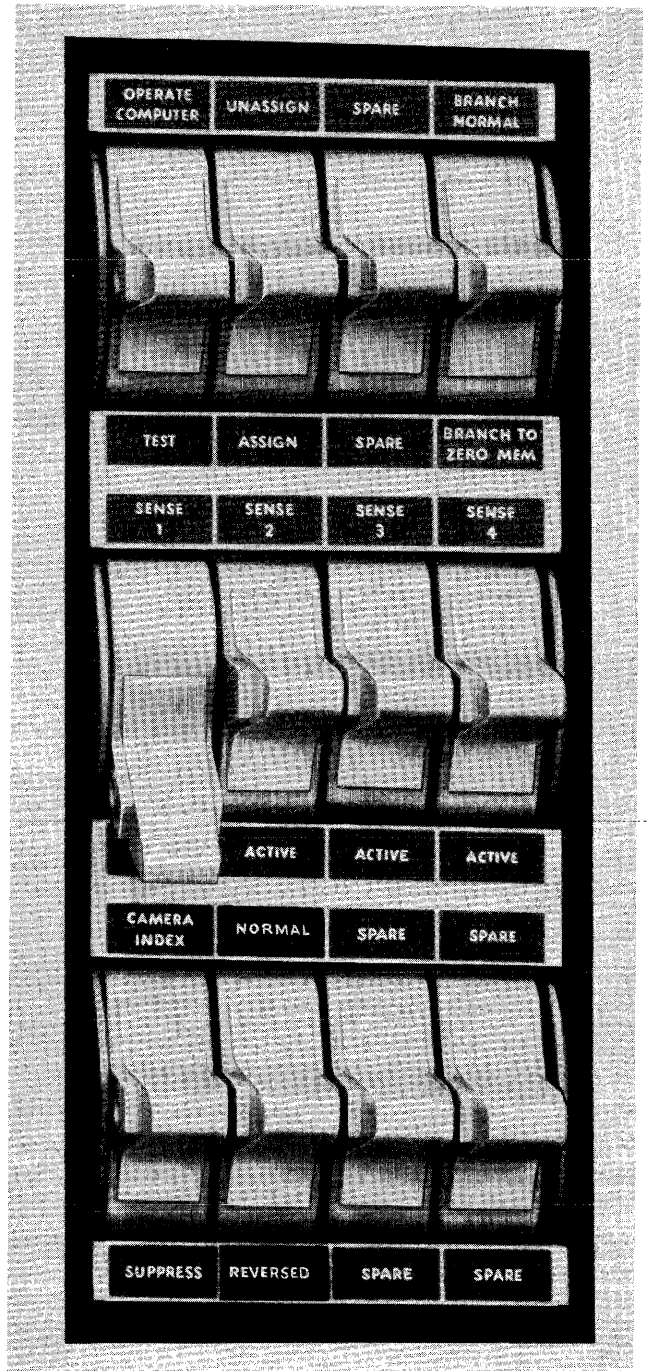


Figure 1-206. Test and SENSE Control Switches

The CLEAR MEMORY pushbutton is interlocked to prevent interference with computation and may be used only when the OPERATE COMPUTER-TEST switch is in the TEST position.

2.4.17.3 RESET FLIP-FLOPS Pushbutton

The RESET FLIP-FLOPS pushbutton clears all flip-flop registers and all control flip-flops in the Central

Computer System. This pushbutton provides identical starting conditions for all programs, eliminating the necessity of ascertaining the condition of each flip-flop before starting a new program. The RESET FLIP-FLOPS pushbutton, like the CLEAR MEMORY pushbutton, is interlocked to prevent interference with computation, and is usable only when the OPERATE COMPUTER-TEST switch is in the TEST position.

2.4.17.4 CLEAR ALARMS Pushbutton

The CLEAR ALARMS pushbutton clears the parity and overflow alarms and the INDICATION alarm neons in the Central Computer System and all alarms in the Drum System. The CLEAR ALARMS pushbutton is not

interlocked since its operation cannot interfere with computation.

2.4.17.5 RESET AUDIBLE ALARM Pushbutton

When depressed, the RESET AUDIBLE ALARM pushbutton resets all alarm flip-flops. This deconditions the audible alarm relay drivers and the audible alarm indicating neons associated with alarm flip-flops. A new alarm can again set off the audible alarm after it has been reset.

2.4.17.6 MASTER RESET Pushbutton

The MASTER RESET pushbutton combines certain functions of the CLEAR MEMORY, READY IO UNIT and RESET FLIP-FLOPS pushbuttons in one control.

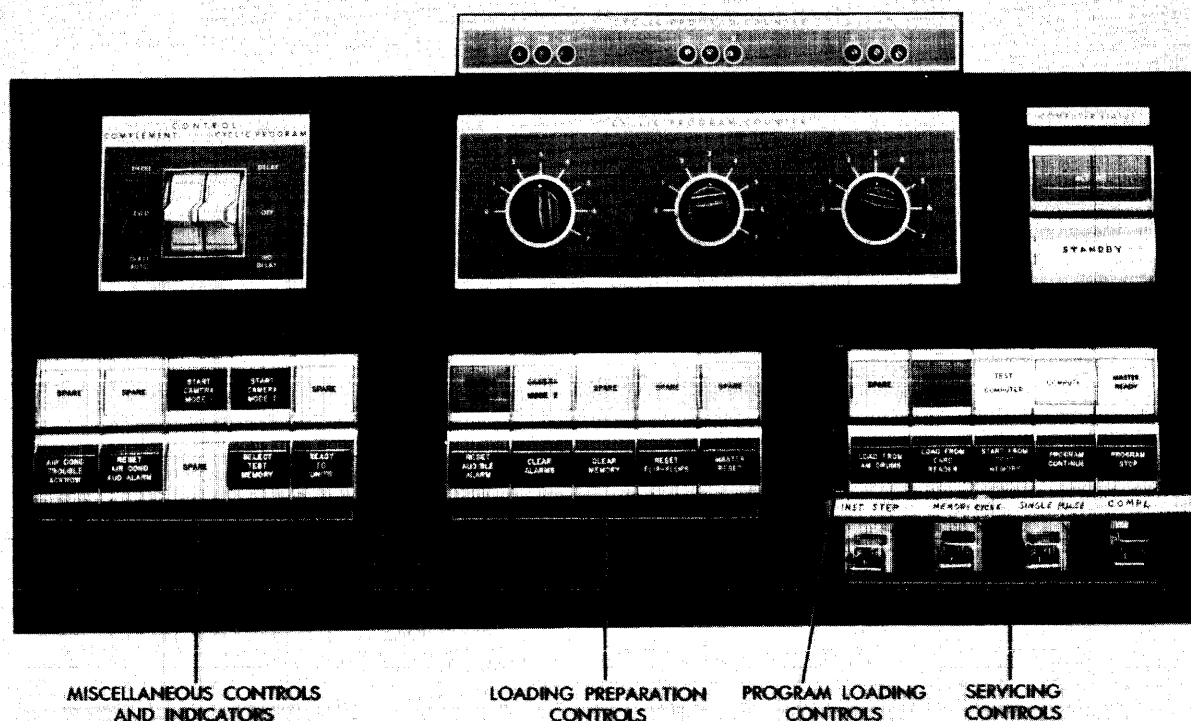


Figure 1-207. Operation Controls and Indicators

TABLE 1-99. CLEAR MEMORY PUSHBUTTON
EQUIVALENT PROGRAM

INSTRUCTION		
OPERATION	ADDRESS	FUNCTION
LDC	0000 ₈	Loads IO address counter with first address in core memory unit 1.
SEL (04) ₈	—	Selects IO register which has contents of 0.
RDS	20000	Reads 8,192 ₁₀ words containing positive zero into core memory.

The MASTER RESET pushbutton performs certain functions of the CLEAR ALARMS pushbutton but does not clear all drum alarms. When the MASTER RESET pushbutton is used to prepare for a change to a new program, the RESET FLIP-FLOPS pushbutton must be depressed as well, since the MASTER RESET pushbutton leaves the IO address counter set to 20001₈ and the time pulse distributor at TP 7. The program-loading pushbuttons are not effective unless the time pulse distributor (TPD) is first cleared to TP 0. The sequence of operations required in the preparation of a new program is then:

- a. MASTER RESET
- b. RESET FLIP-FLOPS
- c. The appropriate program loading pushbutton.

Since the MASTER RESET pushbutton can interfere with computation, it is interlocked and is operative only when the OPERATE COMPUTER-TEST switch is in the TEST position. A summary of the functions of the loading-preparation controls is given in table 1-100.

2.4.18 Program-Loading Controls

2.4.18.1 General

The program-loading controls include those controls which actually initiate the loading of a program into the Central Computer System. The program-loading method employed in AN/FSQ-7 Combat Direction Central starts with the loading of a short program, initiated manually, which controls the loading of the rest of the program. The short program-loading program may be obtained from the card reader, the auxiliary memory drums of the Drum System, or from test memory. The program-loading controls choose one of these sources. In addition, the program-loading controls provide the means of stopping a program before completion if a program changeover or servicing is required and of

restarting a program from the point of interruption. These controls, shown in the lower righthand section of figure 1-207, will be discussed individually.

2.4.18.2 PROGRAM STOP Pushbutton

The PROGRAM STOP pushbutton is equivalent in function to the *Program Stop (HLT)* instruction; the Central Computer System is brought to a halt upon completion of the preceding instruction. If an IO operation is in progress when the PROGRAM STOP pushbutton is depressed (or the *HLT* instruction is decoded), that operation is completed before the halt. Since the PROGRAM STOP pushbutton has no effect when the Central Computer System is not calculating, it can be said to be interlocked with the COMPUTE indicator.

2.4.18.3 PROGRAM CONTINUE Pushbutton

The PROGRAM CONTINUE pushbutton restarts the Central Computer System from the point at which its program was stopped by an *HLT* instruction, the PROGRAM STOP pushbutton, or by the STOP-BRANCH switch. This switch determines whether the Central Computer System will halt or branch when a program alarm is generated.

The PROGRAM CONTINUE pushbutton does not clear the parity and overflow alarms, but it does cause the program to continue from the point of interruption. This pushbutton cannot be used to start a new program unless the new program is in memory, starting at the location following the *HLT* instruction that ended the previous program.

2.4.18.4 LOAD FROM CARD READER Pushbutton

The LOAD FROM CARD READER pushbutton is one of the basic program-loading controls. Depression of this pushbutton gives a *control clear* command, then

TABLE 1-100. LOADING PREPARATION CONTROLS

PUSHBUTTON	INTERLOCKED	FUNCTION
RESET AUDIBLE ALARM		Turns off audible alarm
CLEAR ALARMS		Clears parity, overflow, and Drum System alarms
CLEAR MEMORY	TEST	Clears all core memory registers and live register to positive zero
RESET FLIP-FLOPS	TEST	Clears all flip-flop controls and registers in Central Computer System
MASTER RESET	TEST	Combines functions of CLEAR MEMORY, RESET FLIP-FLOPS, and READY INPUT OUTPUT UNITS pushbuttons

causes operations equivalent to the execution of the program listed in table 1–101.

The 24 instructions loaded by the LOAD FROM CARD READER pushbutton control the loading of an entire program into the Central Computer System, whether from the card reader, the auxiliary memory drums, or the tape units. Initially, almost all programs are introduced through the card reader. Some regularly used programs are stored on the auxiliary memory drums when not in use. They can be reloaded into the Central Computer System by a loading program contained on a punched card. The LOAD FROM CARD READER pushbutton would load this loading program which, in turn, would load the program stored in the Drum System. The LOAD FROM CARD READER pushbutton is interlocked to prevent interference with computation and is operative only while the MASTER READY indicator is on. If a program in progress must be replaced by another program, it can be halted by the PROGRAM STOP pushbutton, allowing the new program to be loaded.

A short program may be prepared on a punched card to effect the same operations as the LOAD FROM CARD READER pushbutton. Such a card might be used when loading a series of library routines to make up a full program. This card would initiate the reading of the loading program contained on the next card (part of the next routine) which, in turn, would load the next library routine.

2.4.18.5 LOAD FROM AM DRUMS Pushbutton

The LOAD FROM AM DRUMS pushbutton performs the same function as the LOAD FROM CARD READER pushbutton, with the exception that the 24 instructions are taken from the first field of the auxiliary memory drums rather than from a punched card.

In general, the LOAD FROM AM DRUMS pushbutton causes the loading of the master program of AN/FSQ-7 Combat Direction Central which is stored on the auxiliary drum group during maintenance operations. Other programs stored on the auxiliary memory drums can also be loaded into the Central Computer System under control of a loading program but initiated by the LOAD FROM CARD READER pushbutton. The LOAD FROM AM DRUMS pushbutton is not operative while the Central Computer System is following a program. A program in progress must be halted before the LOAD FROM AM DRUMS pushbutton can be used.

2.4.18.6 START FROM TEST MEMORY Pushbutton

The START FROM TEST MEMORY pushbutton is similar in function to the LOAD FROM CARD READER pushbutton. However, the START FROM TEST MEMORY pushbutton does not initiate any writing in core memory. Instead, the program counter directs the transfer of the first word in test memory to the operation register, thus taking the contents of the first word in test memory directly as the first program step which the Central Computer System is to execute. The short program contained in test memory may direct the loading of a program from some other source. Generally, however, test memory is used for short test programs which can be contained in 16 binary words. (Refer to 2.6.)

The START FROM TEST MEMORY pushbutton is inoperative while a program is being executed. Computation can be halted by use of the PROGRAM STOP pushbutton; the START FROM TEST MEMORY pushbutton can then be used. The functions of the program-loading controls discussed in the preceding text are summarized in table 1–102.

TABLE 1–101. LOAD FROM CARD READER EQUIVALENT PROGRAM

INSTRUCTION		
OPERATION	ADDRESS	COMMENT
LDC	0000 ₈	Loads IO address counter with memory address 0000
SEL (01) ₈	—	Selects card reader
RDS	30 ₈	Reads 24 ₁₀ words (contents of one card) into core memory; no LDC instruction necessary; IO address counter is cleared by <i>control clear</i> command; 24 words written in addresses 0000 ₈ through 0027 ₈
SEL (01) ₈	—	Selects card reader to delay computer until contents of card are written into core memory
BPX	0000 ₀	Unconditional branch to memory address 0000 ₈ , first word read from card

2.4.19 Program-Loading Control and Test Indicators

The four indicators located near the program-loading controls group are described in table 1-103. (See fig. 1-207.)

The MASTER READY indicator is on when the Central Computer System is operative but not calculating. It is extinguished when the OPERATE COMPUTER TEST switch is in the TEST position or when a program is in progress. When a program is being executed, the COMPUTE indicator is on instead of the MASTER READY indicator.

Several controls which would disrupt calculation if operated while a program is being executed are inoperative while the COMPUTE indicator is on. These controls are:

- a. LOAD FROM CARD READER pushbutton
- b. LOAD FROM AM DRUMS pushbutton
- c. START FROM TEST MEMORY pushbutton.

2.4.20 Servicing Controls**2.4.20.1 General**

The pushbutton controls included within the group called servicing controls (located in the lower right-hand portion of the panel) are used during testing of the Central Computer System. (See fig. 1-207.) The

servicing controls allow a division of the instruction into its component parts. These servicing controls are interlocked in order to perform their functions only when the Central Computer System is stopped and in the test status. The function of each pushbutton control will be discussed individually.

2.4.20.2 INST STEP Pushbutton

The INST STEP pushbutton, when depressed, sets the instruction step sync FF, the continue FF, and the TPD control set sync FF, causing the computer to go through one instruction step. The switch is interlocked in order to perform its functions only when the Central Computer System is stopped and in the test status.

2.4.20.3 MEMORY CYCLE Pushbutton

The MEMORY CYCLE pushbutton, when depressed, sets the memory cycle FF, the continue FF, and the TPD control set sync FF, causing the execution of one memory cycle. This switch is interlocked in order to perform its functions only when the Central Computer System is stopped and in test status.

2.4.20.4 SINGLE PULSE Pushbutton

The SINGLE PULSE pushbutton, when depressed, advances the program one time pulse. This switch is interlocked in order to perform its function only when the Central Computer System is stopped and in the test status.

TABLE 1-102. PROGRAM-LOADING CONTROLS

CONTROL PUSHBUTTON	INTERLOCKED	FUNCTION
PROGRAM STOP	COMPUTE indicator	Stops program in progress at completion of instruction or at end of IO break
PROGRAM CONTINUE		Restarts program from point of interruption
LOAD FROM CARD READER	MASTER READY indicator	Gives <i>control clear</i> command, reads 24 words from card reader into core memory, then branches to location 0 of core memory 1
LOAD FROM AM DRUMS	MASTER READY indicator	Identical to LOAD FROM CARD READER, except that 24 words are read from auxiliary memory drum
START FROM TEST MEMORY	MASTER READY indicator	Initiates execution of 16-word program in test memory without transfer of those words to core memory

TABLE 1-103. PROGRAM-LOADING CONTROL AND TEST INDICATORS

INDICATOR LIGHT	INDICATION
OPERATE COMPUTER	The Central Computer System in the operate status
TEST COMPUTER	OPERATE COMPUTER-TEST switch in the TEST position
COMPUTE	Program in progress
MASTER READY	Central Computer System operative but idle

2.4.20.5 COMPLEMENT Pushbutton

The COMPLEMENT pushbutton, when depressed, complements all FF's in the Central Computer System either two or three times, as specified by the COMPLEMENT CONTROL switch. This switch is interlocked to perform its function only when the Central Computer System is stopped and in the test status.

2.4.21 Miscellaneous Controls and Indicators

2.4.21.1 General

The controls and indicators included in the miscellaneous controls and indicators group are located in the lower left-hand portion of the panel. (See fig. 1-207.) As the functions of the controls cover operations of the SD camera, air conditioning, IO units, and test memory that are not directly related to each other, the function of each pushbutton control will be discussed individually. The two situation display (SD) camera indicator lights located above the loading preparation controls are associated with the camera control pushbuttons and will be discussed in relationship to them.

2.4.21.2 AIR COND TROUBLE ACKNOW Pushbutton

The AIR COND TROUBLE ACKNOW pushbutton switch prevents the automatic shutdown of the system when trouble occurs in the air conditioning equipment. This allows the operator to complete any important procedures before turning off the computer.

2.4.21.3 RESET AIR COND AUD ALARM Pushbutton

The RESET AIR COND AUD ALARM pushbutton, when operated, resets the audible alarm for air conditioning trouble if the alarm has been acknowledged in the air conditioning control area.

2.4.21.4 SELECT TEST MEMORY Pushbutton

The SELECT TEST MEMORY pushbutton, when operated, resets FF's and sets the program counter to 20000₈, selecting test memory as the source of the next instruction. This switch is interlocked in order to perform its functions only when the Central Computer System is in test status. Pressing the PROGRAM CONTINUE pushbutton after pressing this pushbutton is equivalent to pressing the START FROM TEST MEMORY pushbutton.

2.4.21.5 READY IO UNITS Pushbutton

The READY IO UNITS pushbutton is a control which puts the IO units associated with the Central Computer System in the ready condition. These units include the tape units, IBM type 713 card reader, IBM type 723 card punch, and IBM type 718 line printer. Although each unit can be readied by a control located

at the unit, the READY IO UNITS pushbutton is provided as a convenient control to eliminate the necessity of walking over to each unit to ready it. The pushbutton cannot interfere with computation and therefore is not interlocked.

2.4.21.6 START CAMERA MODE 1 and 2 Pushbuttons

The START CAMERA MODE 1 and 2 pushbuttons are used to manually select SD camera mode 1 or 2. The display at the SD camera console is switched to the mode 1 or the mode 2 situation display established by the display console category switches. The CAMERA MODE 1 or CAMERA MODE 2 indicator light is illuminated when the associated pushbutton is depressed.

2.4.22 COMPLEMENT Lever Switch

The COMPLEMENT three-position lever switch (located in the lower left-hand portion of the panel, above the miscellaneous controls) allows testing of certain flip-flops in the Central Computer System not readily tested by a reliability program. (See fig. 1-207.)

When the COMPLEMENT lever switch is in the TWO position, depressing the COMPLEMENT pushbutton switch complements certain control flip-flops twice at a 2-mc rate. With the COMPLEMENT lever switch in the THREE position, depressing the COMPLEMENT pushbutton complements the same flip-flops three times at a 2-mc rate. When the COMPLEMENT lever switch is in the THREE AUTO position, the same control flip-flops are continually complemented three times at a 2-mc rate, with sufficient pause between each sequence to observe appropriate neons.

In practice, testing of flip-flops using these two switches precedes the running of a reliability program. Since the program makes use of the control flip-flops, erroneous operation of the flip-flops could lead to incorrect indication of errors in other parts of the Central Computer System. The flip-flops to be tested are cleared by the use of the RESET FLIP-FLOPS pushbutton (loading preparation controls group). When the COMPLEMENT lever switch is in the TWO position and the COMPLEMENT pushbutton is depressed, there should be no resultant change in the state of the flip-flops. When the COMPLEMENT lever switch is in the THREE position and the COMPLEMENT pushbutton is depressed, all flip-flops should reverse their states. The COMPLEMENT lever switch is interlocked and is operative only when the Central Computer System is in the test status.

2.4.23 COMPUTER STATUS Indicators

The COMPUTER STATUS indicators (located in the lower right-hand portion of the panel, above the program-loading controls) indicate whether the computer associated with the duplex maintenance console

on which the light appears is active or in standby, as controlled by the duplex switch. When the ACTIVE indicator is on, this light indicates that the computer associated with the duplex maintenance console on which the light appears is active, as controlled by the duplex switch. When the STANDBY indicator light is on, the computer is in the standby status.

2.4.24 Cyclic Program Controls

Cyclic program controls (located on the lower portion of the panel, above the loading preparation controls) provide the means of continuously repeating steps of a maintenance program in which a malfunction has occurred in order to locate more accurately the cause of the malfunction. (See fig. 1-207.) When a portion of a program is cycled, a specified number of timing pulses (TP) and instruction pulses (IP) are supplied to the control element by the time pulse distributor (TPD) under indirect control of the cyclic program controls. The number of pulses supplied is sufficient to generate the commands which execute those instructions in the cycled loop. When one cycle is completed, the pulses are interrupted momentarily while a counter is automatically reset and the cycle is repeated. The pause between cycles may be lengthened by inserting a relay in the circuit which must be activated to restart the cycle.

The cyclic program controls include:

- a. CYCLIC PROGRAM COUNTER rotary switches
- b. CYCLIC PROGRAM DELAY-OFF-NO DELAY lever switch
- c. CYCLIC PROGRAM COUNTER neons.

The CYCLIC PROGRAM COUNTER switches, three 8-position switches, set the number of instruction pulses within the repeating program cycle. They insert a nine-bit binary number whose value may be between decimal zero and decimal 511, inclusive. (The counter can be set to zero. However, since two pulses are required before the loop restarts, only a loop of from 2 to 511 pulses in length can be selected.) Since each memory cycle includes 12 timing or instruction pulses, a cyclic

program loop up to 42 complete memory cycles in length can be employed. The counter counts down from the number inserted by the CYCLIC PROGRAM COUNTER switches to zero during each cycle, is then reset to that number, and counts down again, generating a *control clear* command at the end of each loop.

The CYCLIC PROGRAM DELAY-OFF-NO DELAY switch chooses the mode of operation of the cycle program equipment. In the OFF position, the equipment is inoperative. In the NO DELAY position, the pause between cycles is only as long as necessary to reset the cyclic program counter to the number from which it counts down. In the DELAY position, a relay must be energized before the counter resets. The mechanical delay introduced by the relay is sufficient to allow photographing the indicator neons between cycles. The program sequence being cycled is executed at normal speeds in both the DELAY and NO DELAY modes. These two switch positions merely determine the interval between cycles.

The CYCLIC PROGRAM COUNTER neons indicate the contents of the cycle program counter. During a cycle, they change too rapidly to be read. In operation, if a malfunction is detected while a particular sequence of a maintenance program is being run, that sequence can be cycled continuously while observations are made on suspected components to locate the trouble. The program is stopped by use of the PROGRAM STOP pushbutton. The number of instruction pulses needed to execute the desired sequence of instructions is inserted with the CYCLIC PROGRAM COUNTER switches, and the DELAY or NO DELAY mode of operation is chosen. An unconditional branch instruction, specifying the address of the first instruction of the sequence to be cycled, is set up on test memory switch register A. The cyclic program is then started by depressing the START FROM TEST MEMORY pushbutton. Each cycle begins by clearing control and computing flip-flops, then starting from test memory. A summary of the functions of the cyclic program controls is given in table 1-104.

TABLE 1-104. CYCLIC PROGRAM CONTROLS

CONTROL	POSITION	FUNCTION
CYCLIC PROGRAM switch	OFF	Cyclic program equipment rendered inoperative.
	NO DELAY	Allows loop to recycle with pause between repeats long enough to reset counter.
	DELAY	Pauses between repeats by time necessary to activate relay.
CYCLIC PROGRAM COUNTER switches		Sets nine-bit binary number in octonary form, specifying number of timing pulses within cyclic loop of instructions.

2.5 REGISTER NEONS PANEL

The register neons panel houses a bank of indicator neons which show the contents of all flip-flop registers in the Central Computer System. (See fig. 1-208.) Every neon indicator indicates the status of a particular bit position. An illuminated neon indicates that the bit position contains a 1. If the neon is not illuminated, the bit position contains a 0. Each row of neons represents the contents of a specific register or counter. In order to be able to read the contents of these registers or counters with comparative ease, the neons for each register are grouped in series of threes so the octonary equivalent can be read directly. The register and counter neons are used primarily during testing since, during computation, their indications change too rapidly to be perceived visually. One example of the use to which the register neons are put is in clearing a hangup. By reading the contents of the PROGRAM COUNTER register (shown in the left half, fourth row from the bottom of figure 1-208), the point in the program at which the hangup occurred can be determined. The PRO-

GRAM COUNTER register in this case would contain the address of the instruction after the one which caused the hangup.

A list of the register neons in the order in which they appear on the register neon panel is given in table 1-105.

2.6 TEST MEMORY

2.6.1 Capacity

Test memory, consisting of 2 toggle switch registers, 16 plugboard registers, and one flip-flop register called the test or live register, provides storage for a relatively small number of words. Only 16 words in test memory may be addressed specifically since only 16 addresses are available for the selection of test memory registers.

2.6.2 Register Characteristics

The contents of toggle switch registers A and B are set by the manual positioning of 64 toggle switches mounted below the register neons. (See fig. 1-209.) The number of switches required for each toggle switch

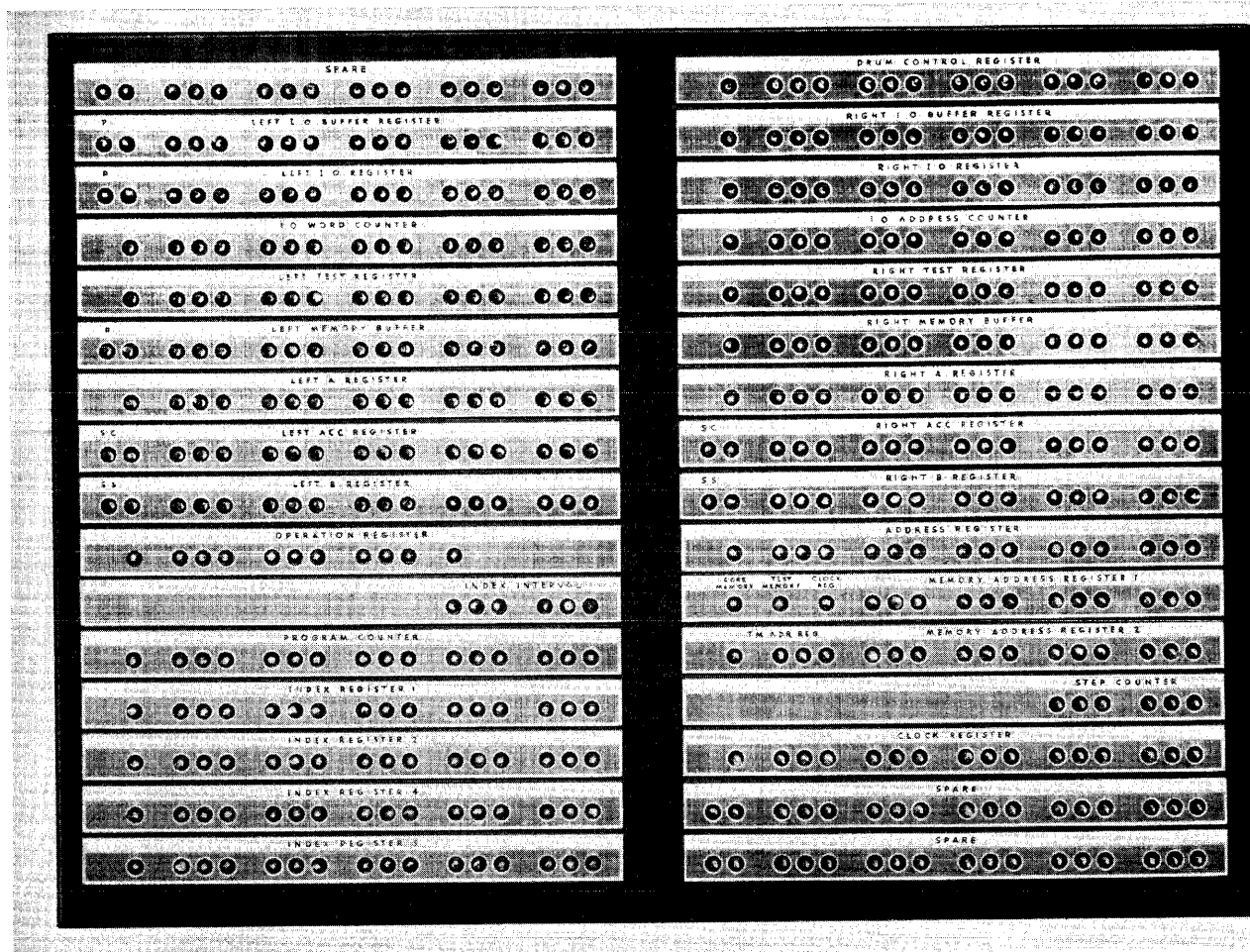


Figure 1-208. Register Neons

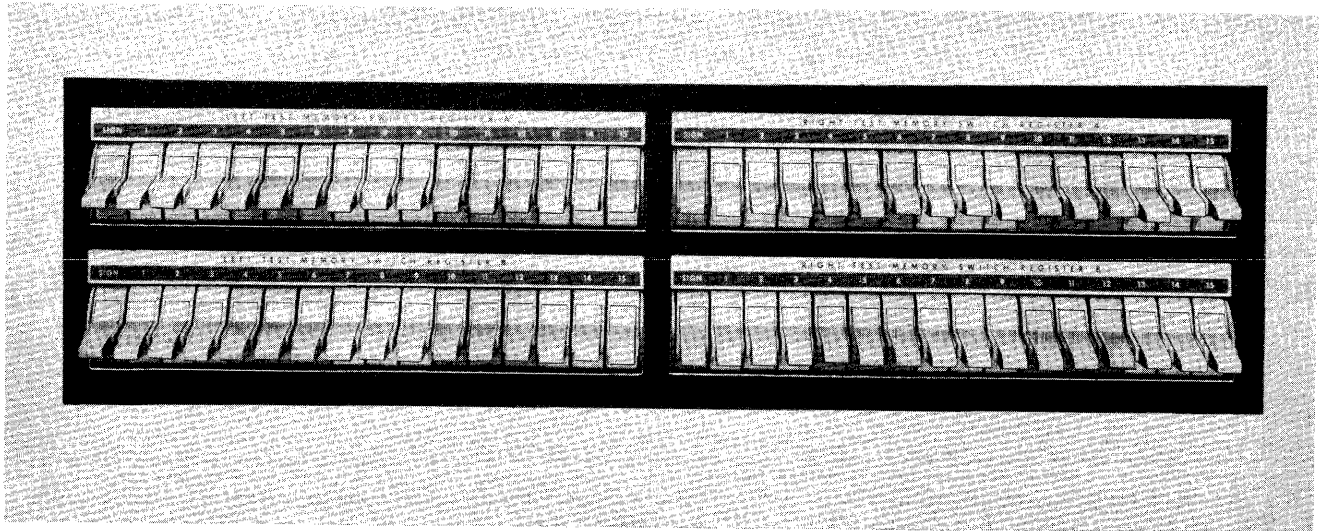


Figure 1-209. Test Memory Toggle Switch Registers

register imposes a practical limit on the number of switch registers. For example, if 16 switch registers were provided in test memory, 512 switches would require

setting each time a new set of words was to be stored in test memory. Accordingly, a plugboard is used to determine the contents of up to 16 registers in test memory.

TABLE 1-105. REGISTER NEONS

LEFT HALF		RIGHT HALF			
SPARE		DRUM CONTROL REGISTER			
* P	LEFT IO BUFFER REGISTER	RIGHT IO BUFFER REGISTER			
* P	LEFT IO REGISTER	RIGHT IO REGISTER			
	IO WORD COUNTER	IO ADDRESS COUNTER			
	LEFT MEMORY BUFFER	RIGHT TEST REGISTER			
* P	LEFT MEMORY BUFFER	RIGHT A REGISTER			
	LEFT A REGISTER	RIGHT A BUFFER			
**SC	LEFT ACC REGISTER	**SC	RIGHT ACC REGISTER		
***SS	LEFT B REGISTER	***SS	RIGHT B REGISTER		
	OPERATION REGISTER		ADDRESS REGISTER		
	INDEX INTERVAL	CORE	TEST	CLOCK	MEMORY ADDRESS REGISTER 1
		MEMORY	MEMORY	REG	
	PROGRAM COUNTER	TM ADR REG			MEMORY ADDRESS REGISTER 2
	INDEX REGISTER 1				STEP COUNTER
	INDEX REGISTER 2	CLOCK REGISTER			
	INDEX REGISTER 4	SPARE			
	INDEX REGISTER 5	SPARE			

* Parity

** Sign control

*** Sign storage

The contents of each plugboard register are set by plugging connections from each exit hub for a bit position which should contain a 1 to a common 1 line. (See fig. 1-210.) Once set up, the plugboard is inserted in the receptacle provided for it. This receptacle is located on the tape element and manual data input indicator panel. When it is necessary to change the contents of test memory, the plugboard can be removed and another plugboard inserted in its place. The removed plugboard can be stored for future use without disturbing

the preset connections on it. Thus, with a number of plugboards, the contents of the plugboard registers of test memory can be changed rapidly and conveniently.

The test register is referred to as register L since it is a live register; it is the only register of test memory which may be written as well as read. The test register is used primarily during maintenance operations.

2.6.3 Test Memory Reading

Words are read from test memory (or written into

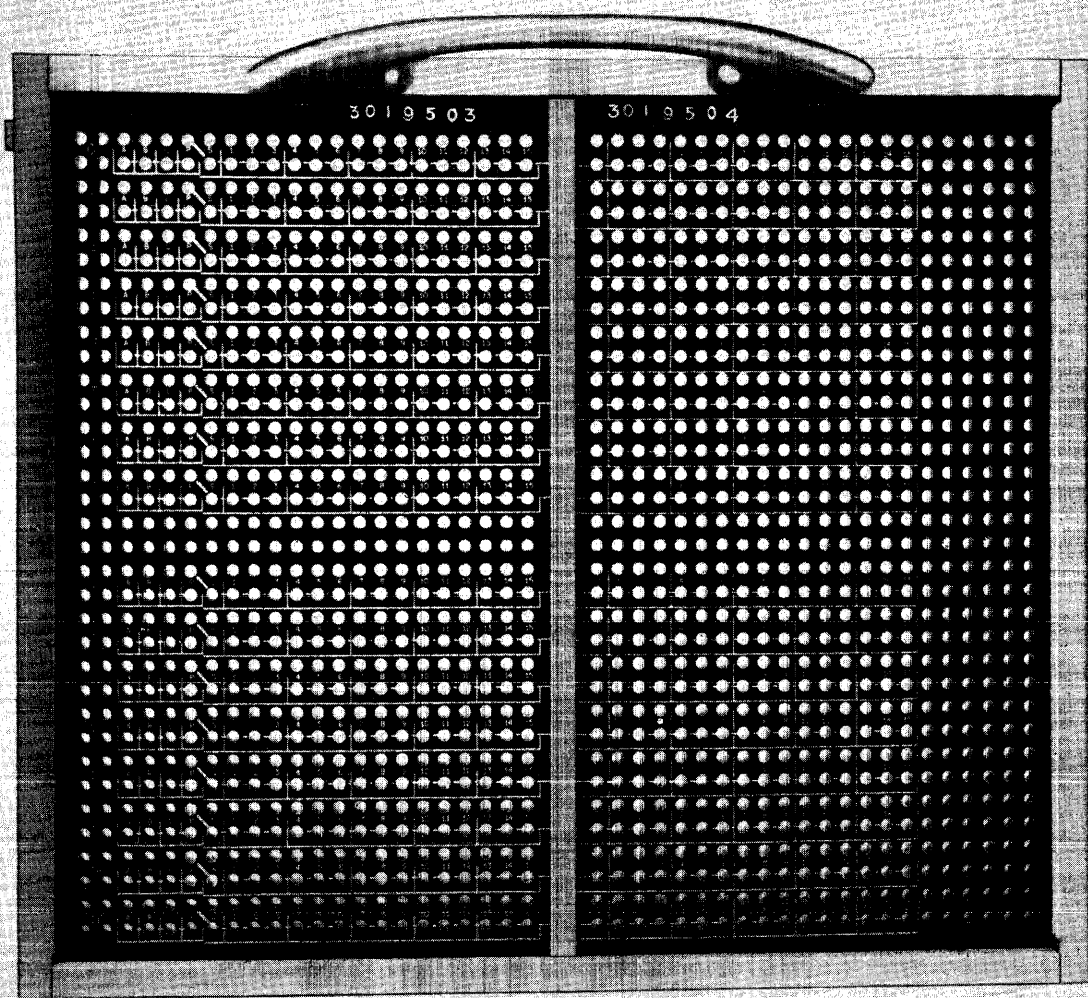


Figure 1-210. Test Memory Control Panel

the test register) via the memory buffer register. An address is supplied to the test memory address register to select a word for reading. The word in the selected register is transferred to the memory buffer register and then treated like a word read from core memory. There is one exception: no parity check is performed on words read from test memory. Further, no rewrite is necessary, since readout from test memory is not destructive.

2.6.4 Test Memory Selection and Addressing

The test memory address register is a 4-bit flip-flop register. Its contents can identify 16 addresses (2^4). These addresses are generally given in octonary notation and range from 0 through 17_8 . The assignment of a specific register to a specific address is accomplished by connections on the test memory plugboard and by setting the test memory UNASSIGN-ASSIGN switch on the duplex maintenance console. (Refer to 2.4.16.4.)

Four address assignment hubs are associated with each address entrance hub on the test memory plugboard. (See fig. 1-210.) These four hubs are designated:

- a. P — plugboard register
- b. A — switch register A
- c. B — switch register B
- d. L — test register.

Each P exit hub corresponds to the plugboard register aligned with it. By plugging a P exit hub on a particular line, the plugboard register is assigned the address of that line. For example, if the line labelled 16 on figure 1-210 is plugged P, test memory address 16 selects plugboard register 16 for reading.

The A, B, and L exit hubs are common connections. All A exit hubs make connection to the same register selection line. Thus, two addresses can be assigned for selection of one of these registers by plugging two address entrance hubs to A, B, or L. For example, lines 4 and 17 might be plugged A, thus causing the reading of switch register A if either address 4 or 17 is given.

Plugboard address assignments establish the correspondence of address to register in all cases when the test memory UNASSIGN-ASSIGN switch is in the UNASSIGN position. However, when this switch is in the ASSIGN position, the plugged address assignments for

the A, B, and L registers are overridden. These registers are assigned the following addresses:

- a. Switch register A — address 0
- b. Switch register B — address 1
- c. Test register L — address 17_8 .

The plugged address assignments for the plugboard registers apply regardless of the position of the test memory UNASSIGN-ASSIGN switch.

An interesting possibility exists in the use of plugged address assignments. Two registers may be assigned the same address. If this is done, the contents of both registers are selected for reading by the one address. The resulting word transferred to the memory buffer register is the logical sum (rather than the arithmetic sum) of the two words read. There will be a 1 in any bit position where either or both words have a 1, and a 0 where both words contain a 0. Using 5-bit words to illustrate this, if the two words read are 0.0101 and 0.0011, the resulting word in the memory buffer register is 0.0111.

When the Central Computer System is halted, test memory may be selected manually, using the SELECT TEST MEMORY pushbutton. (Refer to 2.4.21.4.) This facility is particularly useful in conjunction with the INST STEP, MEMORY CYCLE, and SINGLE PULSE pushbuttons, since a program can be entered by setting an unconditional branch instruction on toggle switch register A, depressing the SELECT TEST MEMORY pushbutton, and then using whichever switch is desired to advance through the program one step at a time. The START FROM TEST MEMORY pushbutton (described in 2.4.18.6) can be used to begin execution of a program contained in test memory or to enter another program at a selected point; in the latter case, test memory toggle switch register A is set to contain a branch to the desired point of program entry, the UNASSIGN-ASSIGN switch is placed in the ASSIGN position, and the START FROM TEST MEMORY pushbutton is depressed.

2.7 TAPE INDICATORS AND CONTROLS

The TAPES section (located above the MANUAL INPUTS section on the tape element and manual data input indicator panel) houses indicator lights, neons, and pushbutton switches which control various test op-

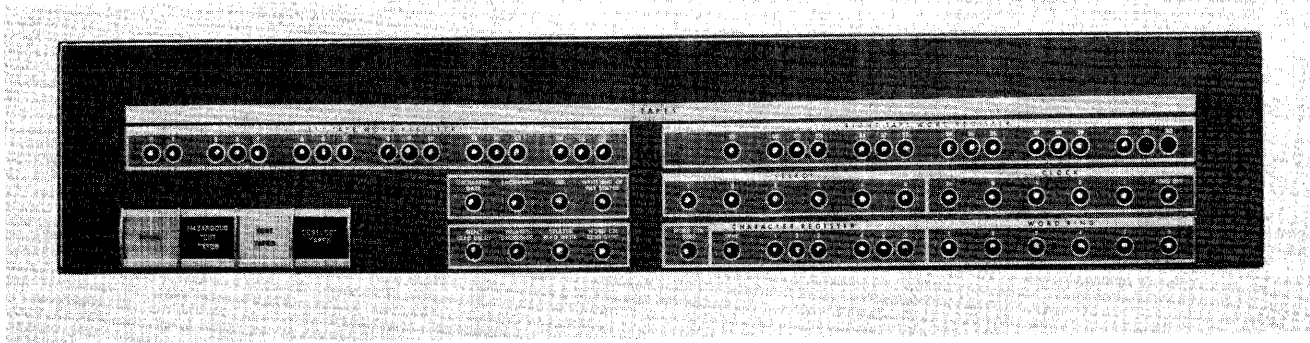


Figure 1-211. Tape Element Controls and Indicators

erations and monitor magnetic tape operations. (See fig. 1-211.) The left and right tape word register neons represent the status of each bit in the left and right half-word being written on tapes or read from tapes. An illuminated neon indicates that the bit position contains a 1. An extinguished neon indicates that the bit position contains a 0. The SELECT neons indicate which of the six tape units has been selected for tape oper-

ations. An example of the use of a combination indicator light and switch is the HAZARDOUS TEST TAPE indicator light which goes on whenever the tapes are placed in the test status by operating the HAZARDOUS TAPE TEST switch. The functions of the tape neon indicators, indicator lights, and controls pertaining to magnetic tape operations are listed in tables 1-106 and 1-107.

TABLE 1-106. TAPE NEONS

INDICATOR NEON	FUNCTION
LEFT TAPE WORD REGISTER and RIGHT TAPE WORD REGISTER	These 32 neons represent the status of each bit in the left and right half-words being written on tapes or read from tapes.
OSCILLATOR GATE	Turned on by the start-write pulse in write status or by the gated sync pulse in read status; when on, the 1190-kc oscillator output is gated to the frequency divider FF used to step the clock.
BACKWARD	Turned on by a read-write-backspace pulse operation; causes the tape-reversing mechanism to be operated; tapes move backward until the end-of-record pulse.
SELECT 1 through 6	These six neons indicate which of the six tape busses has been selected for tape operations; any and all tape drives connected to the selected tape bus are selected; neon is turned on by a PT 5 pulse and remains on until the next OT 5-deselect pulse.
CLOCK 1, 2, 4, 8, 16	These five neons represent the status of each bit in a scale-of-32 counter which is stepped at a 595-kc rate.
FREQ DIV	This neon is complemented on and off at 1,190-kc rate from a crystal oscillator; it gates every other oscillator pulse to step the clock; consequently, the clock is stepped at a 595-kc rate.
SELECT READ DELAY	Turned on by the backspace-go pulse if the tape unit is in write status; when the backspace instruction is given, causes the read-write-go pulse to be delayed 6 microseconds instead of 4 microseconds so that the read heads do not read erroneous information that may have been written in changing from write to read status; neon is turned off by the deselect pulse.

TABLE 1-106. TAPE NEONS (cont'd)

INDICATOR NEON	INDICATION
DELAYED BACKSPACE	Turned on 25 milliseconds after the backspace (RD) or backspace (WR) pulse is generated; while on, it gates the beginning-of-record pulse to stop the tapes and shift from forward to reverse; this same pulse turns the neon off.
DELAYED READ WRITE	Turned off by the read-write-go pulse; if the tape is at the load point, the neon is turned on 24 milliseconds after the read-write-go pulse comes in; if the tape is not at the load point, neon is turned on 4 milliseconds after the read-write-go pulse.
WORD CTR ZERO SYNC	Turned on at TP 11 when the IO word counter =0 for write status; neon is turned off by the disconnect or the deselect pulse.
WORD CTR ZERO	Neon is turned on by the word ring =5 pulse when WORD CTR ZERO SYNC is on; word ring =6 pulse is then gated to reset and stop the clock and turn this neon off.
SYNC	Once each 53.7 microseconds when the tape sync bit is read or written; neon is turned off at clock =13 in reading or at clock =29 in writing.
CHARACTER REGISTER 1 through 6	These six neons represent the status of each bit in the character read from tapes; characters are transferred from this register to the word register; this register is not used in writing.
WORD RING 1 through 6	These six neons represent the status of each FF in a 6-FF ring; ring is stepped once each 53.7 microseconds by the output of the clock in write status and by the check =13 pulse (after receiving the sync pulse) in read status; ring divides the 33-bit word into six characters for writing and assembles the characters in reading.

TABLE 1-107. TAPE INDICATOR LIGHTS AND CONTROLS

CONTROL PUSHBUTTON	INDICATOR LIGHT	FUNCTION
	TEST TAPES	Goes on whenever the tapes are being tested; if the OPERATE COMPUTER-TEST switch is in the OPERATE position, the tapes are removed from test status.
HAZARDOUS TEST TAPE	HAZARDOUS TEST TAPE	Indicator light goes on whenever the tapes are placed in test status by operating the HAZARDOUS TEST TAPE pushbutton switch; in this condition, the tapes cannot be removed from test status by the OPERATE COMPUTER-TEST switch.
DESELECT TAPES		Generates a pulse which causes certain FF's in the tape adapter to be cleared.

2.8 MAIN DRUM GROUP INDICATOR PANEL**2.8.1 General**

The main drum group indicator panel houses indi-

cator neons, indicator lights, pushbutton switches, a rotary switch, and toggle switches which indicate various operations performed by the Drum System. (See fig. 1-212.)

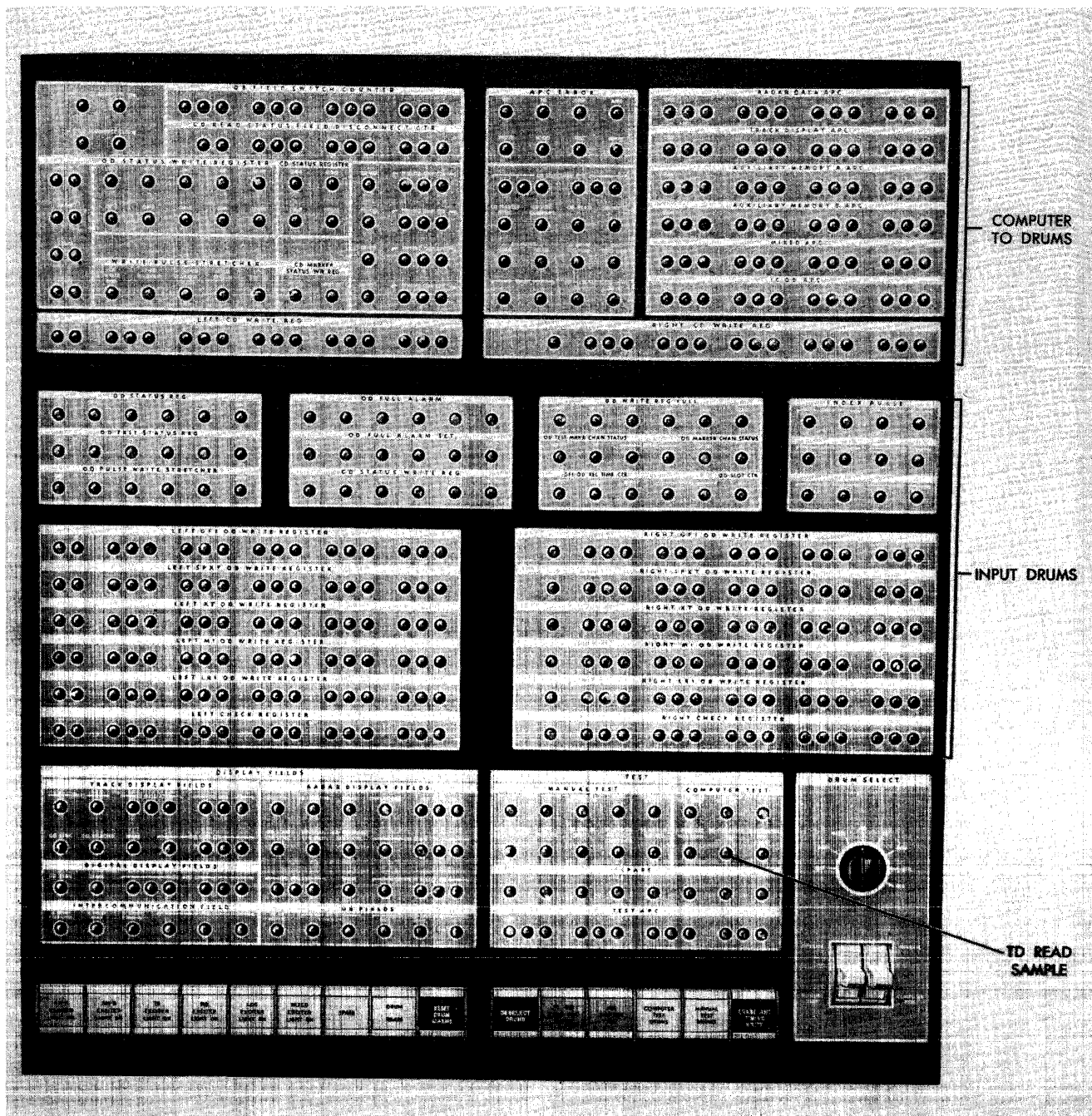


Figure 1-212. Main Drum Group Controls and Indicators

In case of a Drum System malfunction, pushbuttons and toggle switches are provided on this panel to aid maintenance personnel when performing semi-automatic (manual) tests or computer tests of the Drum System. Indications of Drum System test operations also appear on the main drum group indicator panel. When the indicators of a selected field are illuminated, a 1 bit is signified; lack of illumination signifies a 0 bit. The func-

tion of each neon, indicator light, pushbutton switch, and toggle switch is discussed in the following text.

2.8.2 COMPUTER TO DRUMS Neons Group

An explanation of the neons in the COMPUTER TO DRUMS neons group and their associated indications during a Central Computer System-Drum System (CD) information transfer is given in table 1-108.

TABLE 1-108. COMPUTER TO DRUMS NEONS

INDICATOR NEON	INDICATION
OB - CD FIELD COUNTER	These two neons represent status of bits in a 2-stage binary counter which performs the OB - CD field switching operation; for example, both neons on indicate field No. 3 is being used.
OB DISCONNECT CTR	These two neons represent status of each bit of a 2-stage binary counter which counts the number of OB fields that have been written upon.
OB FIELD SWITCH COUNTER	These first four neons represent status of the bits in a 4-stage binary counter used to suppress reading operations for the first 120 microseconds (12 registers) of each OB drum revolution to allow for switching fields during OD operations on the OB drum; the last four neons represent status of bits in a 4-stage binary counter used to suppress writing operations for the first 120 microseconds (12 registers) of each OB drum revolution to allow for switching fields during CD operations on the OB drum.
CD READ STATUS FIELD DISCONNECT CTR	These 11 neons represent status of each bit of an 11-stage binary counter which counts CD 4 pulses when certain fields on either the MIXD or LOG drum are selected for reading.
STEP DISCONNECT CTR	Turned on by the select-CD-3 pulse; remains on for one drum revolution.
LRI STATUS SLOT IND	Turned on by CD-IX pulse and complemented by LOG CD 3 pulses to insure that the step disconnect counter is not stepped until the beginning of an LRI slot.
MARKER STATUS SLOT IND	Turned on by XTL CD marker or SP-XTL CD marker pulse and turned off by the MIXD CD 3 pulse; this insures that the step disconnect counter is not stepped until beginning of the slot.
OD STATUS WRITE REGISTER	
LRI 1	When on, indicates a 1 is being written in the write control status channel of LRI 1 field.
LRI 2	Same as LRI 1 (OD STATUS WRITE REGISTER) but for LRI 2 field.
MI	Same as LRI 1 (OD STATUS WRITE REGISTER) but for MI field.
XT	Same as LRI 1 (OD STATUS WRITE REGISTER) but for XTL field.
SP XT	Same as LRI 1 (OD STATUS WRITE REGISTER) but for SP XTL field.
OB 1	Turned on by a CD 2 pulse if a 1 is to be written on the read control status channel during write operations on OB field No. 1.
OB 2	Same as OB 1 (OD STATUS WRITE REGISTER) but for OB 2 field.
OB 3	Same as OB 1 (OD STATUS WRITE REGISTER) but for OB 3 field.
GFI	Same as LRI (OD STATUS WRITE REGISTER) but for GFI field.
WRITE PULSE STRETCHER	
CD WRITE REG	Turned on by write pulse at CD 3.
XT CD MARKER	Same as SP XT CD MARKER STATUS neon (write pulse stretcher) but for the XTL marker status control channel.
SP XT CD MARKER STATUS	Turned on by the CD write pulse; if select-spare-XTL-marker level is present.

TABLE 1-108. COMPUTER TO DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
MIXD STATUS	Same as LOG STATUS neon (Write PULSE STRETCHER) but for the MIXD drum.
LOG STATUS	Turned on by DTP 3.
CD STATUS REGISTER	
OB 1	Turned on by a CD 1 pulse if a 1 appears in write control status channel during write operations on OB field No. 1.
OB 2	Same as OB 1 neon (CD status register)
OB 3	Same as OB 1 neon (CD status register)
CD MARKER STATUS WR. REG. XT-CD	Turned on by the left sign bit whenever computer is programmed to write a 1 in marker status channel.
CD MARKER STATUS WR. REG. SP XT CD	Same as XT CD neon (CD marker status write register)
RD-CD	Represents the status of first FF in timing gap counter described under TIMING GAP CTR.
TIMING GAP CTR	These three neons represent status of last three bits in a 4-stage binary counter used to suppress timing pulses to allow for switching fields during CD operations on RD drum.
RD-OD	This neon represents the status of the first FF in the timing gap counter described under TIMING GAP CTR.
TIMING GAP CTR	These three neons represent status of last three bits in a 4-stage binary counter used to suppress timing pulses to allow for switching fields during OD operations on the RD drum.
TD-CD	Same as for RD-CD but for the TD drums.
TIMING GAP CTR	Same as TIMING GAP CTR but for the TD drums.
TD-OD	Same as RD-OD but for the TD drum.
TIMING GAP CTR	Same as TIMING GAP CTR but for the TD drum.
LEFT CD WRITE REG	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on main drums by the computer.
APC ERROR	
RD	If there is an error in the RD drum register count, turned on by RD index pulse.
TD	Same as RD neon (APC ERROR) but for TD drum.
AM-A	Same as RD neon (APC ERROR) but for AM-A drum.
AM-B	Same as RD neon (APC ERROR) but for AM-B drum.
MIXD	Same as RD neon (APC ERROR) but for the MIXD drum.
IC OD	Same as RD neon (APC ERROR) but for IC OD operation.
TEST	When on, indicates an error has occurred in the test APC.

TABLE 1-108. COMPUTER TO DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
CD SELECTION REG.	These six neons represent status of each bit in the index interval (R10 through R15) used to select drum and drum field for CD read or write operations.
CD MASTER SYNC	Turned on by first select-CD-4 pulse after a <i>Read</i> or <i>Write</i> instruction is given.
CD ADR COMPARE	Turned on by compare-address pulse.
CD READ	Turned on by start-read pulse and remains on during read operations.
CD WRITE	Turned on by the start-write pulse and remains on during write operations.
CD MODE READ	When on, indicates the computer is reading from drums.
CD MODE WRITE	When on, indicates the computer is writing on drums.
CD WRITE REG FULL	Turned on each time a word is transferred from the IO register to the CD write register.
OB ODD REG	Turned off by the LOG CD IX pulse; thereafter, it is alternately turned on and off by CD pulses; if OB odd is selected, reading or writing occurs each time the neon is on; if OB even is selected, reading or writing occurs when neon is off.
CD SLOT CTR LRI-1	Turned off by LRI 1 CD IX pulse, and thereafter is turned on and off by CD 3 pulses; when on, it indicates that the no-compare pulse is being gated to write circuit for write control status channel.
CD SLOT CTR LRI-2	Same as LRI-1 (CD SLOT CTR) but for LRI field No. 2.
RADAR DATA APC	The first of these 12 neons (CHK) is turned on by RD CD1; if drum register count is correct, it is turned off by the 2,048th CD 3 pulse; if not, index pulse sets the RD APC error FF; the remaining 11 neons represent status of each bit of the APC.
TRACK DISPLAY APC	Same as RADAR DATA APC but for TD drums.
AUXILIARY MEMORY A APC	Same as RADAR DATA APC but for AM-A drum.
AUXILIARY MEMORY B APC	Same as RADAR DATA APC but for AM-B drum.
MIXD APC	Same as RADAR DATA APC but for MIXD drums.
IC OD APC	Same as RADAR DATA APC but for OD operation on the IC drum field.
RIGHT CD WRITE REG	These 16 neons represent status of each bit in coded binary right half-word which is being written on main drums by the computer.

2.8.3 INPUT DRUMS Neon Group

Neon indications for other than Central Computer System-Drum System (OD) operations are explained in table 1-109.

TABLE 1-109. INPUT DRUMS NEONS

INDICATOR NEON	INDICATION
OD STATUS REG	
GFI	Turned on by an OD 1 pulse if a 1 is read in the GFI write control status channel.
SP XT OD	Same as GFI (OD STATUS REG) but for spare XTL field.

TABLE 1-109. INPUT DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
XT	Same as GFI (OD STATUS REG) but for XTL field.
MI	Same as GFI (OD STATUS REG) but for MI field.
LRI-1	Same as GFI (OD STATUS REG) but for LRI 1 field.
LRI-2	Same as GFI (OD STATUS REG) but for LRI 2 field.
OD TEST STATUS REG	
GFI	Turned on by an OD 3 pulse if there is a 0 in write control status channel of GFI field.
SP XT	Same as GFI (OD TEST STATUS REG) but for spare XTL field.
XT	Same as GFI (OD TEST STATUS REG) but for XTL field.
MI	Same as GFI (OD TEST STATUS REG) but for MI field.
LRI-1	Same as GFI (OD TEST STATUS REG) but for LRI 1 field.
LRI-2	Same as GFI (OD TEST STATUS REG) but for LRI 2 field.
OD PULSE WRITE STRETCHER	
GFI	On until write pulse turns it off at OD 3.
SP XT	Same as GFI (OD PULSE WRITE STRETCHER) but for the spare XTL field.
XT	Same as GFI (OD PULSE WRITE STRETCHER) but for the XTL field.
MI	Same as GFI (OD PULSE WRITE STRETCHER) but for the MI field.
LRI-1	Same as GFI (OD PULSE WRITE STRETCHER) but for the LRI 1 field.
LRI-2	Same as GFI (OD PULSE WRITE STRETCHER) but for the LRI 2 field.
OD FULL ALARM	
GFI	When on, indicates drum field is full and no more data can be written until field is readied by the computer; the OD IX pulse turns this neon on if the write control status channel contains all 1's.
SP XT	Same as GFI (OD FULL ALARM) but for spare XTL field.
XT	Same as GFI (OD FULL ALARM) but for XTL field.
MI	Same as GFI (OD FULL ALARM) but for MI field.
LRI-1	Same as GFI (OD FULL ALARM) but for LRI 1 field.
LRI-2	Same as GFI (OD FULL ALARM) but for LRI 2 field.
OD FULL ALARM SET	
GFI	Turned on by an OD IX pulse.
SP XT	Same as GFI (OD FULL ALARM SET) but for spare XTL field.
XT	Same as GFI (OD FULL ALARM SET) but for XTL field.
MI	Same as GFI (OD FULL ALARM SET) but for MI field.
LRI-1	Same as GFI (OD FULL ALARM SET) but for LRI 1 field.

TABLE 1-109. INPUT DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
LRI-2	Same as GFI (OD FULL ALARM SET) but for LRI 2 field.
CD STATUS WRITE REG	
GFI	Turned on whenever a 1 is to be written in GFI read control status channel.
SP XT	Same as GFI (CD STATUS WRITE REG) but for spare XTL field.
XT	Same as GFI (CD STATUS WRITE REG) but for XTL field.
MI	Same as GFI (CD STATUS WRITE REG) but for MI field.
LRI-1	Same as GFI (CD STATUS WRITE REG) but for LRI 1 field.
LRI-2	Same as GFI (CD STATUS WRITE REG) but for LRI 2 field.
OD WRITE REG FULL	
GFI	Turned on by GFI data available pulse, indicating a full write register.
SP XT	Same as GFI (OD WRITE REG FULL) but for spare XTL field.
XT	Same as GFI (OD WRITE REG FULL) but for XTL field.
MI	Same as GFI (OD WRITE REG FULL) but for MI field.
LRI-1	Same as GFI (OD WRITE REG FULL) but for LRI 1 field.
LRI-2	Same as GFI (OD WRITE REG FULL) but for LRI 2 field.
OD TEST MRKR CHAN STATUS SP XT	Turned on by an OD 3 pulse whenever the XTL OD marker channel status neon is on.
OD TEST MRKR CHAN STATUS XT	Same as SP XT (OD TEST MRKR CHAN STATUS) but for XTL field.
OD MRKR CHAN STATUS SP XT	Turned on by an OD 1 pulse whenever a 1 is read in the XTL marker channel.
OD MRKR CHAN STATUS XT	Same as SP XT (OD MRKR CHAN STATUS) but for XTL field; it is turned off by an OD 4 pulse and remains off until a 1 is read again.
GFI OD REL TIME CTR	
SYNC	Once every 8 seconds a sync-relative-time-counter pulse comes from the computer and turns this neon on.
STEP	Turned on by first OD 1 pulse after INPUT (GFI OD REL TIME CTR) goes on.
INPUT	Turned on every 1/4 second by step-relative-time-counter pulses.
OD SLOT CTR LRI	Turned on by an OD-IX pulse and complemented by OD 1 pulses; the LRI demand pulses are sent only when this neon is on; provides a slot consisting of two registers.
INDEX PULSE	
AM-A	The left neon is connected to one side of a FF while the second neon is connected to the other side; the AM-A index pulse complements the FF; consequently, each neon will be on 20.48 milliseconds and then turned off while the other one is on for same time interval.

TABLE 1-109. INPUT DRUMS NEONS (cont'd)

INDICATOR NEON	INDICATION
RD	The left neon is turned on by the RD-CD IX pulse and remains on approximately 1 millisecond; it is then turned off and the right neon turned on by the RD-OD IX pulse; the right neon remains on approximately 19 milliseconds and then process is repeated.
AM-B	Same as AM-A (INDEX PULSE) but for AM-B drum.
MIXD	Same as RD (INDEX PULSE) but for MIXD drums.
TD	Same as RD (INDEX PULSE) but for TD drum.
LOG	Same as RD (INDEX PULSE) but for LOG drum.
LEFT GFI OD WRITE REGISTER	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on drums by GFI equipment.
LEFT SP XT OD WRITE REGISTER	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on drums by spare XTL equipment.
LEFT XT OD WRITE REGISTER	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on drums by XTL equipment.
LEFT MI OD WRITE REGISTER	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on drums by MI equipment.
LEFT LRI OD WRITE REGISTER	The first neon is parity bit neon; the remaining 16 neons represent status of each bit in the coded binary left half-word which is being written on drums by LRI equipment.
LEFT CHECK REGISTER	These 17 neons represent status of each bit in the binary left half-word of the test pattern being read from the drum under manual test.
RIGHT GFI OD REGISTER	These 16 neons represent status of each bit in the coded binary right half-word which is being written on drums by GFI equipment.
RIGHT SP XT OD WRITE REG	These 16 neons represent status of each bit in the coded binary right half-word which is being written on drums by spare XTL equipment.
RIGHT XT OD WRITE REG	These 16 neons represent status of each bit in the coded binary right half-word which is being written on drums by XTL equipment.
RIGHT MI OD WRITE REG	These 16 neons represent status of each bit in the coded binary right half-word which is being written on drums by MI equipment.
RIGHT LRI OD WRITE REG	These 16 neons represent status of each bit in the coded binary right half-word which is being written on drums by the LRI equipment.
RIGHT CHECK REGISTER	These 16 neons represent status of each bit in the binary right half-word of test pattern being read from drum under manual test.

2.8.4 DISPLAY FIELDS Neons Group

Neon indications for selected display field operations are explained in table 1-110.

TABLE 1-110. DISPLAY FIELDS NEONS

INDICATOR NEON	INDICATION
TRACK DISPLAY FIELDS	
OD FIELD CTR	These three neons represent status of each bit in a binary number which selects TD field to be read.
OD REG CTR	These three neons represent status of each bit in a binary 8 counter which counts OD 2 pulses.
OD SYNC	Turned on by end-RD-start-TD pulse.
OD SELECTED	Turned on by an OD IX pulse after OD SYNC (TRACK DISPLAY FIELDS) is on; stays on until all of the TD fields are read.
OD SLOT CTR	These four neons represent status of each bit of binary counter which counts to 12.
DIGITAL DISPLAY FIELDS	
OD SYNC	Turned on by the start-DD-read pulse; the neon remains on during DD reading.
OD REG CTR	These six neons represent status of each bit in a binary counter which counts to 64.
INTERCOMMUNICATION FIELD	
IC OTHER READ SYNC	In operate condition, turned on by an OD 1 pulse if IC OTHER READ (INTERCOMMUNICATION FIELD) is on.
IC OTHER READ	In operate condition, turned on by an OD 1 pulse if IC OTHER READ SYNC (INTERCOMMUNICATION FIELD) is on.
IC OTHER COMPARE	In operate condition, turned on by an OD 1 pulse if IC OTHER READ SYNC (INTERCOMMUNICATION FIELD) is on; neon is turned off and the IC read-sample pulse is not generated.
RADAR DISPLAY FIELDS	
OD SYNC	Turned on by an end-TD-start-RD pulse.
OD SELECTED	Turned on by an OD IX pulse after OD SYNC (RADAR DISPLAY FIELDS) is on; this neon remains on during RD read operations.
DISPLAY DIM	When on, this neon indicates that the oldest data is being displayed dimly; when off, the latest data is being displayed brightly.
OD FIELD CTR	These three neons represent status of each bit in a 3-stage binary counter which indicates number of fields read since beginning read operations.
OD SCAN CTR SYNC	Turned on by an OD 1 pulse after OD SCAN CTR STEP (RADAR DISPLAY FIELDS) is on.
OD SCAN CTR STEP	Turned on by a step-scan-counter pulse from the computer.
OD TRANSFER SCAN CTR	Turned on by an OD IX pulse, if OD SELECTED (RADAR DISPLAY FIELDS) is off.
OD SCAN CTR	These four neons represent status of each bit in a binary number which represents number of the field on which the computer will write the next RD reading.

TABLE 1-110. DISPLAY FIELDS NEONS (cont'd)

INDICATOR NEON	INDICATION
OD REG CTR	These three neons represent status of each bit in a binary counter which counts to 5.
OD FIELD SELECT CTR	These four neons represent status of each bit in a binary number which selects RD field to be read.
OB FIELDS	
OB OD RESTART	Turned on by restart pulse from output equipment; must be on in order to read the OB fields in operate condition.
OB OD FIELD SWITCH CTR	These two neons represent the status of each bit in a binary counter used to select the OB field for reading; it counts the OB field switch pulses from the OB field switch counter (OB OD GAP CTR and OB CD GAP CTR).
OB 1 through 3	Turned on whenever a 1 is to be written in the write control status channel of the OB 1 through 3 field.

2.8.5 Test Indicators and Controls

(manual) test or computer tests. These indicators and controls are discussed in table 1-111.

2.8.5.1 General

Certain error indications and test controls are made available on this panel to aid maintenance personnel when the Drum System is performing semiautomatic

2.8.5.2 TEST Neon Group

An explanation of each neon indicator and switch in the TEST group is given in table 1-111.

TABLE 1-111. TEST NEONS

INDICATOR NEON	INDICATION
MANUAL TEST	
MANUAL TEST ERROR	Turned on in test status by a precessing error, a test APC error, or a status channel error; in operate status; it is turned on by a relative-time-counter-error pulse; turned off manually with an alarm-reset pulse (RESET DRUM ALARMS switch).
PRECESSING ERROR	At the beginning of a precession test, this neon is manually turned on with the alarm-reset pulse (RESET DRUM ALARMS switch); if a precession error occurs, neon stays on and MANUAL TEST ERROR (MANUAL TEST) will go on; if no error occurs, neon is turned off by APC end carry and turned on again by error-sense pulse; however, the error FF if not set.
STATUS TEST READ SAMPLE	In a status check for 1, this neon is turned off by select-CD-4; if a 1 appears in the status channel, the read-sample pulse turns this light on; if a 0 appears, neon is not turned on and error pulse is generated by SEL CD 4; in a status check for 0, this neon remains off unless an error occurs, in which case the neon is turned on by the read-sample pulse.
TEST PATTERN	Turned on by manual-start-read or manual-start-write pulse; the associated FF, in conjunction with the pattern selector switch and the complement test pattern switch, determines test pattern in reading and writing.
TC AND IX WRITE	Turned on by DTP 1 while the 30-millisecond single-shot level is up during tape core and index channel writing; this gates DTP 3 pulses to the WR pulse stretcher FF described above; turned off by DTP 1 after the single-shot level drops.

TABLE 1-111. TEST NEONS (cont'd)

INDICATOR NEON	INDICATION
TC AND IX PULSE STRETCHER	Turned on by every DTP 3 while the 30-millisecond single-shot level is up during tape core and index channel writing.
TC AND IX WR REG	Turned on by a DTP 1 pulse when 30-millisecond single-shot level is up during tape core and index channel writing; this causes all 0's to be written in the index channel until the single-shot level drops, at which time neon goes off and a single 1 is written in the index channel.
COMPUTER TEST	
COMPUTER TEST SYNC	Turned on by the index pulse of drum under test and remains on until turned off by the read-write-reset pulse.
COMPUTER TEST WR	Turned on by a start-write pulse; remains on during test write operations until turned off by read-write-reset pulse.
TD READ SAMPLE	Turned on by start-TD pulses of SD cycle; turned off by start-RD pulse.
TD COMPUTER TEST START	Turned on by a DTP 1 pulse when the SD pretest level is up; the next DTP I pulse turns the neon off.
TEST APC	In operate condition, test APC acts as the log drum APC, as described under RADAR DATA APC, but for the log drum; in test condition, it can be used as the APC for any drum or be used to test for precession errors in RD, TD, or DD reading.
RADAR DATA APC	The first of these 12 neons (CHK) is turned on by RD CD 1. If the drum register count is correct, the neon is turned off by the 2,048th CD 3 pulse; the remaining 11 neons represent the status of each bit of the APC.

2.8.5.3 DRUM SELECT Controls and Indicators

An explanation of each indicator light and switch

in the DRUM SELECT controls and indicators group is given in table 1-112.

TABLE 1-112. DRUM SELECT CONTROLS AND INDICATORS

INDICATOR LIGHT	CONTROL	FUNCTION
AM-A EXCITER LIGHT ON		This light illuminates when DRUM SELECT switch is operated to AM-A position or ALL position, and indicates that exciter light for writing AM-A drum timing channel is on.
AM-B EXCITER LIGHT ON		Same as AM-A EXCITER LIGHT ON but for AM-B drum and the DRUM SELECT AM-B position.
TD EXCITER LIGHT ON		Same as AM-A EXCITER LIGHT ON but for TD drum and the DRUM SELECT TD position.
RD EXCITER LIGHT ON		Same as AM-A EXCITER LIGHT ON but for RD drum and DRUM SELECT RD position.
LOG EXCITER LIGHT ON		Same as AM-A EXCITER LIGHT ON but for LOG drum and DRUM SELECT LOG position.
MIXED EXCITER LIGHT ON		Same as AM-A EXCITER LIGHT ON but for MIXD drum and DRUM SELECT MIXD position.

TABLE 1-112. DRUM SELECT CONTROLS AND INDICATORS (cont'd)

INDICATOR LIGHT	CONTROL	FUNCTION
DRUM ERASE		This light illuminates when ERASE AND TIMING WRITE switch is operated; it remains on for 30 seconds while one drum or all drums are erased.
	RESET DRUM ALARMS (push-button)	Operation of this switch resets all main drum alarm FF's.
	DESELECT DRUMS (push-button)	When computer is in test status, operation of this switch causes certain FF's in the drum units to be cleared.
IC OD READ BUS RELAY		This light illuminates when computer is in operate condition and indicates that the other computer can read from the IC drum field.
ALL MOTORS ON		This light illuminates when all drum motors are turned on.
COMPUTER TEST DRUMS		This light illuminates when DRUMS OPERATE-COMPUTER TEST switch is operated to the COMPUTER TEST position; it stays on during computer testing.
MANUAL TEST DRUMS		This light goes on when DRUM OPERATE-MANUAL TEST switch is operated to MANUAL TEST position; it remains on during manual test.
	ERASE AND TIMING WRITE (pushbutton)	If the drums are in manual test status and the SELECTOR switch on the manual test panel is set to off, the operation of this switch erases and rewrites the timing tracks on the drums selected by the rotary switch on the maintenance console.
	DRUM SELECT (rotary switch)	This switch selects the drum to be erased when the ERASE AND TIMING WRITE switch is operated; it can select any particular drum or all drums; it is so interlocked that unless it is in the off position, the ERASE AND TIMING switch at the drum unit is inoperative.
	OPERATE-MANUAL TEST (lever switch)	Operation of this switch when the computer is in test status causes the drums to go into manual test status. The drums go to operate status if this switch is moved to the OPERATE position or if the computer goes to operate status.
	OPERATE-COMPUTER TEST (lever switch)	Operation of this switch when the computer is in test status causes the drums to go into computer test status. The drums go to operate status if this switch is moved to the OPERATE position or if the computer goes to operate status.

2.9 AUXILIARY DRUMS INDICATOR PANEL

2.9.1 General

The AUXILIARY DRUMS indicator panel contains indicators and certain controls for the auxiliary drums of the Drum System. Drum System auxiliary drums contain stored programs which are utilized by the Central

Computer System to control central operations. Indications of auxiliary drum operations enable operating personnel to determine if the auxiliary drums are functioning correctly. In case of a malfunction in the auxiliary drum portion of the Drum System, controls and indicators on this panel aid maintenance personnel in localiz-

ing the malfunction in conjunction with semiautomatic (manual) tests or computer tests. Figure 1-213 shows the AUXILIARY DRUMS indicator panel.

2.9.2 Test Controls and Indicators

An explanation of test indicator lights and controls on this panel is given in table 1-113.

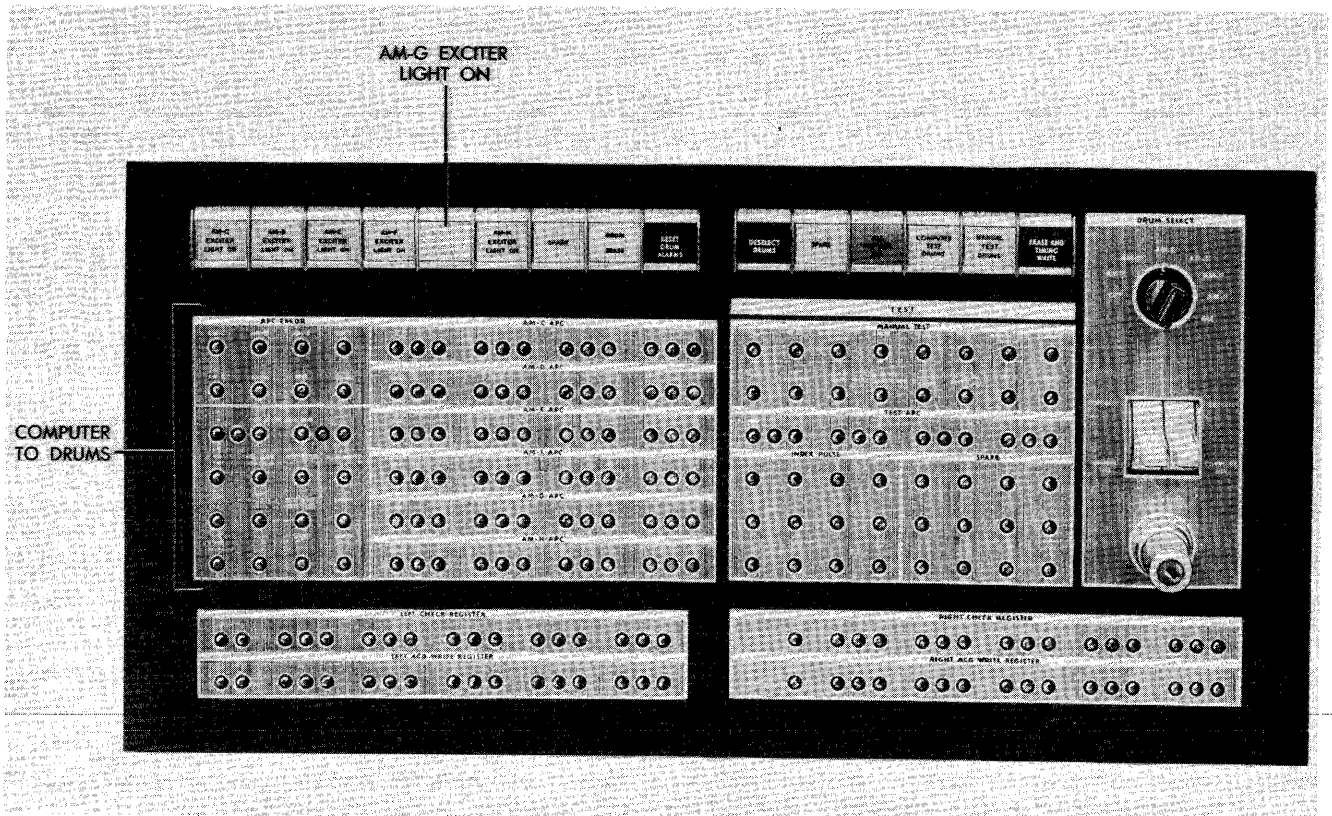


Figure 1-213. Auxiliary Drum Group Controls and Indicators

TABLE 1-113. AUXILIARY DRUMS CONTROL AND INDICATOR GROUP

INDICATOR	CONTROL	FUNCTION
AM-C EXCITER LIGHT ON		Illuminates when the DRUM SELECT switch is operated to the AM-C position, and indicates that the exciter light for writing the AM-C drum timing channel is on.
AM-D EXCITER LIGHT ON		Same as AM-C EXCITER LIGHT ON but for the AM-D drum and the DRUM SELECT switch AM-D position.
AM-E EXCITER LIGHT ON		Same as AM-C EXCITER LIGHT ON but for the AM-E drum and the DRUM SELECT switch AM-E position.
AM-F EXCITER LIGHT ON		Same as AM-C EXCITER LIGHT ON but for AM-F drum and DRUM SELECT switch AM-F position.
AM-G EXCITER LIGHT ON		Same as AM-C EXCITER LIGHT ON but for the AM-G drum and DRUM SELECT switch AM-G position.
AM-H EXCITER LIGHT ON		Same as AM-C EXCITER LIGHT ON but for AM-H drum and DRUM SELECT switch AM-H position.

TABLE 1-113. AUXILIARY DRUMS CONTROL AND INDICATOR GROUP (cont'd)

INDICATOR	CONTROL	FUNCTION
DRUM ERASE		Illuminates when the ERASE AND TIMING WRITE switch is operated; remains on for 30 seconds while one drum or all drums of the auxiliary Drum System are erased.
	RESET DRUM ALARMS (pushbutton)	Operation of this switch resets all auxiliary drum alarm FF's.
	DESELECT DRUMS (pushbutton)	When the computer is in the test status, the operation of this switch causes certain FF's in the auxiliary drum units to be cleared.
ALL MOTORS ON		Illuminates when all auxiliary drum motors are turned on.
COMPUTER TEST DRUMS		Illuminates when COMPUTER-TEST OPERATE switch is placed in the COMPUTER TEST position; it stays on during computer testing.
MANUAL TEST DRUMS		Illuminates when OPERATE-MANUAL TEST switch is placed in the MANUAL TEST position; it remains on during manual test.
	ERASE AND TIMING WRITE (pushbutton)	If the auxiliary drums are in manual test status and the SELECTOR switch on the manual test panel is set to off, the operation of this switch erases and rewrites the timing tracks on the drums selected by the rotary switch on the maintenance console.
	DRUM SELECT (rotary switch)	This switch selects the auxiliary drum which is erased when the ERASE AND TIMING WRITE switch is operated. This switch can select any particular drum or all drums; it is so interlocked that unless it is in the off position, the ERASE AND TIMING WRITE switch at the auxiliary drum unit is inoperative.
	OPERATE-MANUAL TEST (lever switch)	Operation of this switch when the computer is in test status causes the auxiliary drums to go into manual test status; the drums go to operate status if this switch is moved to the OPERATE position or if the computer goes to operate status.
	OPERATE-COM-PUTER TEST (lever switch)	Operation of this switch when the computer is in test status causes the auxiliary drums to go into computer test status; the drums go to operate status if the switch is moved to the OPERATE position or if the computer goes to the operate status.
	ERASE AND INTERLOCK OFF (key-operated switch)	When the auxiliary drums are in any status, unlocking of this switch allows auxiliary drums to be rewritten; if the drums are in manual test status and the switch is unlocked, the auxiliary drums may be erased; when this switch is locked, the auxiliary drum cannot be erased or rewritten.

2.9.3 TEST Neon Group

An explanation of TEST neon indications when the auxiliary drum portion of the Drum System is performing semiautomatic (manual) tests or computer tests is given in table 1-114.

TABLE 1-114. TEST NEONS

INDICATOR NEONS	INDICATION
MANUAL TEST ERROR	This neon is turned on by an auxiliary drum test APC error; it is turned off manually.
MANUAL TEST WRITE	Turned on by start-write pulse; it remains on during test write operations until turned off by the read-write-reset pulse.
TEST PATTERN	Turned on by manual-start-read or manual-start-write pulse.
TC & IX WRITE	Turned on by every DTP 3 during tape core and index channel writing.
TC & IX WR PULSE STRETCHER	Turned on by every DTP 3 during tape core and index channel writing.
TC AND IX WRITE REGISTER	Turned on by a DTP 1 pulse during tape core and index channel writing.
MANUAL TEST READ	Turned on by a start-read pulse; remains on during test operations until turned off by the read-write-reset pulse.
TEST APC	Operates same as AM-C APC (AUX DRUM) but can be used as the APC for any one of the six auxiliary drums.
INDEX PULSE AM-C	The left neon is connected to 1 side of a FF while the second neon is connected to the other side; the AM-C index pulse complements the FF; consequently, each neon is on alternately.
AM-D	Same as AM-C (INDEX PULSE) but for the AM-D drum.
AM-E	Same as AM-C (INDEX PULSE) but for the AM-E drum.
AM-F	Same as AM-C (INDEX PULSE) but for the AM-F drum.
AM-G	Same as AM-C (INDEX PULSE) but for the AM-G drum.
AM-H	Same as AM-C (INDEX PULSE) but for the AM H drum.

2.9.4 COMPUTER TO DRUMS Neon Group

An explanation of COMPUTER TO DRUMS neon indicators is given in table 1-115.

TABLE 1-115. COMPUTER TO DRUMS NEONS

INDICATOR NEONS	INDICATION
APC ERROR AM-C	If there is an error in the AM C drum register count, this neon is to be illuminated by the AM-C index pulse.
AM-D	Same as AM-C (APC ERROR) but for the AM-D drum.
AM-E	Same as AM-C (APC ERROR) but for the AM-E drum.
AM-F	Same as AM-C (APC ERROR) but for the AM-F drum.
AM-G	Same as AM-C (APC ERROR) but for the AM-G drum.

TABLE 1-115. COMPUTER TO DRUMS NEONS (cont'd)

INDICATOR NEONS	INDICATION
AM-H	Same as AM-C (APC ERROR) but for the AM-H drum.
TEST	When on, this neon indicates that an error has occurred in the test APC.
ACD SELECTION REGISTER	These six neons represent the status of each bit in the index interval (L10 through L15) used to select the auxiliary drum and drum field for ACD read or write operation.
ACD MASTER SYNC	Turned on by the first select-ACD-4 pulse after a <i>Read</i> or <i>Write</i> instruction is given; it goes off when a disconnect pulse comes in.
ACD ADR COMPARE	Turned on by the compare-address pulse; turned off as soon as a start-read, start-write, or no-compare pulse comes in.
ACD READ	Turned on by the start-read pulse and remains on during reading operations; turned off by disconnect pulse.
ACD WRITE	Turned on by the start-write pulse and remains on during write operations; turned off by a disconnect pulse.
READ (ACD MODE)	When on, it indicates that the computer is reading from auxiliary drums.
WRITE (ACD MODE)	When on, it indicates that the computer is writing on auxiliary drums.
ACD WRITE REGISTER FULL	Turned on each time a word is transferred from the IO register to the ACD write register.
ACD WR REG WR PULSE STRETCHER	Turned on by the write pulse at ACD 3 and turned off 1.7 microseconds later.
AM-C APC	The first of these twelve neons is turned on by ACD 1; if the drum register count is correct, the neon is turned off by the 2,048th ACD 3 pulse; if not, the index pulse sets the AM-C APC error FF; the remaining 11 neons represent the status of each bit of the APC.
AM-D APC	Same as AM-C APC but for the AM-D drum.
AM-E APC	Same as AM-C APC but for the AM-E drum.
AM-F APC	Same as AM-C APC but for the AM-F drum.
AM-G APC	Same as AM-C APC but for the AM-G drum.
AM-H APC	Same as AM-C APC but for the AM-H drum.

2.9.5 Auxiliary Drums Test Register Neons

The two rows of neons at the bottom of the auxiliary drums panel are useful in evaluating the results of

manual tests of these drums. The exact function of each row of these neons is given in table 1-116.

TABLE 1-116. AUXILIARY DRUMS TEST REGISTER NEONS

NEON REGISTER	INDICATION
LEFT CHECK REGISTER	These 17 neons represent the status of each bit in the binary left half-word of the test pattern being read from the drum under manual test.
RIGHT CHECK REGISTER	These 16 neons represent the status of each bit in the binary right half-word of the test pattern being read from the drum under manual test.
LEFT ACD WRITE REGISTER	The first neon is the parity bit neon; the remaining 16 neons represent the status of each bit in the coded binary left half-word which is being written on auxiliary drums by the computer.
RIGHT ACD WRITE REGISTER	These 16 neons represent the status of each bit in the coded binary right half-word which is being written on auxiliary drums by the computer.

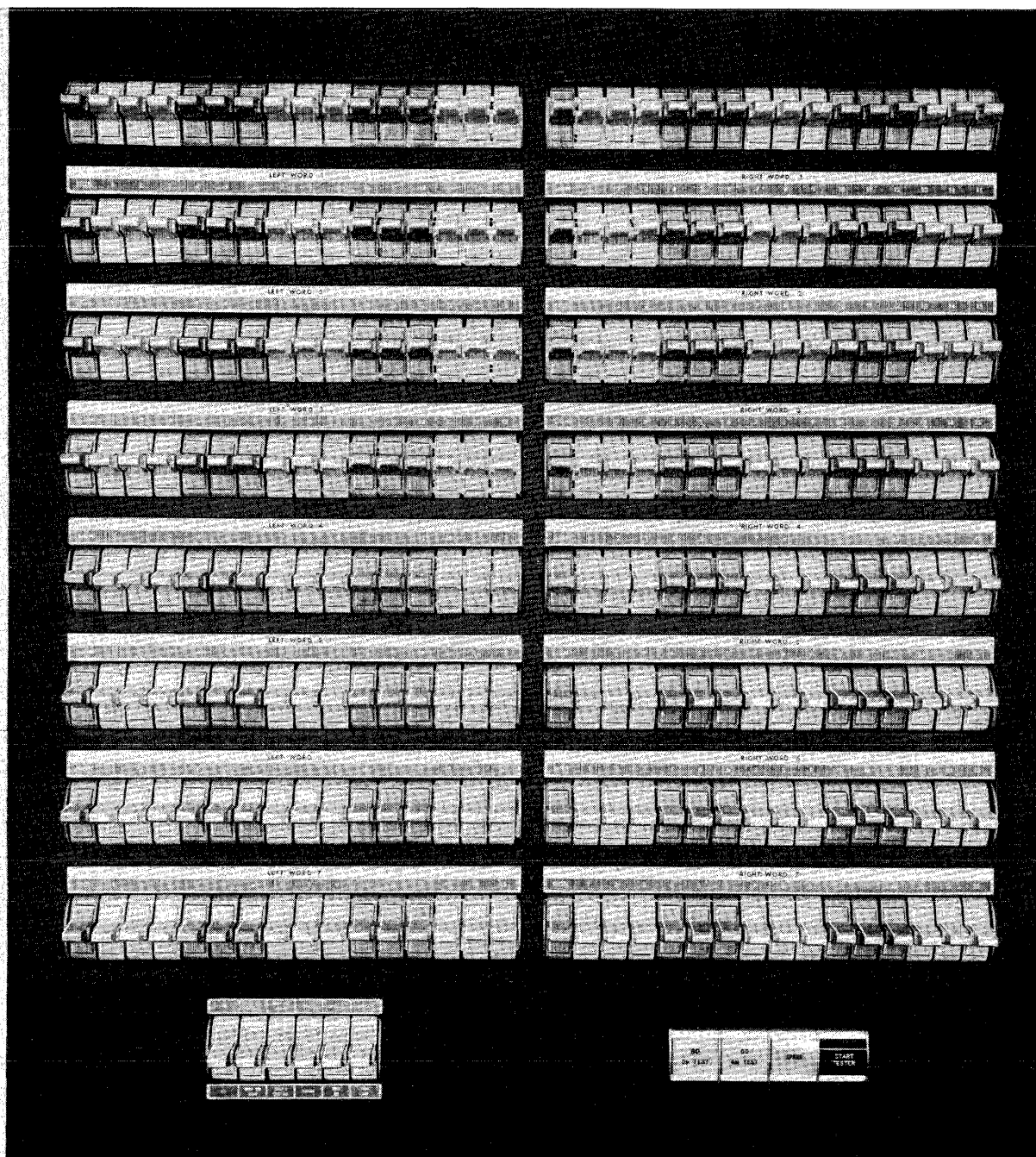


Figure 1-214. Display Tester Controls and Indicators

2.10 DISPLAY TESTER CONTROL PANEL

2.10.1 General

The display tester control panel (located in the upper right corner of the duplex maintenance console) houses switches and indicators used in testing the Display System apart from other systems. Test signals generated under control of the display tester simulate the signals normally supplied to the Display System via the Drum System. These test signals can cause displays to

appear on any or all of the situation display tubes or digital display tubes mounted in consoles whose status matches that of the display generator element receiving signals from the display tester. The displays generated by the tester can be used either in testing generator elements or in testing or alignment of individual display consoles. The panel to the left of the display tester control panel houses neon lights which indicate circuit actions within the generator elements. These

neons are useful in assessing the result of an operation using the display tester.

2.10.2 BIT STORAGE CONTROL Lever Switches

The BIT STORAGE CONTROL lever switches include 256 switches mounted in 8 horizontal rows of 32 switches each. (See fig. 1–214.) These switches comprise eight registers which can contain up to eight 1-word radar data messages, one 8-word track display message, or up to two 4-word digital display messages. When the display tester is used, these switches are positioned to make up one of the three types of display messages just mentioned. (See figs. 1–146, 1–150, and 1–154.) Each lever switch controls the status of a particular bit in the simulated drum word message layout; a switch in the depressed position inserts a 1; in the center or neutral position, it inserts a 0.

2.10.3 Control Switches and Indicators

2.10.3.1 Uses

The switches on the display tester control panel, other than the BIT STORAGE CONTROL switches, select the type of test to be run and initiate the operation of the tester. The indicators inform the maintenance operator of the nature of the test being run. Each control switch is described separately.

2.10.3.2 OPERATE-TEST DD Switch

The OPERATE-TEST DD lever switch (the right-most switch in the lower left-hand group), when in TEST DD position, causes the digital display generator ele-

ment (DDGE) to receive input signals from the display tester rather than from the Drum System. When this switch is placed in TEST DD position, only one further switch selection, the DISP DD CONT-DISP 1 CYC DD switch, is necessary (in addition to setting the BIT STORAGE CONTROL switches) before initiating a test of the digital display equipment. Similarly, returning this switch to the OPERATE position while a digital display test is running terminates that test.

2.10.3.3 OPERATE-TEST SD Switch

The OPERATE-TEST SD lever switch (located to the left of the OPERATE-TEST DD switch), when in TEST SD position, substitutes the display tester for the Drum System as a signal source for the situation display generator element (SDGE). With this switch in TEST SD position, at least one other switch selection (other than setting the BIT STORAGE CONTROL switches) is required, using the TD-RD switch, before initiating test operation. Returning the OPERATE-TEST SD switch to the OPERATE position while a test of the situation display equipment is running terminates the test.

2.10.3.4 DISP DD CONT-DISP 1 CYC DD Switch

With the OPERATE-TEST DD switch in TEST DD position, the DISP DD CONT-DISP 1 CYC DD lever switch selects operation of the display tester so that it either continually initiates rewriting of digital display tubes or writes them once and then stops.

TABLE 1–117. DISPLAY TESTER CONTROL PANEL SWITCHES AND INDICATORS

CONTROL OR INDICATOR	FUNCTION
BIT STORAGE CONTROL lever switches	Sets the contents of eight simulated drum words for delivery to either the DDGE or SDGE under test.
TD-RD lever switch	Establishes BIT STORAGE CONTROL words as TD or RD message when OPERATE-TEST SD switch is in TEST.
DISP RD BRIGHT-DISP RD DIM lever switch	Causes display of RD messages (indicated by setting of TD-RD switch to RD) as present (bright) or past (dim).
DISP DD CONT-DISP 1 CYC DD lever switch	Causes either continuous rewriting of DD tubes (with OPERATE-TEST DD switch in TEST) or one writing of DD tubes initiated by START TESTER pushbutton.
OPERATE-TEST SD lever switch	When in TEST position, connects tester to SDGE; returning switch to OPERATE terminates test.
OPERATE-TEST DD lever switch	When in TEST position, connects tester to DDGE; returning switch to OPERATE terminates test.
SD ON TEST indicator	On when OPERATE-TEST SD switch in TEST position.
DD ON TEST indicator	On when OPERATE-TEST DD switch in TEST position.
START TESTER pushbutton	Initiates test set up by other switches.

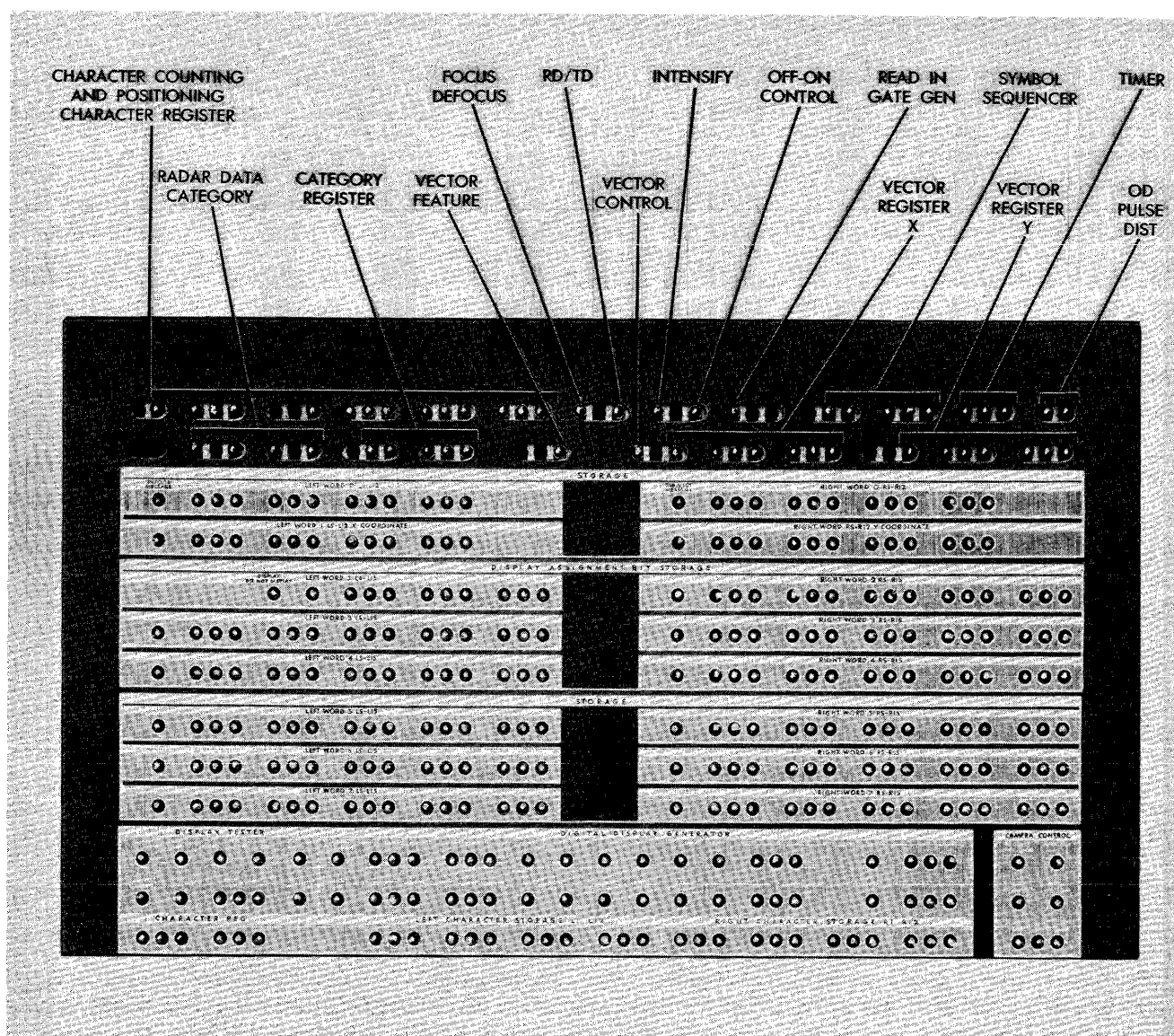


Figure 1-215. Display System Neons

2.10.3.5 TD-RD Switch

The TD-RD lever switch selects operation of the display tester so that, with the OPERATE-TEST SD switch in TEST SD position, the contents of the BIT STORAGE CONTROL switches are accepted by the SDGE as either a track display (TD) message or as eight radar data (RD) messages.

2.10.3.6 DISP RD BRIGHT-DISP RD DIM Switch

With the TD-RD switch in RD position, the DISP RD BRIGHT-DISP RD DIM switch selects the intensification of RD messages. When this switch is in DISP RD BRIGHT position, the RD messages set up on the BIT STORAGE CONTROL are displayed brightly; i.e., as present radar data. When this switch is in DISP RD

DIM position, RD messages are displayed dimly; i.e., as past radar data.

2.10.3.7 STARTER TESTER Pushbutton

Once the selection of type of test is made with the other switches on the display tester control panel, a test is started by depressing the START TESTER pushbutton.

2.10.3.8 Indicators

Two indicators are provided on the display tester control panel: the SD ON TEST and the DD ON TEST indicator lights. Each light is on when the corresponding OPERATE-TEST switch is in TEST position. The functions of all of these controls and indicators are summarized in table 1-117.

2.11 DISPLAY GENERATORS INDICATORS

2.11.1 General

The display generators neon indicators (located above the WARNING LIGHTS REGISTER indicators on the display and warning lights register neons panel) indicate various operations performed by the Display System. (See fig. 1-215.) Maintenance operators concerned with the operation of the Display System are provided visual aid by the display generators indicators, which indicate circuit functions in the situation display and digital display generators during test procedures as well as during normal operation. In addition, indicators

are provided for the display tester element and the SD camera circuits, allowing a visual check of these circuits when they have been brought into operation.

The display generators neon indicators are divided into five groups, each concerned with a specific Display System function. The function of each neon is described in the following text.

2.11.2 SITUATION DISPLAY GENERATOR

Neon Group

The function of each neon in the SITUATION DISPLAY GENERATOR neon group is given in table 1-118.

TABLE 1-118. SITUATION DISPLAY GENERATOR NEONS

INDICATOR NEON	INDICATION
CHARACTER COUNTING AND POSITIONING CHARACTER REGISTER	First six neons represent the binary code number of the maxtrix symbol to be displayed; seventh, eighth, eleventh, and twelfth neons determine the position of the format with respect to the vector; only one of these will be on for a given format position; remaining four neons used to position individual characters and symbols within the format.
FOCUS DEFOCUS	Turned on by OD 1 at timer = 4 to defocus the scopes whenever a character or symbol is to be displayed; neon turned off by OD 4 at timer = 3 to focus the scopes.
RD/TD	Turned on by the start-TD pulse; indicates that the TD fields are being read and displayed; turned off by the start-RD pulse and remains off during RD reading and display.
INTENSIFY	Turned on by an OD 1 at timer = 5 for a tabular message or by an OD 4 at timer = 3 for a vector message in order to unblank the display scopes; turned off by an OD 3 pulse at timer = 7 to end the display of a particular character or vector.
ON-OFF CONTROL	Turned on at OD 4 by a word-on-way pulse; while on, OD pulses are fed to the OD pulse distributor (X4); this in turn feeds OD 3 pulses to the timer (X5); at the end of the message, neon turned off by an OD 3 pulse.
READ IN GATE GEN	Turned on at OD 4 by the word-on-way pulse; when on, gates RD-bright or RD-dim pulses and also an OD 1 pulse which adds 1 to the symbol sequencer; this same OD 1 pulse turns the neon off.
SYMBOL SEQUENCER	These five neons represent the status of each bit in a scale-of-32 counter which is used to store and display the words and symbols at the proper time.
TIMER	These three neons represent the status of each bit in a scale-of-8 counter used to count OD 3 pulses; thus, the timer provides pulses of 10-microsecond duration.
OD PULSE DIST	These two neons represent the status of the bits in a scale-of-4 counter used to distribute OD 1, OD 2, OD 3, and OD 4 pulse over separate lines after having all of them arrive on a single line; OD 1 pulses are distributed on binary 00, OD 2 pulses on 01, OD 3 pulses on 11, and OD 4 pulses on 10.

TABLE 1-118. SITUATION DISPLAY GENERATOR NEONS (cont'd)

INDICATOR NEON	INDICATION
RADAR DATA CATEGORY	These four neons represent the status bits for the radar category; first neon is on for radar bright, indicating most recent information; second neon is on for radar dim, indicating less recent information; third neon indicates the identity (on - IFF, off - not IFF) and the last neon indicates whether information is correlated or not correlated (on - correlated, off - not correlated).
CATEGORY REGISTER	This register indicates the status of bits LS through L4 of word 2 which are the category bits for the TD message.
VECTOR FEATURE	Turned on by an OD 1 pulse when the point feature line is up on timer = 6; this indicates that the SD bypass feature is in effect, which means that the vector will be displayed and cannot be turned off by feature selection switches; neon is turned off when the character position counter is advanced.
VECTOR CONTROL	Turned on by OD 3 at timer = 7 to start the vector display; vector is traced out for 50 microseconds, at which time an OD 3 pulse at timer = 7 turns this neon off.
VECTOR REGISTER	These 14 neons represent the status of the bits which specify the magnitude and direction of the vector to be displayed; there are seven X co-ordinate bits and seven Y co-ordinate bits.

2.11.3 Display Registers Neon Group

The display registers group of neons consists of the WORD 0 through WORD 7 registers, including the STORAGE, DISPLAY ASSIGNMENT BIT STORAGE, and STORAGE register neons. Every neon indicator in this group of registers indicates the status of a particular bit position. An illuminated neon indicates that the bit position contains a 1. If the neon is not illuminated, this indicates that the bit position contains a 0. Each row of neons represents the contents of these registers.

In order to be able to read the contents of these registers with comparative ease, the neons are grouped in series of threes so the octonary equivalent can be read directly. These register neons are used primarily during testing since, during normal operations, their indications change too rapidly to be perceived visually.

2.11.4 DISPLAY TESTER Neon Group

The function of each neon in the DISPLAY TESTER neon group is given in table 1-119.

TABLE 1-119. DISPLAY TESTER NEONS

INDICATOR NEON	INDICATION
TEST OD PULSE DISTRIBUTION	These two neons represent the status of two FF's used to form test-OD pulses from a 400-kc oscillator; when operated, the START TESTER switch turns both neons off; the 400-kc oscillator output is applied to four gate tubes conditioned by these FF's; each gate tube is conditioned successively to produce the OD pulse.
TEST DD WORD CTRL	Turned on by the first test-interleave pulse after TEST DD INITIAL DELAY neon is turned off; when on, gates an OD 1 pulse to the read gates and sends the test-DD-loading pulse to the DD generator; same OD 1 pulse turns the neon off.
TEST DD INITIAL DELAY	Turned on by the START TESTER switch; remains on during the initial DD generator delay period; gates test interleave pulses which occur every 640 microseconds to become test-index pulses; start-test-DD-read pulse from the DD generator turns this neon off.

TABLE 1-119. DISPLAY TESTER NEONS (cont'd)

INDICATOR NEON	INDICATION
TEST TD WD SYNC	Turned on by the end-cycle pulse from the SD generator; when on, the next test-OD-3 pulse is gated to turn this neon off and to turn on TEST SD WD CTRL neon; this insures proper timing for reading in the test word.
END TEST CONTROL	Turned on when the START TESTER switch is operated; when on, gates the output of the 400-kc oscillator to the OD pulse distributor where the OD pulses are formed; turned off at the end of the test.
TEST WORD SEQ	These three neons represent the status of three FF's used to read out the eight test words sequentially, one every 10 microseconds.
TEST SD WD CTRL	Turned on by an OD 3 pulse when TEST DD INITIAL DELAY neon is on; when TEST SD WD CTRL neon goes on, it gates an OD 4 pulse to the SD generator as a word-on-way pulse and then gates the subsequent OD 1 pulses to the gate tubes for reading in words 0 through 7.

2.11.5 DIGITAL DISPLAY GENERATOR Neon**Group**

PLAY GENERATOR neon group is given in table 1-120.

The function of each neon in the DIGITAL DIS-

TABLE 1-120. DIGITAL DISPLAY GENERATOR NEONS

INDICATOR NEON	INDICATION
CONTRAST GATE GEN	Turned on by an OD 1 pulse once every 640 microseconds; next OD 1 pulse turns neon off; this contrast gate of 10-microsecond duration is applied to all OD tubes to improve the contrast between the used and unused portions of the tube face.
CHARACTER TIMING & INTENSIFY	These six neons represent the status of each bit in a counter used to produce a 90-microsecond intensification gate every 120 microseconds; counter is stepped by OD 1 pulses when ON-OFF CTRL neon is on; when in the display phase, the fourth pulse is gated to turn INTENSIFY GATE GEN neon on; this is the start of the intensity gate which ends when the 13th OD 1 pulse comes in and turns INTENSIFY GATE GEN neon off and clears the counter; in addition to the above function, the 13th pulse turns the last neon on; next OD 3 pulse clears the character register; then, when the 14th OD 1 pulse comes in, it turns last neon off and steps the X position counter and shift control.
ERASE GATE COUNTER	These two neons represent the status of the bits in a scale-of-4 counter used to count end-word pulses from the X position counter and shift control; fourth such pulse is used to turn off ERASE GATE GEN neon and to stop the erase gate.
PHASE CTR	These two neons represent the status of the bits in a two-stage counter used to establish the various timing phases used in the DD generator; initiate-DD pulse turns the second neon on; an OD 1 pulse is then gated to turn on the first neon; with both neons on, drum-index pulses are gated to the delay counter, fourth index pulse turns the second neon off, thus initiating the display phase; first neon is turned off by the stop-DD-display pulse to end the cycle.

TABLE 1-120. DIGITAL DISPLAY GENERATOR NEONS (cont'd)

INDICATOR NEON	INDICATION
DELAY CTR	These four neons represent the status of each bit in a scale-of-8 counter used to count MIXD drum-index pulses; the eighth index pulse clears the counter and becomes the start-DD-read pulse.
SLOT COUNTER	These seven neons represent the status of each bit in a scale-of-128 counter used to select the particular DD tube for display; counter is stepped each time an 01 control bit combination appears at the start of a new slot.
ON-OFF CTRL	Turned off by initiate-DD pulse or end-word pulse; when off, it gates the first drum-index pulse to erase gate as the initial-erase pulse; this same index pulse also turns neon on, thereby gating OD 1 pulses to step the character timing and intensify counter.
CONTRAST COUNTER	These six neons represent status of each bit in a scale-of-64 counter used to produce a 10-microsecond contrast gate every 640 microseconds; this improves the contrast between used and unused portions of DD tube face; counter is stepped by OD pulses; the 64th OD 1 pulse turns on TEST TD WD SYNC neon and generates the contrast gate; the 65th OD 1 pulse turns CONTRAST GATE GEN neon off.
CONTROL BIT STORE	These two neons store the sign bits of the DD word; the LS and RS bits are controls bits, used as follows: LS = 0, RS = 1, advance slot counter by 1 to select a new DD tube; LS = 1, RS = 0, word displayed top line, opposite side of tube; LS = 1 RS = 1, word displayed two lines below previous word; LS = 0, RS = 0, word displayed directly beneath previous word.
INTENSIFY GATE GEN	In the display phase, the fourth OD 1 pulse to the character timing and intensify counter turns this neon on; this produces the intensify gate for unblanking DD tube; when 12th OD 1 pulse comes in 90 microseconds later, it turns this neon off and drops the intensify gate.
ERASE GATE GEN	Turned on by the sense-erase pulse when ERASE CONTROL neon is on; this action generates the erase gate for erasing old information on DD tube face; turned off by the fourth end-word pulse from erase gate counter.
ERASE CONTROL	Turned on by initial-erase pulse during delay phase of DD cycle and thereafter by an 01 control bit arrangement each time a new DD tube is selected; a sense-erase pulse from the X position counter and shift control turns neon off, resets erase counter, and turns erase gate FF on.
CHARACTER COUNTING & POSITIONING	The first neon is turned on when display is to occur on left half of the tube face and off for right half; the next three neons establish position of the characters in the X direction; last four neons comprise a scale-of-16 counter for positioning the words in Y direction of tube face.
CHARACTER REG	First three neons represent binary x co-ordinates used for character selection; last three neons represent binary y co-ordinates of the character; the H character is displayed first; it is immediately stored here (bits L13 through L15 and R13 through R15).
LEFT CHARACTER STORAGE L1-L12	These 12 neons reflect the status of L1 and L12 bits of DD word when it first comes in; these bits represent x co-ordinates for selection of characters I, J, K, and L.
RIGHT CHARACTER STORAGE R1-R12	These 12 neons reflect status of R1 and R12 bits of DD word when it first comes in; these bits represent y co-ordinates for selection of characters I, J, K, and L.

2.11.6 CAMERA CONTROL Neon Group

The function of each neon in the CAMERA CONTROL neon group is given in table 1-121.

TABLE 1-121. CAMERA CONTROL NEONS

INDICATOR NEON	INDICATION
REQUEST OPERATION	Turned on at OT 9 by the <i>Operate</i> (31) or (32) instruction; gates a TP 5 pulse which performs a number of operations as described above and which turns this neon off.
INTERLOCK	Turned on by power reset or by TP 1 if a film indexing operation has occurred; it gates a TP 5 pulse to clear the synchronizer FF and, if no operation is requested, it clears the SD camera sense FF; if an <i>Operate</i> (31) or (32) instruction has set the request operation FF, the TP 5 pulse will be gated to turn neon off and set mode selected FF to 1 or 0 side, depending on which mode, 1 or 2, respectively, was selected.
MODE REQUESTED	Turned on at OT of <i>Operate</i> (31) instruction and turned off at OT 9 of <i>Operate</i> (32) instruction; this gates a TP 5 pulse to set mode-selected FF to appropriate status, 1 or 0
MODE SELECTED	Turned on by manually selecting camera mode 1 or by a TP5 pulse after an <i>Operate</i> (31) instruction turns on MODE REQUESTED flip-flop and neon; turned off for mode 2 <i>Operate</i> (32) instruction.
SYNC	Turned off each time camera film is indexed; when off, a TP 1 pulse is gated to turn INTERLOCK neon on. A TP 5 pulse is subsequently gated by interlock FF to turn SYNC neon off.
INTS	Turned on by first start-TD or start-RD pulse, whichever occurs first; remains on until one complete cycle of TD and RD drum reading has occurred; turned off as described under CAMERA CONTROL COUNTER neon and the film is indexed.
COUNTER	Turned on by the second start-TD or start-RD pulse; these two pulses follow each other in a repetitive manner; if this neon is turned on by a start-TD pulse, it gates next start-RD pulse to turn itself off and also to turn off the intensify FF and neon, thus indexing the camera film.

2.12 WARNING LIGHTS INDICATORS

The WARNING LIGHTS indicator neons (located below the SITUATION DISPLAY indicator neons) reflect the status of the warning light registers, eight being used at present, with three as spares. (See fig. 1-216.) Every neon indicator indicates the status of a particular bit position. An illuminated neon indicates that the bit position contains a 1. If the neon is not illuminated, this indicates that the bit position contains a 0. Each row of neons represents the contents of a specific WARNING LIGHT register. In order to be able to read the contents of these registers with comparative ease, the neons are grouped in series of threes

so the octonary equivalent can be read directly. These registers are used primarily during testing since, during normal operations, their indications change too rapidly to be perceived visually. The function of each bit is dependent upon the patchboard wiring and the computer program.

2.13 MANUAL INPUTS INDICATORS

The MANUAL INPUTS indicator neons (located below the TAPES section of the tape element and manual data input indicator panel) indicate various operations of the manual data input element. (See fig. 1-217.) The function of each neon is given in table 1-122.

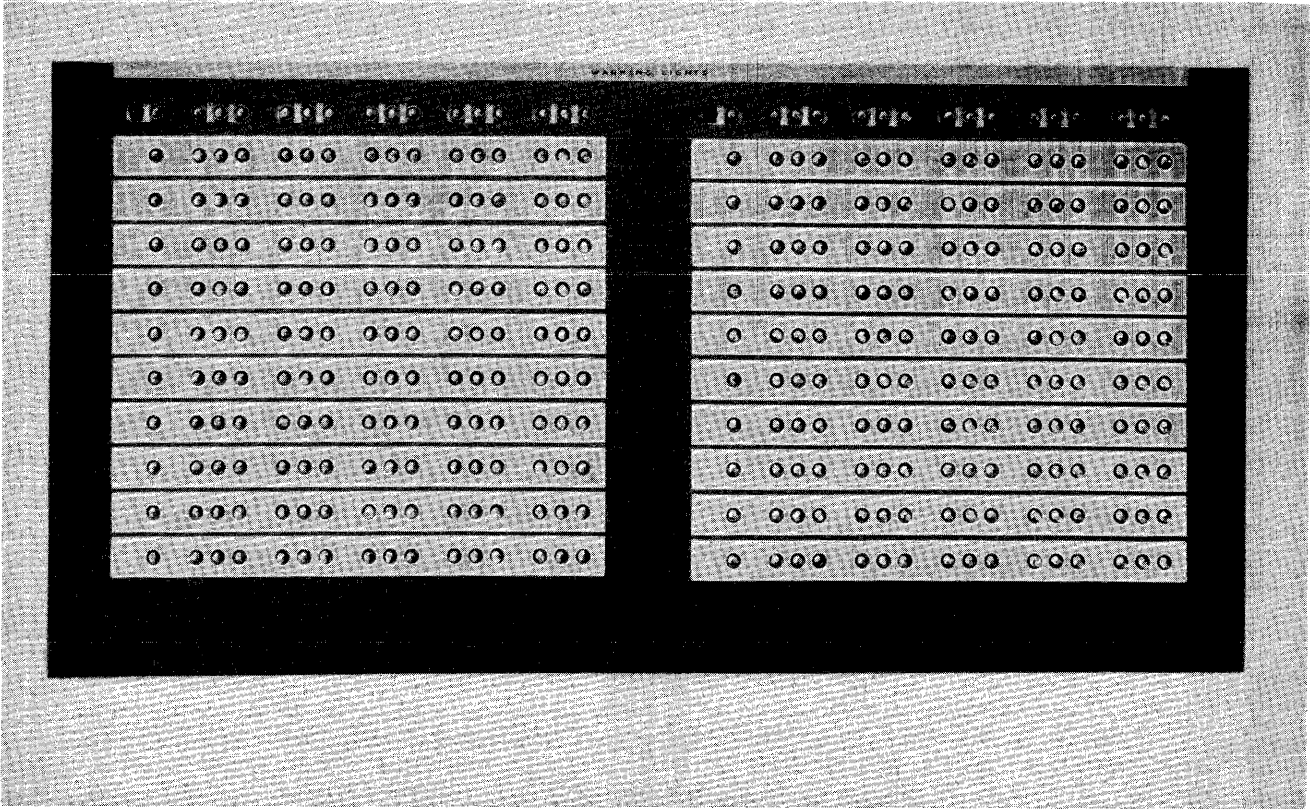


Figure 1-216. Warning Lights Register Neons

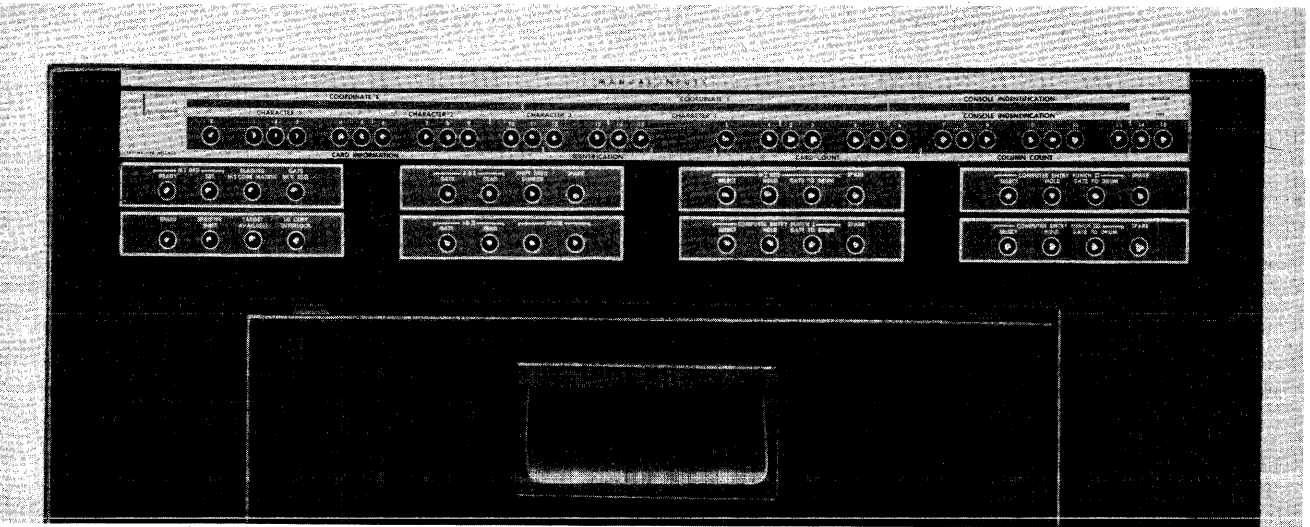


Figure 1-217. Manual Data Input Neons

TABLE 1-122. MANUAL INPUTS NEONS

INDICATOR NEON	INDICATION
MANUAL INPUT REGISTER	These 32 neons represent the status of each bit in the MI message; this register stores messages from light guns and area discriminators prior to being transferred to the MI OD write register for entry on drums.

TABLE 1-122. MANUAL INPUTS NEONS (cont'd)

INDICATOR NEON	INDICATION
MI REG READY	Turned on by the conditional-unblank pulse when SET neon is off; when on, indicates that the MI register is empty and hence ready to receive light gun information; neon is turned off by the MI-register-set pulse.
MI REG SET	Turned on by the MI-register-set pulse and turned off by the MI-register-reset pulse; when off, it gates the conditional-unblank pulse to turn MI REG READY neon on.
READING MI CORE MATRIX	Turned on at the end of the load-shift-register pulse; remains on until core matrix reading is completed and the disconnect pulse comes in to turn this neon off; when on, it gates OD 1 pulses to the shift frequency divider FF.
GATE BRK REQ	Turned on by an OD 1 pulse when the shift frequency divider FF is on; gate break request FF then gates an OD 3 pulse to request a break during MI core readout.
REGISTER SHIFT	Turned on by the gated OD 1 pulse from the shift frequency divider FF; turned off by an OD 2 pulse, thus producing the 2.5-microsecond shift pulse used in reading out the core array.
TARGET AVAILABLE	Turned on when LG CORE INTERLOCK neon is turned off (see below); when on, it enables the light gun to send information to the MI register; neon is turned off by the condition-sample pulse 7.5 microseconds after the conditional-unblank pulse.
LG CORE INTERLOCK	Turned on when reading out the core matrix; turned off when reading is completed; when off, it gates conditional-unblank pulses to turn on the TARGET AVAILABLE neon.
AD-I GATE	Turned on by the start-TD pulse; while on, it enables ADI information to be sent to the MI register; neon is turned off by the next start-TD pulse.
AD-I READ	Turned on by the read-area-discriminator-I pulse; the next start-TD pulse turns on ADI GATE neon; start-RD pulse follows and turns the READ neon off.
SHIFT FREQ DIVIDER	Turned on by an OD pulse when READING MI CORE MATRIX is on; it then gates an OD 1 pulse to the register shift FF and to the gate break request FF; the latter FF gates an OD 3 pulse to turn the SHIFT FREQ DIVIDER neon off.
AD-II GATE	Turned on by the start-RD pulse; while on, it enables AD II information to be sent to the MI register; neon is turned off by the next start-RD pulse.
AD-II READ	Turned on by the read-area-discriminator-II pulse; the next start-RD pulse turns on AD II GATE neon; start-TD pulse follows and turns this READ neon off.
MI REG SELECT	Turned on by a drum demand pulse at OD 3 if the GATE TO DRUM neon is on; when on, the information from the MI register is transferred to the MI OD write register for drum entry.
HOLD	Turned on by an MI-register-information-available pulse; remains on until a drum-demand pulse turns the SELECT neon on and the HOLD neon off.
GATE TO DRUM	Turned on by an OD 1 pulse if the HOLD neon is on; when on, a drum-demand pulse is gated to turn on the SELECT neon and to turn off both the HOLD neon and itself.

TABLE 1-122. MANUAL INPUTS NEONS (cont'd)

INDICATOR NEON	INDICATION
COMPUTER ENTRY PUNCH I SELECT	Same as SELECT (MI REG) neon but for computer entry punch I.
HOLD	Same as HOLD (MI REG) neon but for computer entry punch I.
GATE TO DRUM	Same as GATE TO DRUM (MI REG) neon but for computer entry punch I.
COMPUTER ENTRY PUNCH II SELECT	Same as SELECT (MI REG) neon but for computer entry punch II.
HOLD	Same as HOLD (MI REG) neon but for computer entry punch II.
GATE TO DRUM	Same as GATE TO DRUM (MI REG) neon but for computer entry punch II.
COMPUTER ENTRY PUNCH III SELECT	Same as SELECT (MI REG) neon but for computer entry punch III.
HOLD	Same as HOLD (MI REG) neon but for computer entry punch III.
GATE TO DRUM	Same as GATE TO DRUM (MI REG) neon but for computer entry punch III.

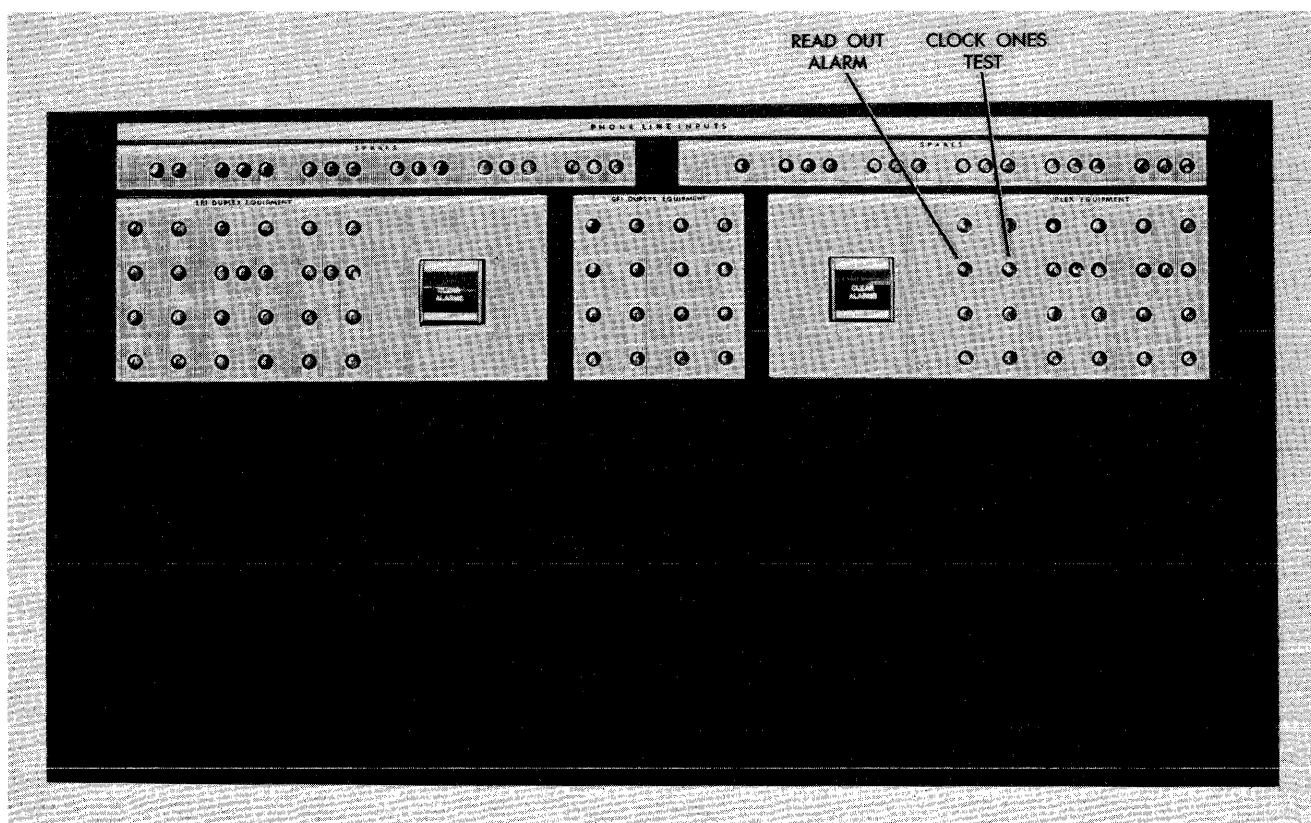


Figure 1-218. Common Input System Controls and Indicators

2.14 PHONE LINE INPUTS INDICATORS AND CONTROLS

2.14.1 General

The PHONE LINE INPUTS neon indicators and controls (located below the OUTPUT SYSTEM neon

indicators on the output and common input indicator panel) indicate various operations of the LRI, GFI, and XTL common input equipment. (See fig. 1-218.) The reason the LRI, GFI, and XTL common input equipment is duplexed and the remainder of the LRI, GFI,

and XTL elements are simplexed is that the failure of the LRI, GFI, or XTL common input sections, which are common to their respective channel input sections, would disable the entire LRI, GFI, or XTL input element. The functions of each neon and control in the LRI, GFI, and XTL duplex equipment are described in the following text.

2.14.2 LRI DUPLEX EQUIPMENT Neons and Control

The CLEAR ALARMS pushbutton switch is the only control in the LRI DUPLEX EQUIPMENT group. This switch operates a pulse generator which resets all alarm flip-flops in the LRI section. The function of each neon in the LRI DUPLEX EQUIPMENT group is given in table 1-123.

TABLE 1-123. LRI DUPLEX EQUIPMENT NEONS

INDICATOR NEON	INDICATION
READOUT ALARM	Turned on by an LRI 2 pulse if new data comes in on the channel before data in core array is read out.
CLOCK ALARM	Turned on by an OD 1 pulse in the event a clock error has occurred, as determined by a sync pulse from the real-time clock.
CLOCK PARITY 1-0	These two neons indicate the parity of the clock reading at any given time; odd parity is indicated when the 1 neon is on and even parity is indicated when the 0 neon is on.
CLOCK RESET CTRL	Turned on once every 16 seconds by sync pulse from real-time clock; gates another pulse from the real-time clock occurring at a 4-pps rate which turns neon off and samples alarm gate for clock error.
CLOCK SYNC	Comes on each time a pulse from the real-time clock comes in; these pulses are sent at a 4-pps rate; an OD 3 pulse is gated to set the clock stepping FF; this FF gates an OD 1 + 1.5 pulse to step the clock and to turn neon off.
SIDE TIME READOUT	Turned on by an OD 1 pulse to cause the site information to be read out in first 26-bit word; after second word is read, an OD 4 pulse turns neon off.
CLOCK TIME 1, 2, 3, 4, 5, 6	These six neons indicate status of each bit in a scale-of-64 counter, stepped at a 4-pps rate, which serves as the LRI clock.
CLOCK STEPPING	Turned on by an OD 3 pulse after CLOCK SYNC (LRI DUPLEX EQUIPMENT) is on; gates an OD 1 + 1.5 pulse to step the clock and to turn neon off.
PULSE GEN, 1-0	These two neons are complemented back and forth between 1 and 0 by an OD 4 pulse which causes generation of LRI timing pulses; on the 0 side, LRI 2 and LRI 4 are generated by OD 4 pulses.
LRI 1/2 LEVEL CHAN 1-18, 1-0	The 1 neon is turned on by an LRI 1 pulse, turned off by an LRI 2 pulse to produce the core-shifting pulse for channels 1-18; the 0 neon is turned on when the 1 neon goes off.
MONITOR	Turned on to suppress OD timing pulses while the monitor is in use; it is ordinarily off.
DRUM DEMAND, 1-2	The drum demand 1 neon is turned on by an OD 3 drum-demand pulse from LRI 1 field while the 2 neon is set by the corresponding LRI 2 field pulse; if both pulses come in simultaneously, preference is given to LRI 1 for writing on drums.
LRI 1/2 LEVEL CHAN 19-36, 1-0	Same as LRI 1/2 LEVEL CHAN 1-18 neons but for channels 19 through 36.

2.14.3 GFI DUPLEX EQUIPMENT Neons

There are only two active neons in the GFI DUPLEX EQUIPMENT group. The function of each neon is given in table 1-124.

2.14.4 XTL DUPLEX EQUIPMENT Neons and Control

The CLEAR ALARMS pushbutton switch is the only control on the XTL DUPLEX EQUIPMENT group. This switch operates a pulse generator which resets all alarm flip-flops in the XTL section. The function of each neon in the XTL DUPLEX EQUIPMENT group is given in table 1-125.

TABLE 1-124. GFI DUPLEX EQUIPMENT NEONS

INDICATOR NEON	INDICATION
DRUM DEMAND, 1-0	The 1 neon is turned on at OD 1 by the drum-demand pulse and turned off at OD 2, at which time the 0 neon goes on.
OD-2 FREQ CONV 1	Complemented on and off by an OD 2 pulse; every other OD 2 pulse is gated to form core-shift pulses for reading out of the array.

TABLE 1-125. XTL DUPLEX EQUIPMENT NEONS

INDICATOR NEON	INDICATION
CLOCK ALARM	Turned on by an OD 1 pulse in the event a clock error has occurred, as determined by sync pulse from the real-time clock.
CLOCK 1	Comes on each time a pulse from real-time clock comes in; these pulses are sent at a 4-pps rate; an OD 3 pulse is gated to set clock stepping FF; this FF gates an OD 1 pulse to step clock and turn neon off.
CLOCK 0	Turned on by an OD 3 pulse after CLOCK 1 neon is on; an OD 1 pulse is then gated to step the clock and turn this neon and CLOCK 1 neon off.
CLOCK PARITY, 1-0	These two neons indicate parity of the clock reading at any given time; odd parity is indicated when the 1 neon is on and even parity is indicated when the 0 neon is on.
CLOCK RESET CTRL	Turned on every 16 seconds by a pulse from the real-time clock; this gates another pulse from the real-time clock which comes in at a 4-pps rate; this pulse turns neon off and samples alarm gate to check for clock errors.
READOUT ALARM	Turned on if new data comes in on the channel before data already in the channel is transferred to drum.
CLOCK TIME 1, 2, 3, 4, 5, 6	These six neons indicate status of each bit in a scale-of-64 counter which is stepped at a 4-pps rate to serve as the XTL clock.
XTL PULSE GENERATOR	Complemented by an OD 3 pulse to cause generation of XTL timing and level pulses.
XTL 2/3 LEVEL	Turned on by every other OD 2 pulse and turned off at OD 3 to form a 2.5-microsecond pulse for switching, etc.
XTL 5/6 LEVEL	Turned on by every other OD 1 pulse and turned off at OD 2 to form a 2.5-microsecond pulse for switching, etc.

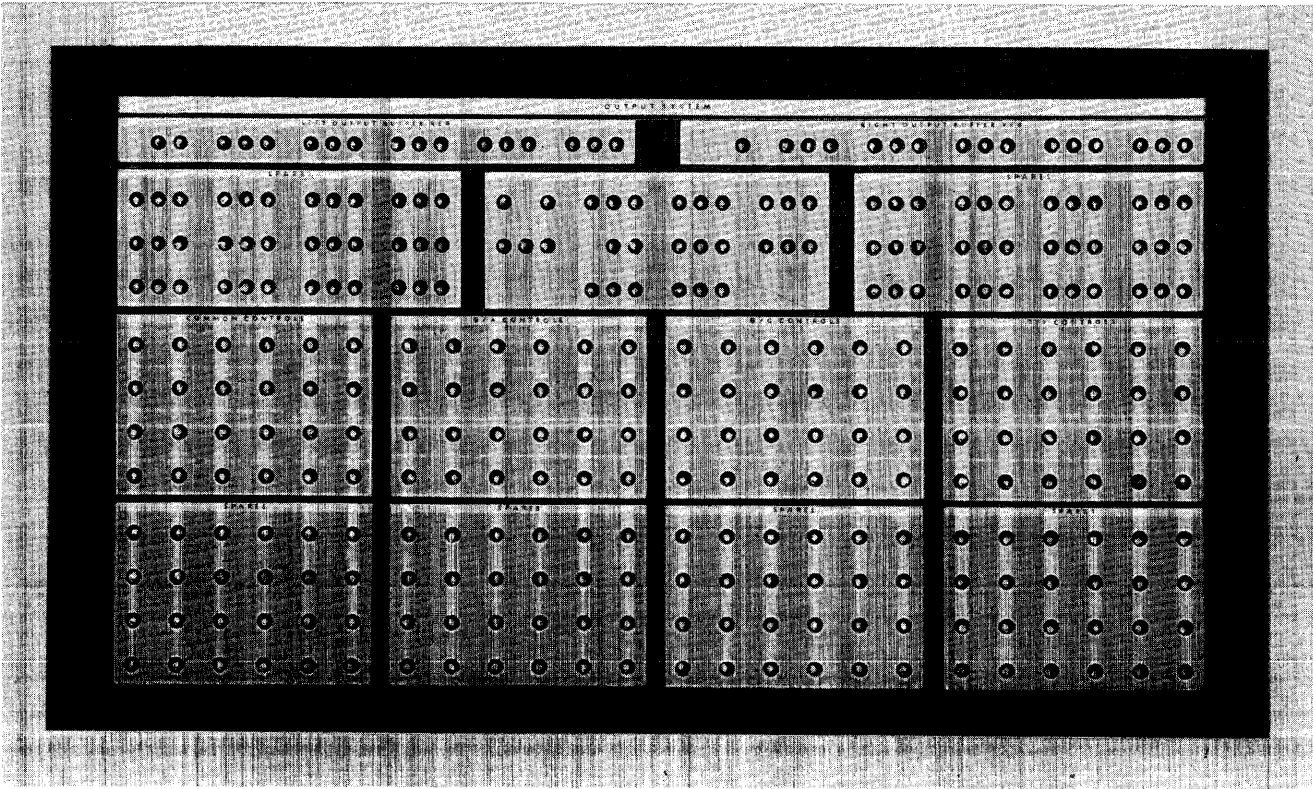


Figure 1-219. Output System Indicators

2.15 OUTPUT SYSTEM INDICATORS

2.15.1 General

The OUTPUT SYSTEM neon indicators (located above the PHONE LINE INPUTS indicators and controls on the output and common input system indicator panel) indicate various operations of the Output System. (See fig. 1-219.) These operations start with the transfer of the right and left half-word read from the OB fields to the LEFT and RIGHT OUTPUT BUFFER REG. These 32 neons represent the status of each bit in the left and right half-word read from the OB fields. An illuminated neon indicates that the bit position contains a 1. If the neon is not illuminated, the bit position contains a 0.

The counter group of neons (located in the center of the first row of neon blocks), the COMMON CON-

TROLS neons, the G/A CONTROLS neons, and the TTY CONTROLS neons indicate various stages of processing of output information by the Output System. This covers the output control element accepting drum words, performing certain checks on them, routing the output information to the selected G/A, G/G, or TTY storage section, converting the output information, and supplying it to the telephone transmission equipment. The register neons, counter neons, and other neons are used primarily during testing since, during normal operation, their indications change too rapidly to be perceived visually. The function of each neon is described in the following text.

2.15.2 Counter and Output Parity Generator Neons Group

The function of each neon in the counter and output parity generator group is given in table 1-126.

TABLE 1-126. COUNTER AND OUTPUT PARITY GENERATOR NEONS

INDICATOR NEON	INDICATION
OUTPUT PARITY GEN	Turned on whenever parity count of the drum right half-word is odd; this new parity bit makes the parity count even for word written in the array; turned off by output-buffer-register-clear pulse at OD 4 + 0.5.

TABLE 1-126. COUNTER AND OUTPUT PARITY GENERATOR NEONS (cont'd)

INDICATOR NEON	INDICATION
G/A ELAPSED TIME CTR	These four neons represent status of each bit in a scale-of-16 counter which is cleared by OD 4-13 pulse at 25 ctr = 1 and then stepped at a 32-pps rate; computer can thus determine the elapsed time since beginning of transmission of present burst.
G/A BURST COUNTER	These six neons represent status of each bit in a scale-of-64 counter which is stepped by an OD 4-13 when 25 ctr = 19; thus, counter is stepped once each burst time.
BURST COUNT SELECT CTR	These three neons represent status of each bit in a scale-of-8 counter used to read out burst counters and elapsed time counter; counter is stepped by OD 1 pulses which also read various burst counters as selected by this counter; elapsed time counter and G/A burst counter are selected by 000, G/G by 001, and TTY by 010; cleared by an OD 1 pulse after the reading is completed.
G/G BURST COUNTER	These eight neons represent status of each bit in a scale-of-256 counter which is stepped by an OD 4-13 pulse at 19 ctr = 1; thus, counter is stepped once each burst time.
TTY BURST COUNTER	These six neons represent status of each bit in scale-of-64 counter which is stepped by OD 3 pulses at 51 ctr = 45; thus, counter is stepped once for each burst time.

2.15.3 COMMON CONTROLS Neon Group

The function of each neon in the COMMON CONTROLS group is given in table 1-127.

TABLE 1-127. COMMON CONTROLS NEONS

INDICATOR NEON	INDICATION
OUTPUT BUFFER LOAD	Turned on at OD 1 by output-buffer-loading pulse; at OD 1+2, test-parity pulse is applied to parity gates to perform parity count on OB drum word in output buffer register; turned off at OD 2.
SET PULSE	If parity of drum word is satisfactory, neon turned on at OD 2+2 and turned off at OD 4; this produces the gating pulse to set drivers that feed half-write current generator matrix used for core array read-in.
INHIBIT G/A	Turned on at OD 2 and off at OD 4+0.5 if G/A has not been selected for writing; if G/A is selected for writing, turned on at OD 4+0.5 and off at OD 1+1.1.
G/G-A	Same as G/A (INHIBIT) neon but for G/G-A core array.
G/G-B	Same as G/A (INHIBIT) neon but for G/G-B core array.
TTY	Same as G/A (INHIBIT) neon but for TTY core array.
BURST TIME CTR	Turned on by OD 3 pulse after BURST TIME CTR SEL & READ neon (COMMON CONTROLS) is on; gates OD 1 pulses to step select counters and read burst time counters; turned off by OD 3 after BURST TIME CTR SEL & READ neon (COMMON CONTROLS) is off.
BURST TIME CTR SEL & READ	Turned on by read-burst-time-counters pulse at PT 6; remains on during reading of burst counters and turned off by disconnect pulse or by burst drive count select ctr=8.

TABLE 1-127. COMMON CONTROLS NEONS (cont'd)

INDICATOR NEON	INDICATION
OD 91 FREQ DIV	These two neons represent status of each bit in a scale-of-4 counter used to count the pulses from a 364-pps oscillator; every fourth pulse clears the counter and is sent out to give 91 pps for TTY timing circuits.
RESET & PRIME SYNC	Turned on when manual reset switch is operated; turned off by an OD 4 pulse.
RESET & PRIME	Turned on by first OD 2 pulse after RESET & PRIME SYNC neon is on; gates an OD 4 pulse to reset and prime the 19, 13, 5, 25, and 51 counters; this OD 4 pulse also turns off this neon and RESET & PRIME SYNC neon.
OD 91 PULSE START	Operates the same as OD 13 PULSE STOP neon but at a 91-pps rate.
OD 91 PULSE STOP	Operates the same as OD 13 PULSE STOP neon but at a 91-pps rate to form OD 2-91, OD 3-91, and OD 4-91 pulses.
OD 13 PULSE START	Turned on by pulses from a 1,300-pps oscillator; the first OD 2-13 pulse which is gated turns this neon off.
OD 13 PULSE STOP	When OD 13 PULSE START neon is on, an OD 4+0.5 pulse is gated to turn this neon on; when on, OD 2, OD 3, and OD 4 pulses are gated; the OD 4 gated pulse turns this neon off; since it turned on at a 1,300-pps rate, drum pulses which are gated become OD 2-12, OD 3-13, and OD 4-13 pulses.
OD 32 PULSE START	Turned on by pulses from real-time clock at a rate of 32 pps; when on, an OD 4+0.5 pulse is gated to turn (G/G CONTROLS) G/G OUTPUT SYNC 5 neon on; this in turn gates an OD 2 pulse to step the elapsed time counter and turn this neon off.
OD 32 PULSE STOP	When OD 32 PULSE START neon is turned on, an OD 4+0.5 pulse is gated to turn this neon on; when on, an OD 2 pulse is gated to step elapsed time counter and to turn off this neon and OD 32 PULSE START neon.

2.15.4 G/A CONTROLS Neon Group

The function of each neon in the G/A CONTROLS group is given in table 1-128.

TABLE 1-128. G/A CONTROLS NEONS

INDICATOR NEON	INDICATION
SHIFT CONTROL	Turned on by an OD 4-13 when 25 ctr=1; turned off by an OD 4-13 when the 25 ctr=19; when on, OD 2-13 pulses are gated as shift pulses to output shift register shift pulse generator.
OUTPUT SHIFT	Turned on by OD 2-13 pulses; turned off by OD 3-13 pulses to produce output-shift-register-shift pulses.
AUTO BUSY BIT	Turned on once each second by an OD 4-13 when 25 ctr=1 and turned off by an OD 4-13 at 25 ctr=2; this enables automatic busy bits to be sent out once each second even if array is not being read out.
SEARCH	Set on by an OD 4-13 at 25 ctr=19; set off by an OD 4-13 at 25 ctr=25 and 13 ctr=7; when on, indicates information is being read into the G/A section from the OB fields.

TABLE 1-128. G/A CONTROLS NEONS (cont'd)

INDICATOR NEON	INDICATION
SKIP PULSE	Turned on by OD 4-13 at 25 ctr=25; the next OD 2-13 pulse turns neon off; this introduces one extra OD 2-13 timing pulse to make a total of 326 timing pulses for each G/A burst period.
25 COUNTER CARRY	Turned on by OD 4-13 at 25 ctr=25; an OD 3-13 will then be gated, normally at 13 ctr=11, to produce the G/A sync pulse; this sync pulse turns neon off; in G/A looped to LRI, the sync pulse is gated at 13 ctr=12.
25 COUNTER SHIFT	Turned on by an OD 3-13 pulse at 13 ctr=13; turned off by OD 4-13 pulses to produce the 25-ctr-shift pulse; at 25 ctr=25, neon stays on 10 microseconds to prime the 13 ctr.
13 CTR SHIFT	Set on by an OD 2-13 pulse, set off by an OD 3 pulse to form 13-ctr-shift pulses.
G/A OUTPUT SYNC	Same as G/G output sync 1, 2, 3, 4, 5 neons (G/G CONTROL) but for G/A section.

2.15.5 G/G CONTROLS Neon Group

The function of each neon in the G/G CONTROLS group is given in table 1-129.

TABLE 1-129. G/G CONTROLS NEONS

INDICATOR NEON	INDICATION
COM MSG CSR SHIFT GEN	Turned on by an OD 3 shift-CMSR pulse, turned off at OD 3+0.5; this produces the CMSR-shift pulse if one of the registers in a slot has been addressed.
COMPLETED MSG CONTROL CHAN 0-4, 5-9, 10-14, 15-19, 20-24	Turned on by an OD 1 pulse at 19 ctr=19 if all five registers in the slot have been addressed during G/G read-in; this enables the message to be sent out on the telephone lines; turned off by OD 4 at ctr=18.
OUTPUT SHIFT	Turned on by an OD 2-13 pulse and turned off by an OD 3-13 pulse to form the output-shift-register-shift pulses.
SHIFT CONTROLS	Turned on by OD 4-13 at 19 ctr=1 to supply shift pulses to output shift register shift pulse generator; turned off by an OD 4-13 at 19 ctr=18 to stop shifting after core array is empty.
19 COUNTER CARRY	Turned on by an OD 4-13 pulse at 19 ctr=19; an OD 3-13 pulse is then gated to form the G/G sync pulse and this same OD 3-13 pulse turns neon off.
SHIFT	Turned on by an OD 3-13 pulse at ctr=5 and off by an OD 4-13 pulse to produce 19-ctr-shift pulse; at 19 ctr=19, remains on 10 microseconds to clear the 19 counter and CMSR and to prime the 19 counter, 5 counter, and CMSR.
PHASE	Complemented by an OD 4-13 pulse occurring at ctr=18; when on, it indicates G/G-B is being read out; when off, it indicates G/G-A is being read out.
SEARCH	Turned on by an OD 4-13 pulse when 19 ctr=1; remains on during the search period until turned off by an OD 4-13 pulse at 19 ctr=18, indicating the end of search time.
5 CTR SHIFT	Turned on by OD 2-13 and cleared at OD 3-13 to form the 5-ctr-shift pulse.
G/G OUTPUT SYNC 1, 2, 3, 4, 5	These five neons are turned on once each burst time by the G/G sync pulse at OD 3-13; it gates output of the 1,300-cps timing oscillator to telephone lines; turned off by the next OD 2-13 pulse after passing one cycle of outputs to telephone lines.

2.15.6 TTY CONTROLS Neon Group

The function of each neon in the TTY CONTROLS group is given in table 1-130.

TABLE 1-130. TTY CONTROLS NEONS

INDICATOR NEON	INDICATION
SEARCH	Turned on by an OD 3 pulse at 51 ctr=45; turned off by an OD 3 pulse at 51 ctr=51; the OB drum fields are read when neon is on.
51 CTR SHIFT	Normally set on by OD 2-91 pulses and set off by OD 3 pulses to produce the 51-counter-shift pulses; during high-speed shift, turned on by gated OD 2 pulses and off by OD 3 pulses.
OUT SR SHIFT	Normally turned on by OD 3-91 pulses and turned off by next OD 4 pulse to produce the shift-core-shift-register pulses; during high-speed shift, neon is set on by OD 3 pulses and off by OD 4 pulses.
HS SHIFT	Turned on by an OD 3 pulse at 51 ctr=2 to start high-speed shift operation; turned off by an OD 3 pulse at 51 ctr=8 and core shift register is then stepped at the normal rate of 91 pps.
HS SHIFT CONTROL	Turned off by an OD 2-91 pulse; during high-speed shift, it is complemented on and off by OD 2 pulses and, in conjunction with the high-speed shift FF, acts to step the core shift register four times in 120 microseconds, thus eliminating the unwanted parity bit.

2.16 OUTPUT TEST CONTROL PANEL

2.16.1 General

The OUTPUT TEST CONTROL PANEL (located above the output and common input system indicator panel) houses indicator neons, indicator lights, a rotary switch, toggle switches, and pushbutton switches which control and indicate various test operations performed by the Output System. (See fig. 1-220, foldout.) These

controls and indicators are used during the performance of unit loop tests and computer loop tests. The Output System controls and indicators (located on the back of unit No. 42) are used in conjunction with the output test control panel during the execution of a test.

2.16.2 OUTPUT ALARMS Neon Group

The function of each neon indicator in the OUTPUT ALARMS neon group is given in table 1-131.

TABLE 1-131. OUTPUT ALARMS NEONS

INDICATOR NEON	INDICATION
NON SCH COMP	If a word is read into output buffer register with correct burst number for the section to which it is addressed but during a nonsearch period, neon is turned on; an audible alarm sounds to indicate that the word will be lost; search period is the first phase of the burst period; consequently, the array burst number changes, and a no-compare signal is sent when word is read again.
PARITY NG	A parity count is made on each word transferred from the OB drum fields to output buffer register; in this case, the parity count should be odd; if it is even, the neon illuminates and an audible alarm is sounded to indicate a parity error.
ILL ADR	In the G/A core array there are six unassigned register addresses (26 through 31); in the G/G and TTY core arrays there are seven unassigned register addresses (25 through 31); if left half-word should specify any unassigned register, neon goes on and an audible alarm is sounded to indicate an illegal address.

TABLE 1-131. OUTPUT ALARMS NEONS (cont'd)

INDICATOR NEON	INDICATION
ILL SEC	At present, only sections 1, 2, and 3 of the eight output sections are being used; consequently, if the address in left half-word read from the OB drum fields specifies section 0, 4, 5, 6, or 7, neon lights and an audible alarm is sounded to indicate an illegal address.
OUTPUT PARITY	Turned on by an OD 4-13 pulse when 25 ctr=2 if a busy bit parity error occurs or when 25 ctr=19 if an output parity error occurs on either of the two G/A output lines; an audible alarm is sounded; neon remains on until cleared manually.
G/G (OUTPUT PARTY)	Turned on by an OD 4-13 pulse when 19 ctr=19 if a parity error has occurred on any one of the five G/G lines; an audible alarm is sounded and remains on until cleared manually.
TTY (OUTPUT PARTY)	Turned on by stop-search pulse at OD 3 if a parity error has occurred on any one of the 25 TTY lines; an audible alarm is sounded; remains on until cleared manually.
G/A OUTPUT PARITY COUNTERS 1 and 2	If the G/A parity alarm is given, the neon that is on indicates the number of G/A slot in which parity error has occurred; neon is turned off by an OD 4-13 pulse when 25 ctr=19; it is complemented on and off by an OD 4-13 pulse each time a register is addressed during read-in and each time a 1 bit is shifted out of the core shift register during read-out time; if number of busy bits equals number of registers addressed, no alarm is given when 25 ctr=2; if parity count is even, no alarm is given when 25 ctr=19; if the count is odd, neon remains on and the G/A parity alarm is given.
TTY OUTPUT PARITY COUNTERS 1 through 25	If TTY parity alarm is given, neon that is on indicates number of the register or TTY line in which the error has occurred; neon is turned off by an OD 3 pulse when the 51 ctr=51; each time a 1 is shifted out of shift register onto the line, an OD 4 pulse complements this neon on and off; if parity count is even, no alarm is given; if count is odd, neon remains on and the TTY parity alarm is given.
G/G OUTPUT PARITY COUNTERS 1 through 5	If G/G parity alarm is given, neon that is on indicates number of G/G slot in which error has occurred; neon is turned off by OD 3-13 pulse when the 19 counter=19; it is then complemented on and off by an OD 3-13 pulse each time a 1 bit is shifted out of core shift register; if parity count is even, no alarm is given; if parity count is odd, neon remains on and G/G parity alarm is given.

2.16.3 G/A LOOPED TO LRI Controls and Indicators

2.16.3.1 General

The function of each control and indicator in the G/A LOOPED TO LRI controls and indicators group will be discussed individually. The indicator lights that are associated with certain pushbutton switches will be discussed in relationship to them.

2.16.3.2 SELECT G/A Controls and Indicators

The SELECT G/A (0-12) indicator light goes on when the pushbutton switch beneath it is operated. It indicates that the first G/A slot register is feeding test messages to the LRI equipment from the G/A core storage array. The SELECT G/A (13-25) pushbutton and indicator light is used to perform the same function but for G/A slot 2, consisting of registers 13 through 25. The function of the neons is given in table 1-132.

TABLE 1-132. G/A LOOPED TO LRI NEONS

INDICATOR NEON	INDICATION
SCALE OF 4 COUNTER	These two neons represent status of each bit in a binary counter used to count four OD 3-13 pulses with G/A looped to LRI; this counter forms the blank and sync pulses described above; the first OD 3-13 pulse sets counted to 1; the second pulse is then gated by the counter to produce the sync pulse; the fourth pulse clears the counter and the pause FF.
PAUSE	In test status with G/A looped to LRI, message sent to LRI consists of 48 bits preceded by a blank pulse, sync pulse, and two more blank pulses to make 52 bits per message; after first message has been sent, neon is turned on; this suppresses OD 2-13 pulses to the 13 counter shift FF; a blank pulse follows and an OD 3-13 pulse is then gated as the sync pulse; two more blank pulses follow, at which time neon is turned off and the 48 information bits are sent to complete the message.
LRI COMPLETED MESSAGE 1 through 4	Neons 1, 2, 3, and 4 are turned on by OD 4-13 pulses when 25 ctr=4, 8, 12, and 15, respectively; they are turned off by OD 3-13 pulses when 13 ctr=9, 5, 1, and 10, respectively, thus indicating that a complete message of 52 bits has been sent to LRI when G/A is looped to LRI.

2.16.4 SELECT G/G Controls and Indicators

The SELECT G/G controls and indicators are separated into two sections, the G/G LOOPED TO XTL, and the G/G LOOPED TO GFI.

The G/G LOOPED TO XTL indicator lights 1 through 5 are illuminated when pushbutton switches located beneath them are operated. Each indicator light indicates which G/G slot is looped to XTL for test purposes.

The G/G LOOPED to GFI indicator light 1 and 2 or 3 and 4 is illuminated when the pushbutton switch beneath it is operated. Each indicator light indicates that G/G slots 1 and 2 output data or slots 3 and 4 output data are being transferred to the GFI equipment for test purposes.

2.16.5 TTY LOOPED TO TTY RCVR Controls and Indicators

The function of each indicator light (in the TTY LOOPED TO TTY RCVR controls and indicators) that is associated with a pushbutton switch will be discussed in relationship to it. The SELECT TTY indicator light 0, 10, or 20 is illuminated when the pushbutton switch located beneath it is operated. Each indicator light indicates that one of the TTY channels (0 through 9, 10 through 19, or 20 through 25) has been selected and looped to the TTY monitor for test purposes. The rotary switch (0 through 9) operates in conjunction with the SELECT TTY pushbuttons to select one of the 25 TTY channels if the outputs are in test status.

2.16.6 UNIT LOOP TEST Neon Group

The function of each neon indicator in the UNIT LOOP TEST neon group is given in table 1-133.

TABLE 1-133. UNIT LOOP TEST NEONS

INDICATOR NEON	INDICATION
TEST WORD	Turned on by an OD 4-13, a G/G 19 ctr=1, a G/A 25 ctr=25, or by an OD 3 TTY start-search pulse, depending on which section is being tested; the next OD 1 pulse turns neon off and writes the test word in selected array.
TEST TRANSFER DELAY	Turned on by OD 4-13 at 25 ctr=18 for G/A testing or at 19 ctr=17 for G/G testing; it gates an OD 2 pulse to set the test transfer FF; this OD 2 pulse turns the neon off.

TABLE 1-133. UNIT LOOP TEST NEONS (cont'd)

INDICATOR NEON	INDICATION
RESTART TO DRUMS SYNC	Turned on whenever Output System is changed from active to test and vice versa; it gates an OD 3 pulse to turn on; this neon is turned off by an OD 1 pulse as described under RESTART TO DRUM FF neon.
RESTART TO DRUMS FF	Turned on by an OD 3 pulse if RESTART TO DRUMS SYNC neon is on; when on, an OD 1 is gated to restart the OB drum field reading; this same OD 1 pulse turns this neon off and turns RESTART TO DRUMS SYNC neon.
DATA PULSE STRETCHER	Turned on by G/A or G/G data pulses when selected test word bit is shifted out of the array; this occurs at OD 3-13; turned off by an OD 1 pulse and hence is on for 5 microseconds to prime test core shift register with test word bit.
TEST SHIFT CTRL	Turned on by an OD 3-13 pulse each time a bit of selected test word is read out of the array; this gates an OD 1 pulse which sets test shift FF and turns neon off.
SINGLE CYCLE OUTPUTS	Turned on by single-cycle-outputs-start pulse; one test word is read into the selected section, read out, and the cycle stopped; remains on until the start switch is operated.
MASTER STOP	Turned off by start pulse or single-cycle-outputs,start pulse; while off, it indicates that the enable-OD-pulse level is up and that OD 13 and OD 91 pulses are being generated, provided manual pulse control is not in effect; turned on by OD 4-13 at 25 ctr=18 or 19 ctr=17 or by OD 3-13 at 51 ctr=45, depending on section under test and then only if there is an error or if single-cycle outputs is in effect.
TEST TRANSFER	Turned on by an OD 2 pulse gated by test transfer delay FF; an OD 2 pulse turns neon off but it is also gated to test core shift register where it gates the outputs of every core to its appropriate OB register FF as a complementary input.
TEST SHIFT	When test shift control FF is set, an OD 1 pulse turns this neon on; the FF output gates an OD 2 pulse to turn this neon off; in this way, the test-core-shift-register-shift pulse is generated.
OD-13 SINGLE CYCLE	In manual pulse control testing, depressing the MANUAL PULSE switch turns this neon on; this gates an OD 4+0.5 pulse to set OD-13 pulse stop FF on; this gates OD 2, OD 3, and OD 4 pulses to make OD 2-13, OD 3-13, and OD 4-13 pulses; the first OD 2-13 pulse turns neon off; thus, only one cycle of OD 13 pulses results.

2.16.7 OUTPUT Controls and Indicators

The OUTPUT controls and indicators group is composed of indicator lights, toggle switches, and push-

button switches. The function of each indicator light and control switch is given in table 1-134.

TABLE 1-134. OUTPUT CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
OPERATE TEST (toggle)	TEST OUTPUTS	Moving this switch to the TEST position causes the outputs to go to the test status if the computer is in test status; the outputs may be removed from test status by moving the switch to the OPERATE position or by changing the computer to operate status.

TABLE 1-134. OUTPUT CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR	FUNCTION
STOP TO DRUMS-STOP (toggle)		If this switch is on the OPERATE-TEST switch is in the TEST position, a parity error, illegal section, or illegal address stops the drum words from being sent to the output unit and prevents the writing of new information on the OB fields.
G/A-OUTER TEST BUS (toggle)		If the G/A-OUTER TEST BUS switch is in the OUTER TEST BUS position, the output of each G/A channel is fed to the telephone transmission equipment; when in the G/A position, the G/A output data from channel 1 or 2 (depending upon selection of SELECT G/A 0-12 switch or the SELECT G/A 13-25 switch) is channeled into the test loop instead of the telephone transmission equipment.
	UNIT LOOPS ON TEST	This light goes on during unit testing of Output System; unit testing can be over-ridden by computer testing.
RESET FLIP-FLOPS (pushbutton)		This switch operates a pulse generator which resets and clears core counters, core shift registers, and core shift flip-flops.
RESET ALL ALARMS (pushbutton)		This switch operates a pulse generator which resets all output alarm FF's and sends a restart-drum pulse when outputs are in test status.

2.16.8 G/A, TTY G/G OUTPUT DATA Neon Groups

The G/A, TTY, and G/G OUTPUT DATA neons

indicate the status of each bit being transferred out on the G/A channels, TTY channels, or G/G channels. The function of the neons is given in table 1-135.

TABLE 1-135. G/A, TTY, G/G OUTPUT DATA NEONS

INDICATOR NEON	INDICATION
G/A OUTPUT DATA 1 and 2	These neons indicate status of each bit being sent out on G/A channels 1 and 2; if a 1 is sent out, the neon is turned on at OD 3-13 and turned off by the next OD 2-13 pulse.
TTY OUTPUT DATA 1 through 25	Same as G/A OUTPUT DATA 1 and 2 but for TTY; neon is turned on at OD 4-91 and off at OD 3-91.
G/G OUTPUT DATA 1 through 4	Same as G/A OUTPUT DATA 1 and 2 but for G/G.

SECTION 3

SIMPLEX MAINTENANCE CONSOLE

3.1 GENERAL

The simplex maintenance console is divided physically into eight sections, each containing one or more panels. (See fig. 1-221.) These sections, identified for convenience in discussion as follows, include:

- a. Central scope and probes panel
- b. Simplex marginal checking control panel
- c. Power supply control panels
- d. Computer entry punch panel
- e. GFI panels, channels 1 through 18
- f. XTL panels, channels 1 through 12
- g. GFI, XTL, LRI test pattern generator control panel
- h. LRI panels, channels 1 through 36

3.2 CENTRAL SCOPE AND PROBES PANEL

Information not available.

3.3 SIMPLEX MARGINAL CHECKING CONTROL PANEL

The simplex marginal checking controls and indicators are discussed in Section 4.

3.4 POWER SUPPLY CONTROL PANELS

The power supply control panels house indicator lights, toggle switches, and pushbutton switches that control the application and removal of primary power for the simplex power supplies. (See fig. 1-222.) The control of simplex power status is exercised at the duplex switching console. The function of each indicator and control in the POWER SUPPLY controls and indicators group is given in table 1-136. The controls and indicators are duplicated for POWER SUPPLY C and D.

TABLE 1-136. POWER SUPPLY CONTROLS AND INDICATORS

CONTROL	INDICATOR LIGHT	FUNCTION
	PCD UNIT ALARM	Indicates that there has been a failure in the PCD unit; an audible alarm is sounded when the light is illuminated.
	AC SWITCHGEAR UNIT ALARM	This alarm light is illuminated when a failure occurs in the a-c switchgear; an audible alarm is sounded when the light is illuminated.
	AC BACKUP CB UNIT 56	Indicates any open backup CB for the ac which feeds LRI, GFI, or XTL; there are two a-c back CB's for LRI, each feeding half of the channels, one for XTL and one for GFI.
	DC BACKUP OR -48 V CB UNIT 56	Indicates that any of the backup CB's which feed dc for LRI, XTL, or GFI are open; one group of d-c backup CB's feeds half the channels of LRI and one group feeds the other half, one for XTL, one for GFI; also indicates an open CB in the -48 V line which feeds the interlock relays, the -48 V line which feeds the relay power for the simplex switch, or the -48 V line which feeds the status relays.
	AC BACKUP CB UNIT 48	Indicates an open a-c backup for displays; there is an a-c backup CB for every 20 displays.
	DC BACKUP OR -48 V CB UNIT 48	Indicates that any of the d-c backup CB's which feed the displays are open; there is a group of CB's for every 20 displays; it also indicates an open CB which feeds the interlock relays for unit 48, or the -48 V CB which feeds the status relays for unit 48.
	VOLTAGE OFF LIMITS ALARM	Indicates that one of the warning voltages which are on the output of the d-c supplies has gone off limits; these relays are set to closer limits than the monitor which shuts down the power; an audible alarm is sounded when the light is illuminated.

TABLE 1-136. POWER SUPPLY CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR LIGHT	FUNCTION
POWER ON (pushbutton)		When energized, causes the power to be cycled up on the No. 1 generator, then the ac-on signal is sent to the simplex power switch causing the a-c contactors to be closed on the simplex equipment connected to that supply; filament voltage generator is then cycled up; when the filaments are at full voltage, the dc-on signal is sent through the simplex power switch and the contactors are closed, applying dc to the simplex units connected to that supply.
	POWER ON	Indicates that all power is applied to the simplex units connected to the supply.
POWER OFF (pushbutton)		Removes power from the simplex units in sequence, cycles down the d-c supplies and the generators; the motor-generator (MG) sets keep running.
	POWER OFF	Indicates that the power supply is in the power off status; the only unit of the power supply that may be energized in this status is the motor of the motor-generator set.
AC ONLY (pushbutton)		Performs all the steps of the POWER ON switch except that it does not close the d-c contactors to the simplex units.
	AC ONLY	Indicates that the power supply is in the a-c only status; the MG sets are cycled up and the ac is applied to the filaments of the simplex units which are connected to the supply; the d-c supplies are cycled up but the d-c contactors are not closed.
	ACTIVE	Indicates that the supply is functioning as the active simplex supply.
	STANDBY	Indicates that the supply is functioning as the standby simplex supply.
	-48 V UNIT 48	This light does the same for unit 48 as the -48 V unit 56 light does for unit 56.
	-48 V UNIT 56	Indicates that -48 V is present on unit 56 and that the indication CB for unit 56 is closed; audible alarm is sounded if this light is turned off by action of the indication CB; this gives a check on the indication CB.
RESET AUDIBLE ALARM (pushbutton)		Turns off the audible alarm caused by any of the alarm circuits on the power control panel; audible alarm circuit is automatically restored to active status when the trouble is cleared.
	MASTER READY	Indicates that all power has been applied to the simplex units connected to the supply and that a time delay has been allowed to stabilize components.
SERVICE OPPOSITE SIMPLEX SWITCH (lever switch)		Switch is located on panel C; when in its operated position, removes all power going from the C supply to the D side of the simplex power switch panel; switch on D panel removes the power going to the C side of the simplex power switch panel; it is used when servicing the simplex power switch connectors.

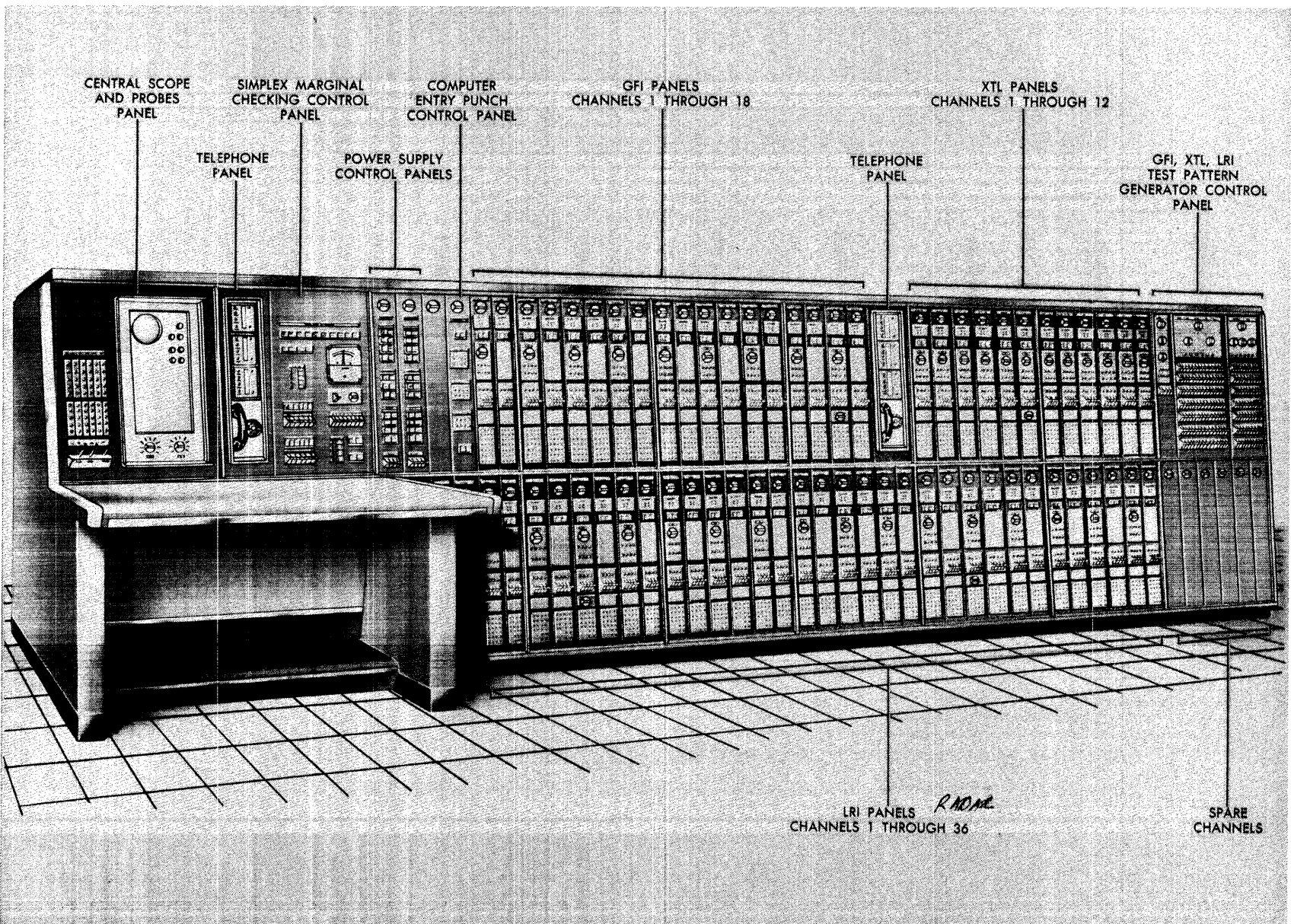


Figure 1-221. Simplex Maintenance Console

UNCLASSIFIED

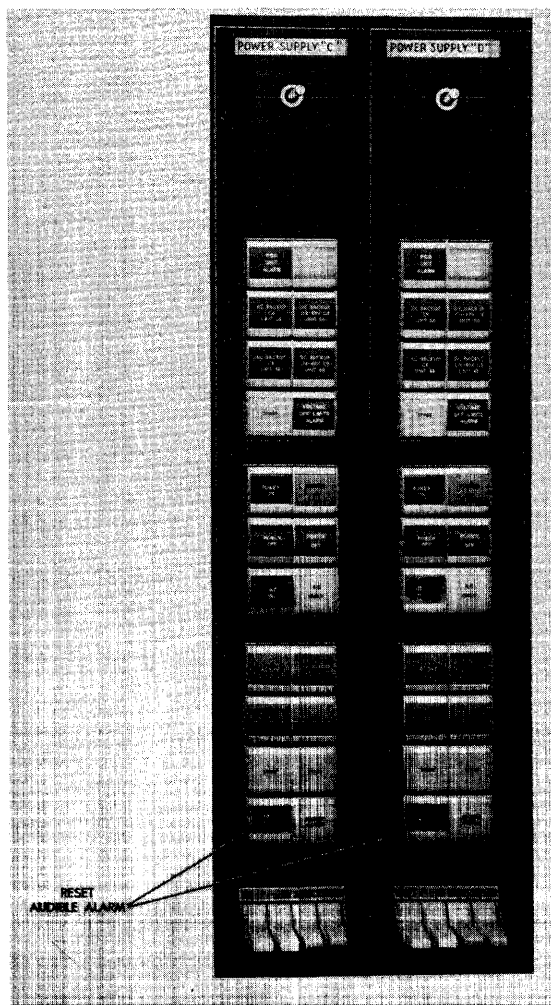


Figure 1-222. Power Supply Controls and Indicators

3.5 COMPUTER ENTRY PUNCH PANEL

3.5.1 General

The computer entry punch panel houses indicator lights and pushbutton controls for the air conditioning system and the computer entry punches. (See fig. 1-223.)

3.5.2 Air Conditioning Controls

The two air conditioning pushbutton switches are the RESET AIR COND AUD ALARM and the AIR COND TROUBLE ACKNOW. The RESET AIR COND AUD ALARM pushbutton is used to turn off the air conditioning audible alarm located in the maintenance room if the alarm has also been reset in the air conditioning control room. The AIR COND TROUBLE ACKNOW pushbutton switch, when operated, prevents

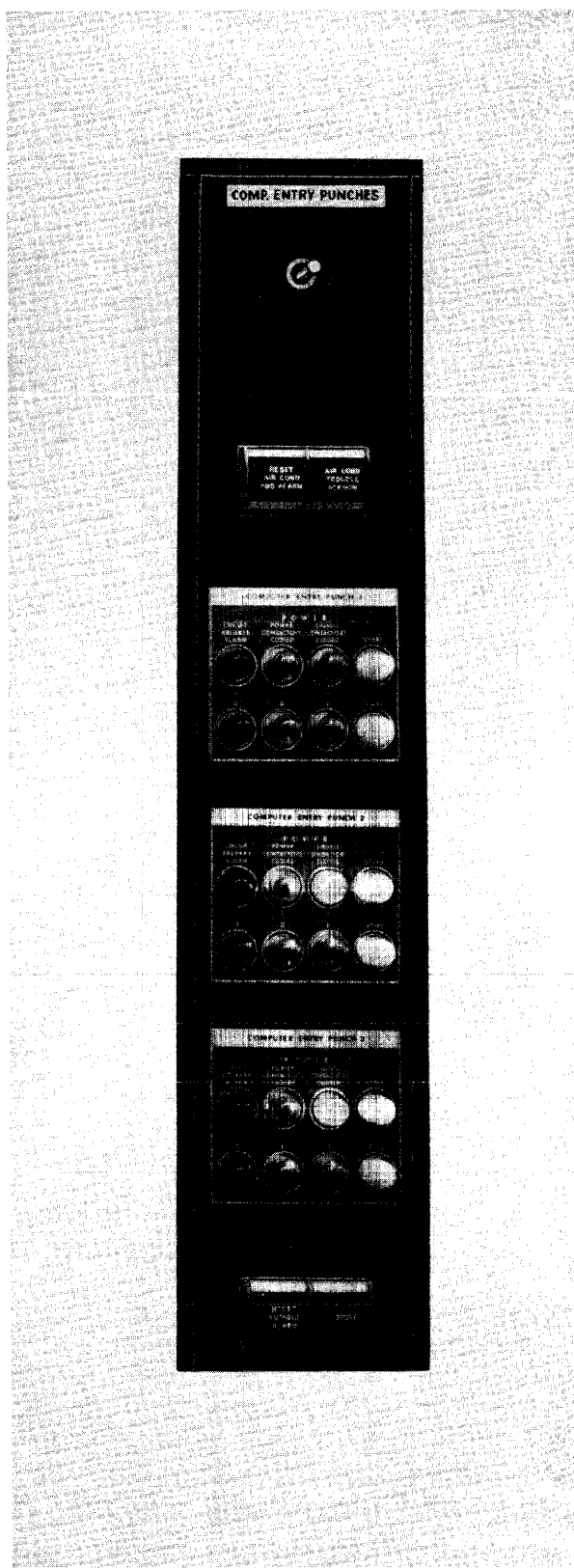


Figure 1-223. Computer Entry Punch Controls and Indicators

an automatic shutdown of the equipment caused by air conditioning trouble; the operator is indicating that he knows about the trouble but must keep the system operating for a limited time.

3.5.3 Computer Entry Punches Indicators

The function of each indicator light is given in table 1-137. There are three computer entry punches; the indicator lights are duplicated for each punch.

The RESET AUDIBLE ALARM pushbutton mounted near the bottom of the panel is used to reset the audible alarm on the computer entry punch panel; the audible alarm circuit is automatically restored to the active status when the trouble is cleared.

3.6 GFI CHANNEL CONTROL PANELS

3.6.1 General

Each GFI channel control panel houses indicators and controls for one GFI channel input section. (See fig. 1-224.) Since GFI channel input sections are paired, two to a module, alternate GFI channel panels contain controls which affect both channels in that pair. The panel housing the controls for spare channels 15 and 16 also contains a channel selector switch which allows

selection of a pair of channels to be replaced by spare channels 15 and 16. Only four panels are shown in figure 1-224. The knob at the top of each panel has no logical function; it controls a bolt which holds the panel in place on the simplex maintenance console.

3.6.2 GFI ALARMS Controls and Indicators

The function of the indicator light and pushbutton switch in the GFI ALARMS controls and indicators group is given in table 1-138.

3.6.3 GFI Unit Status Rotary Switch

The unit status switch is a six-position rotary switch which is used to control the status of each pair of GFI channels. It operates in conjunction with the duplex switch circuits to connect the pair of channels to the proper computer and with the simplex power switch circuits to connect the pair of channels to the proper power supply. There is one unit status switch per pair of channels and it is mounted on the odd-numbered channel. The unit status switch cannot be rotated unless the locking lever is in the UNLOCK position. The six positions for the unit status switch are given in table 1-139.

TABLE 1-137. COMPUTER ENTRY PUNCH CONTROLS INDICATORS

INDICATOR LIGHT	FUNCTION
CIRCUIT BREAKER ALARM C AND D	Indicates that a circuit breaker that supplies power to a particular punch is opened for any reason; an audible alarm is sounded at the same time that the light is turned on.
POWER CONTACTORS CLOSED C AND D	Indicates which power supply is powering a particular punch; the proper supply is selected by the unit status switch and the simplex power switch on the duplex switching console.
SIGNAL CONTACTORS CLOSED C AND D	Shows which computer is connected to a particular punch; the proper computer is selected by the unit status switch and the duplex switch.

TABLE 1-138. GFI ALARMS CONTROLS AND INDICATORS

CONTROL	INDICATOR LIGHT	FUNCTION
	EXCESS TARGET	Operated by a relay which is operated by an analog counter; if filtered targets appear at a rate which exceeds a certain predetermined rate, the alarm is operated; an audible alarm is sounded at the simplex maintenance console on the error channel when the excess target neon is lighted.
CLEAR ALARMS (pushbutton)		There is one CLEAR ALARMS pushbutton per channel; when operated, it resets the audible alarm on the channel where the switch is located; audible alarm circuit is automatically made active when the trouble is cleared.

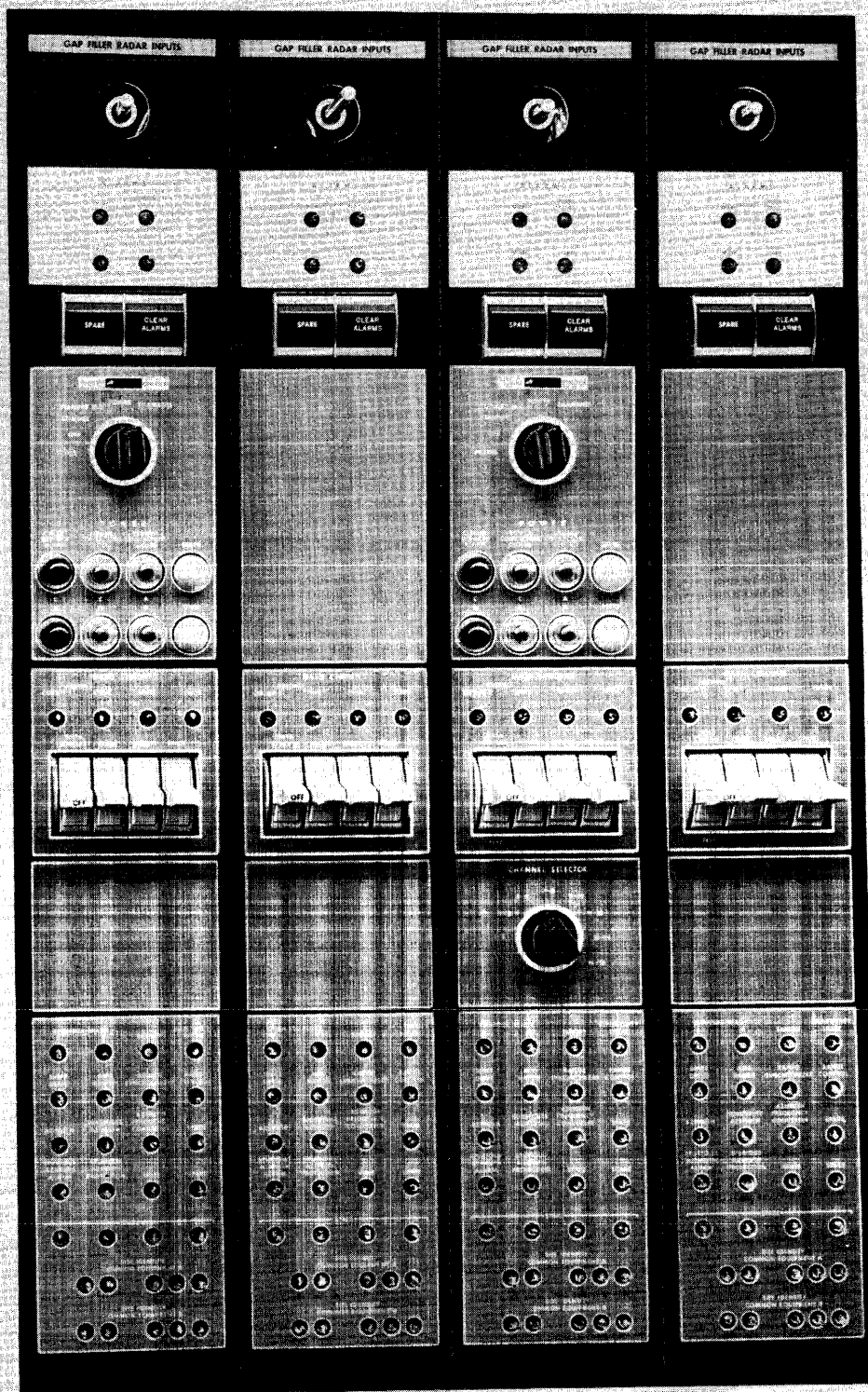


Figure 1-224. GFI Channel Controls and Indicators

TABLE 1-139. GFI UNIT STATUS ROTARY SWITCH POSITIONS

POSITION	FUNCTION
ACTIVE	The active power supply and the active computers are connected to the pair of channels; no marginal checking of the pair of channels is possible.
OFF	The channels are not connected to either computer or to either power supply.
STANDBY	The channels are connected to the standby computer and the standby power supply; no marginal checking of the pair of channels is possible.
STANDBY M.C.	The status is the same as standby except that the pair of channels may be marginal checked.
POWER	The pair of channels is connected to the standby power supply but it is not connected to either computer.
FILAMENTS	The pair of channels is connected to the standby power supply for ac but the d-c contactors are not closed.

TABLE 1-140. GFI POWER INDICATORS

INDICATOR	INDICATION
CIRCUIT BREAKER ALARM C AND D	Lights are present only on the odd-numbered GFI channels and, when lighted, indicate that a CB has opened in the lines which supply the pair of channels on which the light appears; there is one light for C and one for D power supply; an audible alarm is sounded when either of these light comes on.
POWER CONTACTORS CLOSED C and D	Lights are present only on the odd-numbered GFI channels and indicate which power supply is powering the pair of channels on which the light appears.
SIGNAL CONTACTORS CLOSED A and B	Lights are mounted only on the odd-numbered GFI channels and indicate which computer is receiving the signals from the pair of channels on which the light appears.

3.6.4 GFI POWER Indicators

The functions of the indicator lights in the GFI POWER indicators group are given in table 1-140.

3.6.5 GFI DATA CIRCUITS Controls and Indicators

The function of the indicator neons and lever switches in the GFI DATA CIRCUITS controls and indicators group are given in table 1-141.

3.6.6 CHANNEL SELECTOR Rotary Switch

There is one CHANNEL SELECTOR rotary switch mounted on the odd-numbered GFI spare channel. It is used to select the pair of channels which the pair of spare channels replace. To place the spare channels in operation, the following procedure is to be followed:

1. Remove from ACTIVE status the pair of channels which are to be replaced by the spare channels.

2. Check that the DATA CIRCUIT switches on the spare channels are OFF.
3. Using the CHANNEL SELECTOR switch, select the pair of channels which are being replaced.
4. Switch the spare channels to ACTIVE status.
5. Switch the spare channel DATA CIRCUIT switches to the DATA CIRCUIT position.

3.6.7 GFI Channel Input Section Neons

The function of each of the neons in the GFI channel input section group is given in table 1-142.

3.7 XTLC CHANNEL CONTROL PANELS

3.7.1 General

Each XTLC channel control panel houses indicators and controls for one of the 12 XTLC channel input sections. (See fig. 1-225.) The panel housing the controls

TABLE 1-141. GFI DATA CIRCUITS CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
	ABSENCE CIRCUIT I	On the data circuit, is not transmitting data; the signal comes from telephone equipment.
DATA CIRCUIT-OFF-TEST (lever switch)		The DATA CIRCUIT-OFF-TEST switch is used to control the source of the information which is fed into each GFI channel; it is a three-position lever switch; when it is in the DATA CIRCUIT position, the information is coming to the channel over the normal telephone line; when the switch is in the TEST position, the channel is connected to the GFI test bus; test bus may be transmitting information from a G/G output loop or from a pattern generator; when the switch is in the OFF position, all data circuits to the channel are disconnected.

TABLE 1-142. GFI INPUT SECTION NEONS

INDICATOR NEON	INDICATION
CHANNEL READY 1-0	Indicates status of flip-flop which prevents the generation of a data-available pulse until there is target data in the parallel read-out register; neons should be complementing if the channel is operating correctly.
TARGET	Turned on by mapped target pulse; cleared by range pulse.
REINTENSIFY DELAY	Indicates status of flip-flop used to control the reintensifying of filtered targets and the counting of range and azimuth pulses.
SHIFT DRIVE	On after every range pulse; indicated application of nine shift pulses to the azimuth and range counters, causing the range and azimuth pulses to be counted.
ADD RANGE	Indicates generation of a 5-microsecond pulse for every synchronized range pulse to step the range counter.
RANGE SYNCHRONIZER	Turned on when a range pulse occurs to give a synchronized range pulse.
RANGE CONTROL	On when range control flip-flop converts range pulse to level which sets the range synchronizer.
SUBSTITUTE AZIMUTH	On when a substitute azimuth pulse is generated if the normal azimuth pulse has not been received before a certain expected time.
AZIMUTH	On when azimuth pulse is synchronized.
AZIMUTH FREQUENCY CONVERTER	On for alternate incoming azimuth pulses.
SPURIOUS AZIMUTH ALARM	Turned on by spurious azimuth pulses gated through spurious azimuth protection; the output goes to an analog counter which operates an alarm in the mapper console if the spurious azimuth pulses occur faster than at a predetermined rate.
SPURIOUS AZIMUTH PROTECTION	Turned on by an incoming azimuth pulse and cleared by a pulse from azimuth protection just before the normal azimuth pulse is expected; any spurious azimuth pulses which occur while this neon is on are gated to spurious azimuth alarm.
SITE IDENTIFY COMMON EQUIPMENT A and B	Energized by five lines from the common equipment; neons give a visual indication to the operator of the radar site which is associated with a particular channel.

for the spare channel also contains a channel selector switch which allows selection of a channel to be replaced by the spare channel. Only two panels are shown in figure 1-225. The knob at the top of each panel has

no logical function; it controls a bolt which holds the panel in place on the simplex maintenance console.

3.7.2 XTL ALARMS Controls and Indicators

The functions of the indicator light and pushbut-

TABLE 1-143. XTL ALARMS CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
	PARITY ALARM	Turned on if any one of the parity checking neons indicates an error in parity (odd count); the same pulse which sets the parity alarm neon in the channel equipment sets a general alarm neon in the common equipment.
CLEAR ALARMS		There is one CLEAR ALARMS pushbutton switch per XTL channel; when operated, it resets the audible alarm on the channel where the switch is located; it also resets the parity alarm neon in the channel equipment; the audible alarm circuit is automatically made active when the trouble is cleared.

TABLE 1-144. XTL UNIT STATUS ROTARY SWITCH POSITIONS

POSITION	FUNCTION
ACTIVE	The active power supply and the active computer are connected to the channel; no marginal checking of the channel is possible.
OFF	The channel is not connected to either computer or either power supply.
STANDBY	The channel is connected to the standby computer and the standby power supply; no marginal checking of the channel is possible.
STANDBY M.C.	The status is the same as standby except that the channel may be marginal checked.
POWER	The channel is connected to the standby power supply but is not connected to either computer.
FILAMENTS	The channel is connected to the standby power supply for ac but the d-c contactors are not closed.

TABLE 1-145. XTL POWER INDICATORS

INDICATOR	FUNCTION
CIRCUIT BREAKER ALARM C and D	These alarm lights are present on all XTL channels; when lighted, indicate that a CB has opened in one of the lines supplying power to the channel on which the alarm appears; there is one light for the C supply and one light for the D supply; an audible alarm is sounded when either of these lights comes on.
POWER CONTACTORS CLOSED C and D	These lights are present on all XTL channels and indicate which power supply is furnishing power for the channel on which the light appears.
SIGNAL CONTACTORS CLOSED A and B	These lights are present on all XTL channels and indicate which computer is receiving the signals from the channel on which the light appears.

ton switch in the XTL ALARMS controls and indicators group are given in table 1-143.

3.7.3 XTL Unit Status Rotary Switch

The unit status switch is a 6-position rotary switch which is used to control the status of each XTL channel. It functions in conjunction with the simplex power switch to connect the XTL channel to the proper power supply and functions in conjunction with the duplex switch to connect the XTL channel to the proper computer. There is one unit status switch for each XTL channel. The unit status switch does not operate if the locking lever is in the UNLOCK position. The six positions for the unit status switch are shown in table 1-144.

INFORMATION NOT AVAILABLE

3.7.4 XTL POWER Indicators

The functions of the indicator lights in the XTL POWER indicators group are given in table 1-145.

3.7.5 XTL DATA CIRCUITS Controls and Indicators

The functions of the indicator neons and lever switches in the XTL DATA CIRCUITS controls and indicators group are given in table 1-146.

Figure 1-225. XTL Channel Controls and Indicators

TABLE 1-146. XTL DATA CIRCUITS CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
	ABSENCE CIRCUIT 1	Indicates that DATA CIRCUIT 1 is not transmitting data; signal comes from telephone equipment; there is one neon per XTL channel.
	ABSENCE CIRCUIT 2	Indicates that DATA CIRCUIT 2 is not transmitting data; signal comes from telephone equipment; there is one neon per XTL channel.
	SELECTIVE CIRCUIT 1	Illuminated when the XTL channel is receiving its data from DATA CIRCUIT 1.
	SELECTIVE CIRCUIT 2	Illuminated when channel is receiving data from Data CIRCUIT 2.
DATA CIRCUIT-OFF-TEST		There is one DATA CIRCUIT-OFF-TEST switch per XTL channel; it is used to control the source of the data which is fed into each channel; when the switch is in the DATA CIRCUIT position, the data is coming to the channel over the telephone line; when it is in the TEST position, the data enters the channel from the XTL test bus; when the switch is OFF, all data to the channel is cut off.
DATA CIRCUIT		Used to select one of two available telephone lines which feed data to an XTL channel; the switch can select DATA CIRCUIT 1, DATA CIRCUIT 2, or AUTO; when it is in the AUTO position, a circuit is operable, which automatically selects the good line.

TABLE 1-146. XTL DATA CIRCUITS CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR LIGHT	FUNCTION
PARITY DISABLED		The PARITY DISABLED switch is a two-position lever switch per channel; the OFF position functions normally with an incorrect parity; the word is discarded; with the PARITY DISABLED position and the channel in standby operation, an incorrect parity can be placed in the drums for test purposes; the parity alarm neon functions in both cases.

3.7.6 Channel Selector Rotary Switch

The channel selector rotary switch is mounted only on the spare channel. It is used to select the XTL channel which the spare channel replaces. To place the spare channel in operation, the following procedure is to be followed:

1. Remove from ACTIVE status the channel which is to be replaced.
2. Check that the DATA CIRCUIT switch on the spare channel is OFF.
3. Using the channel selector switch, select the channel which is to be replaced by the spare channel.
4. Switch the spare channel to ACTIVE status.
5. Switch the spare channel DATA CIRCUIT switch to the DATA CIRCUIT position.

TABLE 1-147. XTL CHANNEL INPUT SECTION NEONS

INDICATOR NEON	INDICATION
CHANNEL READY 1-0	Neon 1 is on at the end of the fast shift, indicating that a 5-word message is ready to be read to the drums; if a second fast shift occurs before the previous message has been read out, the alarm neon in the common equipment is set.
GOOD MESSAGE	On if received message has correct parity and address.
FAST SHIFT CONTROL	Indicates that the 7-core shift registers and the 17-core shift registers are fast-shifting after the message is completed to clear the 7-core shift registers for the next message.
PARITY CHECKING 1, 2, 3, 4, 5	These neons each indicate the parity count on one word of the incoming message; since the neons are cleared at the beginning of a message, if a neon is on when the message ends, the message is discarded.
TIMING	On when the core shift for the core shift registers and the data distributor cores is initiated.
TIMING SYNC	On for each synced timing pulse.
START READOUT & RESET	Turned on if the channel ready FF is on when a drum-demand pulse comes from the common equipment.
READOUT & RESET 1 and 2	These neons indicate when the start readout and reset FF's come on.
ADDRESS COMPARE	Indicates result of incoming message address comparison.
READOUT PROTECTION	Indicates status of flip-flop used to prevent excessive duty cycle on the core shift drivers.
SHIFT COUPLER	On when the 17-core shift registers are being shifted in step with the 7-core shift registers after 7 shifts have taken place.
SHIFT DELAY	Indicates status of flip-flop which delays the shift pulse to the CSR's and the core counter from XTL 3/6 time to XTL 5/6 time.
SITE IDENTITY COMMON EQUIPMENT A and B	These neons are energized by five lines from site identity cans in the common equipment to give a visual indication to the operator as to the site associated with a particular channel.

3.7.7 XTL Channel Input Section Neons

The function of each of the neons in the XTL channel input section group is given in table 1-147.

3.8 LRI CHANNEL CONTROL PANELS

3.8.1 General

Each LRI channel control panel houses indicators and controls for one LRI channel input section. (See fig. 1-226.) Since LRI channel input sections are paired, two to a module, alternate channel panels contain controls which affect both channels in that pair. The panels housing the controls for spare channels 9 and 10 and 27 and 28 also contain a channel selector switch which allows selection of a pair of channels to be replaced by the spare channels. Four panels are shown in figure 1-226. The knob at the top of each panel has no logical function; it controls a bolt which holds the panel in place on the simplex maintenance console.

3.8.2 LRI ALARMS Controls and Indicators

The functions of the indicator light and pushbutton switch in the LRI ALARMS controls and indicators group are given in table 1-148.

3.8.3 LRI Unit Status Rotary Switch

There is one unit status rotary switch per pair of LRI channels. It operates in conjunction with the simplex power switch to connect the pair of channels with which the switch is associated to the correct power supply; it also operates in conjunction with the duplex switch to connect the pair of channels to the correct computer. The unit status switch cannot be rotated unless the locking lever is in the UNLOCK position. The six positions for the unit status switch are shown in table 1-149.

3.8.4 LRI Power Indicators

The functions of the indicator lights in the LRI power indicators group are given in table 1-150.

TABLE 1-148. LRI ALARMS CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
CLEAR ALARMS	PARITY ALARM	Turned on by the parity count neon when the parity count comes up odd.
		Located on every LRI channel; resets the audible alarm; it also resets the parity alarm neon in the channel equipment; the alarm circuit is automatically reset when the trouble is cleared.

TABLE 1-149. LRI UNIT STATUS ROTARY SWITCH POSITIONS

POSITION	FUNCTION
ACTIVE	Power comes from active power supply and signals go to active computer.
OFF	All power and signal lines are open.
STANDBY	Power comes from standby supply and signals go to standby computer.
STANDBY M.C.	Power comes from standby supply and signals go to standby computer; channels may be marginal checked.
POWER	Power comes from the standby supply but no signal lines are connected.
FILAMENTS	A-c power comes from the standby supply but the d-c contactors are not closed.

TABLE 1-150. LRI POWER INDICATORS

INDICATOR	FUNCTION
CIRCUIT BREAKER ALARM C and D	Indicates that a CB has opened in the power lines for a particular channel pair.
POWER CONTACTORS CLOSED C and D	Indicates which power supply is connected to each channel; there is one C and one D light per channel pair.
SIGNAL CONTACTORS CLOSED A and B	Indicates which computer is receiving the signals from each channel of LRI; there is one A and one B light per channel pair.

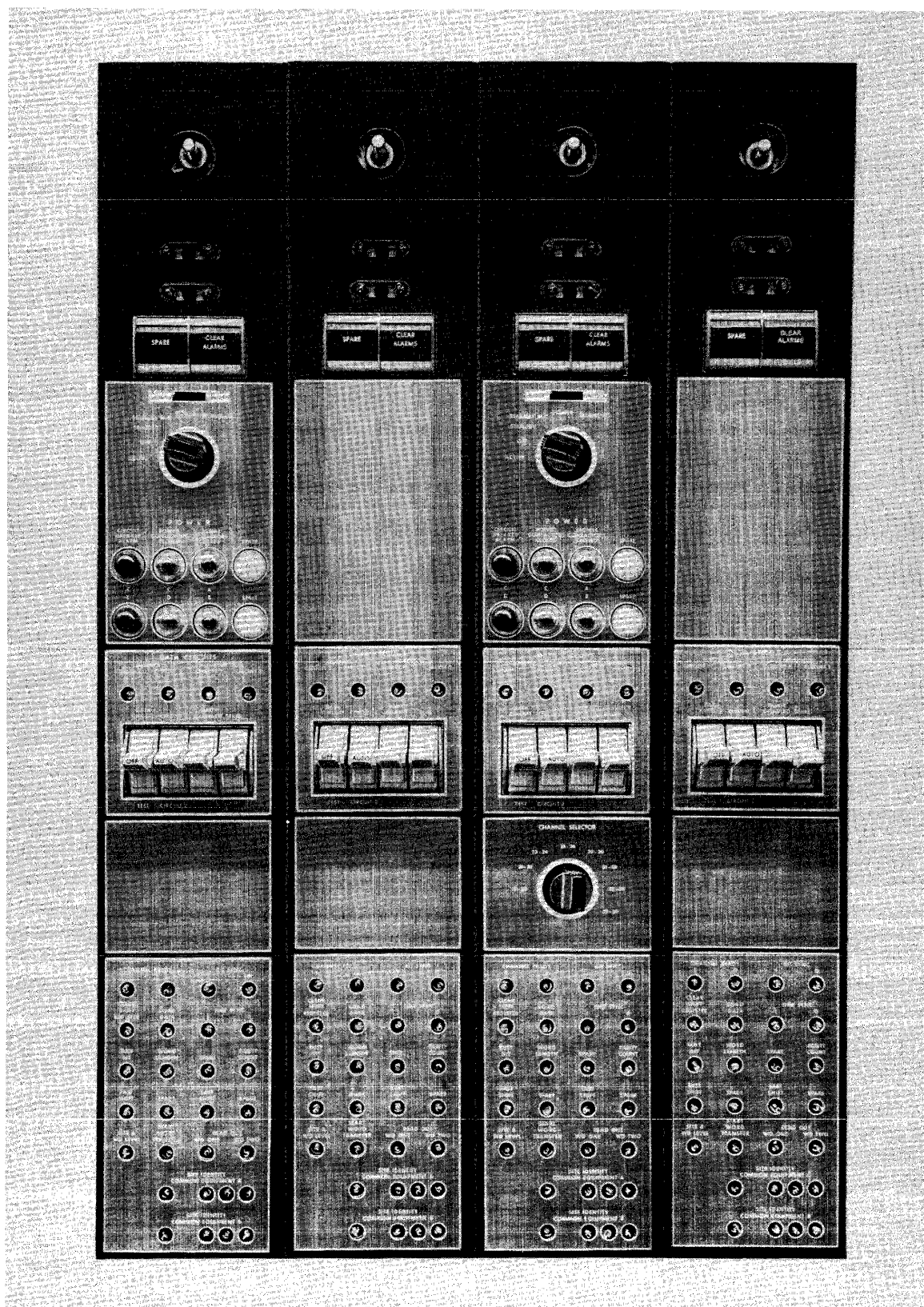


Figure 1-226. LRI Channel Controls and Indicators

3.8.5 LRI DATA CIRCUITS Controls and Indicators

The functions of the indicator neons and lever switches in the LRI DATA CIRCUITS controls and indicators group are given in table 1-151.

3.8.6 Channel Selector Rotary Switch

The channel selector rotary switches are mounted on odd-numbered spare channels 9 and 27; there are two pairs of spare channels in the LRI unit: 9 and 10, and 27 and 28. It is possible to connect each pair of spare channels for only half of the LRI channels. The procedure to be followed in substituting a spare for a regular channel is as follows:

1. Remove the pair of regular channels from the ACTIVE status.
2. Check that the DATA CIRCUIT switches on the pair of spare channels are set OFF.
3. Select the pair of regular channels which are to be replaced by the spares on the channel selector switch.

4. Place the pair of spare channels on ACTIVE status.

5. Switch the DATA CIRCUIT switch to DATA CIRCUIT position.

3.8.7 LRI Channel Input Section Neons

The function of each of the neons in the LRI channel input section group is given in table 1-152.

3.9 GFI, XTL, LRI TEST PATTERN GENERATOR CONTROL PANEL**3.9.1 General**

The test pattern generators provide facilities for testing the GFI, LRI, and XTL channel input sections. The test data is set up manually and applied by *Operate* instructions, using the standby computer. Facilities are provided in the test pattern generator for output-to-input test loops. The test loops include the standby G/A output storage section to LRI and the standby G/G output storage section to XTL.

The pattern generators have switching relays located in each section to switch the pattern generator to

TABLE 1-151. LRI DATA CIRCUITS CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
	ABSENCE CIRCUIT 1 and 2	These neons are controlled from telephone equipment and indicate when a data circuit is not transmitting data; the line is actually sensed for presence of data.
	SELECTION CIRCUIT 1 and 2	These neons indicate which data circuit is feeding data to the channel on which the lights are counted.
DATA CIRCUIT- OFF-TEST		The DATA CIRCUIT-OFF-TEST lever switch is used to control the source of data for each LRI channel; there is one switch per channel; the switch may be OFF, thus cutting OFF, thus cutting off all data to the channel; it may be in the TEST position, which connects the channel to the LRI test bus, or it may be in the DATA CIRCUIT position, which causes data to be fed in through the telephone lines.
PARITY DISABLED		This switch is a two-position lever switch per channel; the OFF position functions normally with an incorrect parity; the word is discarded; with the PARITY DISABLED position and the channel in standby operation, an incorrect parity can be placed in the drums for test purposes; the parity alarm neon functions in both cases.
DATA CIRCUIT		Used to select one of two data circuits for the input to each channel or it may be in the AUTO position, thus activating telephone equipment which automatically selects the good line.

either computer. The switching is controlled so that only the signals from the standby computer are connected to the pattern generator. The controls and switches for the pattern generators are located on the simplex maintenance console. Power for each pattern generator section is supplied by the standby simplex power sup-

ply. The test pattern generators are not marginal checked.

The GFI, XTL, LRI test pattern generator control panel is divided physically into three sections, one for each pattern generator. (See fig. 1-227.) These sections, identified for convenience of discussion, include:

TABLE 1-152. LRI CHANNEL INPUT SECTION NEONS

INDICATOR NEON	INDICATION
CHANNEL READY 1 and 2	The 1 neon is on when two words are in the buffer register waiting to be read to the drums.
BIT SYNC 1 and 0	The 1 neon is on when the corresponding flip-flop has received a data pulse for synchronization.
CLEAR SHIFT REGISTER	Turned on at sync time by a parity error or the absence of the busy bit; it is also set on at the end of the fast shift.
LOAD DATA	Turned on by the data pulse; the first core in the core shift register is primed.
TIMING SYNC 1 and 0	Same as BIT SYNC 1 and 0 neons.
BUSY BIT	Turned on by receipt of busy bit at pulse 6 of LRI message.
WORD LENGTH	Turned on when a busy bit also appears in bit 49 of the core shift register, indicating that a long word has been received.
PARITY COUNT	If on, incoming message has odd parity and is rejected.
FAST SHIFT	Turned on if the busy bit is present and parity is correct; the core shift register fast-shifts into the two core buffer storage registers; when the busy bit reaches the correct position in the core buffer register, the fast shift is stopped.
LAST SHIFT	Turned on by the first LR 8 pulse after the last shift delay flip-flop is turned on; this causes word 1 in the core buffer register to be shifted one more place to leave a blank core between words 1 and 2; since word 2 is read out first and since the last core shifts when it is read out, this blank core is necessary to prevent erroneous data in word 1.
SITE 8 & WORD LEVEL	Turned on during the read-out gates that condition the site identity levels to read out.
START WORD TRANSFER	Turned on if the channel ready flip-flop is set when a word-demand pulse comes from the common equipment.
READ OUT WD ONE and TWO	When the START WORD TRANSFER neon is illuminated, this conditions read-out word 2 FF which is set on by the next OD 4 pulse, illuminating READ OUT WD TWO neon; this causes word 2 to be read out to the drums and conditions read-out word 1 FF which is set on by the next OD 4 pulse, illuminating READ OUT WD ONE neon; this causes word 1 to be read out to the drum.
SITE IDENTITY COMMON EQUIPMENT A and B	The four site identity neons are energized by four lines from site identity cans in the common equipment to give a visual indication to the operator of the site associated with a particular channel.

- a. GFI pattern generator controls and indicators
- b. LRI pattern generator controls and indicators
- c. XTL pattern generator controls and indicators.

3.9.2 GFI Pattern Generator Controls and Indicators

The functions of each of the controls and indicators in the GFI pattern generator controls and indicators group are given in table 1–153.

INFORMATION NOT AVAILABLE

3.9.3 LRI Pattern Generator Controls and Indicators

The functions of each of the controls and indicators in the LRI pattern generator controls and indicators group are given in table 1–154.

3.9.4 XTL Pattern Generator Controls and Indicators

The functions of each of the controls and indicators in the XTL pattern generator controls and indicators group are given in table 1–155.

Figure 1–227. GFI, XTL, LRI Pattern Generators Panel

TABLE 1–153. GFI PATTERN GENERATOR CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
INFORMATION NOT AVAILABLE		

TABLE 1–154. LRI PATTERN GENERATOR CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
INFORMATION NOT AVAILABLE		

TABLE 1–155. XTL PATTERN GENERATOR CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
INFORMATION NOT AVAILABLE		

SECTION 4

MARGINAL CHECKING

4.1 ORGANIZATION

The marginal checking facilities built into AN/FSQ-7 Combat Direction Central are used to provide a high degree of equipment reliability. Marginal checking is performed by varying a supply voltage to a group of circuits. The variation of the supply voltage is called an excursion. The magnitude of an excursion necessary to produce operational circuit failure is called a margin.

The circuit margin can be determined by varying an applied voltage at the same time that a test program is run. In order to derive the maximum benefit from this type of testing, the program used must be designed to fully test the equipment covered by the marginal checking test program.

The probability of circuit malfunction during normal operation is very high when the circuit margin decreases beyond a prescribed value. To improve reliability, those circuits that have decreased margins should be replaced. The probability of trouble-free performance

until the next test program is very high if the circuit functions normally during the marginal checking program. There are three marginal checking elements in AN/FSQ-7 Combat Direction Central. Two identical marginal checking elements are provided for duplex equipment, one for Computer A and one for Computer B. The third marginal checking element is provided for the simplex equipment. Marginal checking programs are normally performed on that section of the duplex equipment that is in the standby status. However, provisions are also made to perform these tests on the active computer. Marginal checking operations can be performed only on simplex units that are in the standby status. One computer and the simplex equipment can be tested at the same time.

Marginal checking is accomplished by varying an applied voltage while a test program is run. If a voltage is varied, all circuits supplied by this voltage have the same excursion applied. Assume the voltage variation causes the gain of each circuit to which it is applied to decrease. Further, suppose that several of these circuits are combined to perform one operation. Although each circuit functions properly, the slight decrease of amplification through cascaded stages could produce an error indication. Thus, the test program would give an erroneous test indication. Now suppose that a positive excursion is applied, using these same circuits. The increase of amplification through cascaded stages could cause some components to be damaged by exceeding the safety limits.

In order to overcome the inherent disadvantages which arise when a system-wide excursion is applied, only one supply voltage to a single portion of the equipment is varied at any one time. Of the five d-c service voltages used for marginal checking, only one voltage can be varied at any one time. (See fig. 1-228.) In each computer the varied d-c voltage is applied to one of the eight marginal checking groups. Each of the equipment groups may be further sub-divided into six circuit groups, and each of the circuit groups may be divided into six lines. The varied voltage may be applied to any combination of circuit groups and lines within the equipment group. There are 1,440 (8x5x6x6) selectable marginal checking lines. At present, only about half of these lines are used.

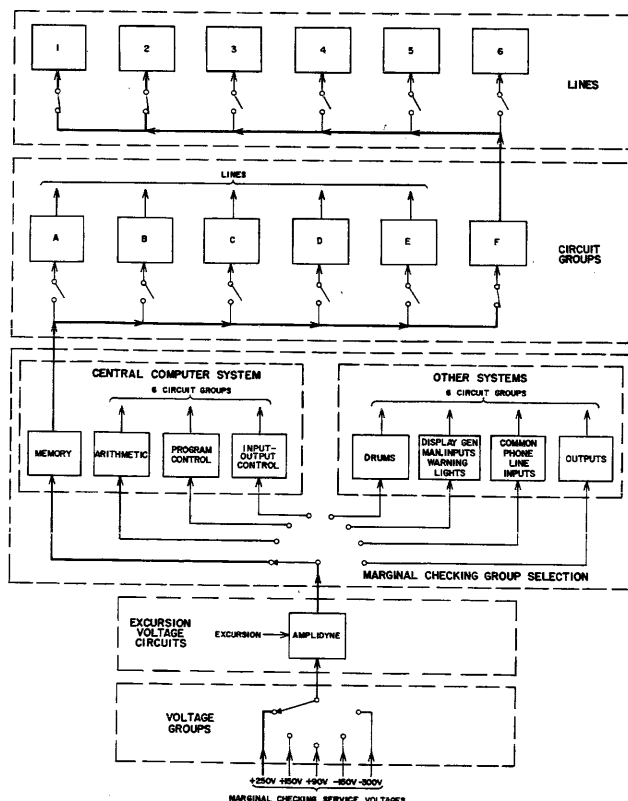


Figure 1-228. Marginal Checking Equipment Divisions

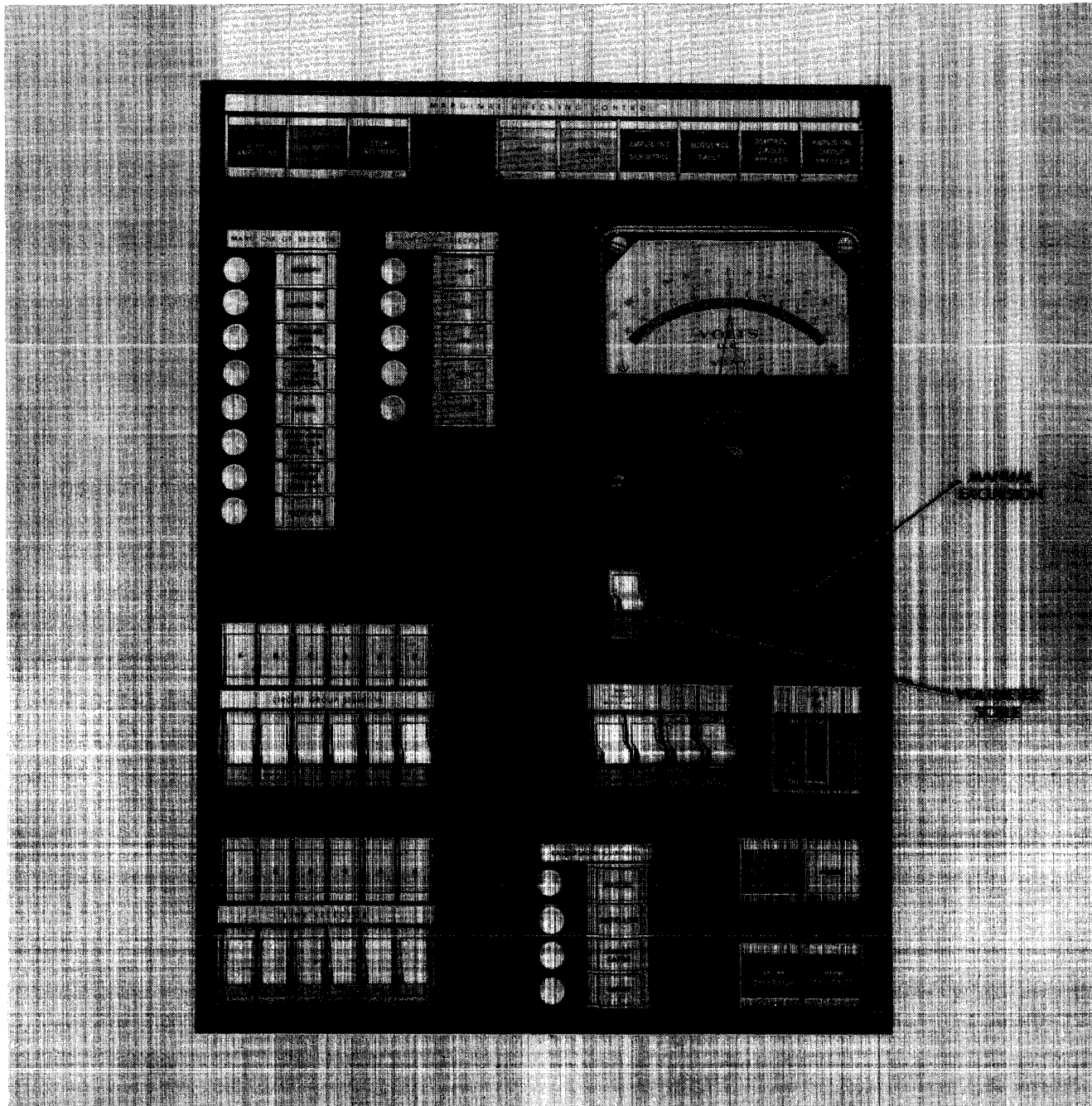


Figure 1-229. Duplex Marginal Checking Controls and Indicators

4.2 MODES OF OPERATION, DUPLEX

4.2.1 Manual Mode

There are three methods available for selecting the specific marginal checking lines, the voltage to be varied, and the polarity and the magnitude of the excursion. The three methods are manual, satellite, and calculator. The method is selected by operating the corresponding

pushbutton on the duplex marginal checking control panel. (See fig. 1-229.) When the MANUAL pushbutton is depressed, selection of the voltage, excursion magnitude and polarity, equipment group, circuit group, and line are made by use of controls on the duplex marginal checking control panel. The controls and indicators and their functions are listed in table 1-156.

TABLE 1-156. DUPLEX MARGINAL CHECKING CONTROLS AND INDICATORS

CONTROL	INDICATOR LIGHT	FUNCTION
START AMPLIDYNE (pushbutton)		When actuated, this switch causes the marginal checking amplidyne to start if the Central Computer System is in test status.
	AMPLIDYNE READY	This light illuminates approximately 50 seconds after the START AMPLIDYNE switch is operated to indicate that the amplidyne is up to speed and ready to be used in marginal checking.
	READY	Same as AMPLIDYNE READY.
EXCURSION CONTROL (pushbuttons)		A group of four interlocked pushbuttons which select the type of control exercised over the excursion.
MANUAL (pushbutton)		When operated, the excursion is controlled by MANUAL EXCURSION control.
	MANUAL	When on, this light indicates that the MANUAL pushbutton has been depressed.
SATELLITE (pushbutton)		When operated, the excursion is controlled by portable satellite units at various remote points.
	SATELLITE	When on, this light indicates that the SATELLITE pushbutton has been depressed.
CALCULATOR (pushbutton)		When operated, the voltage excursion is controlled and the marginal checking lines are selected by the test program.
	CALCULATOR	When on, this light indicates the CALCULATOR pushbutton has been depressed.
SPARE (pushbutton)		
	SPARE	
MARG CK GP SELEC- TOR (pushbuttons)		A group of eight interlocked pushbuttons which select the equipment group to be tested when MANUAL or SATELLITE switches are operated.
MEMORY (pushbutton)		When operated, the memory element is selected for test.
	MEMORY	On when the MEMORY switch has been operated.
ARITHMETIC (pushbutton)		When operated, the arithmetic element is selected for test.
	ARITHMETIC	On when the ARITHMETIC switch has been operated.
PROGRAM CONTROL (pushbutton)		When operated, the program control element is selected for test.
	PROGRAM CONTROL	On when the PROGRAM CONTROL switch has been operated.
IO CONTROL (pushbutton)		When operated, the IO control element is selected for test.
	IO CONTROL	On when the IO CONTROL switch has been operated.

TABLE 1-156. DUPLEX MARGINAL CHECKING CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR LIGHT	FUNCTION
DRUMS (pushbutton)		When operated, the drums are selected for test.
	DRUMS	On when the DRUM switch has been operated.
DISPLAY GEN MAN INPUTS WARNING LTS (pushbutton)		When operated, the display generation, manual inputs, and warning lights are selected for test.
	DISPLAY GEN MAN INPUTS WARNING LTS	On when the DISPLAY GEN MAN INPUTS WARNING LTS switch has been operated.
COMMON PHONE LINE INPUTS (pushbutton)		When operated, the common telephone line inputs are selected for test.
	COMMON PHONE LINE INPUTS	On when the COMMON PHONE LINE INPUTS switch has been operated.
OUTPUTS (pushbutton)		When operated, the outputs equipment is selected for test.
	OUTPUTS	On when the OUTPUTS switch has been operated.
VOLTAGE SELECTOR (lever switches)		A group of five interlocked pushbuttons which select the voltage to be used for the marginal checking test when MANUAL or SATELLITE switches are operated.
+250		When operated, the excursion is applied to the +250-volt line.
	+250	On when the +250 switch has been operated.
+150		When operated, the excursion is applied to the +150-volt line.
	+150	On when the +150 switch has been operated.
+90		When operated, the excursion is applied to the +90-volt line.
	+90	On when the +90 switch has been operated.
-150		When operated, the excursion is applied to the -150-volt line.
	-150	On when the -150 switch has been operated.
-300		When operated, the excursion is applied to the -300-volt line.
	-300	On when the -300 switch has been operated.
CIRCUIT GROUP SE- LECTOR A through F* (lever switches)		When actuated, the corresponding circuit group within the selected marginal checking group is selected for test; these circuit groups are composed of circuits which have similar requirements relative to magnitude and polarity of excursion for a marginal checking test.

TABLE 1-156. DUPLEX MARGINAL CHECKING CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR LIGHT	FUNCTION
	A through F	When on, these lights indicate which switches have been actuated.
LINE SELECTOR 1 through 6 (lever switches)		When depressed, the corresponding lines within the selected circuit group are selected for test.
	1 through 6*	When on, these lights indicate which lever switches have been depressed.
START EXCURSION (pushbutton)		This switch is not effective until the READY light is on. If on, depressing this switch causes an excursion to be applied to the line as determined by the MARG CK GP SELECTOR, VOLTAGE SELECTOR, CIRCUIT GROUP SELECTOR, and LINE SELECTOR. If this switch is depressed while an excursion is in process, the switch causes the existing marginal check operation to be cleared and then initiates the application of an excursion to the line as determined by the selection switches.
	EXCURSION ON	On when an excursion is in progress.
	VOLTMETER	Indicates the amount and polarity of the excursion voltage.
VOLTMETER SCALE (lever switch)		Determines which scale of the voltmeter is used; nonlocking in the 30 v position.
MANUAL EXCURSION (potentiometer)		Controls the magnitude and polarity of the excursion voltage when in MANUAL control.
STOP EXCURSION (pushbutton)		Returns the excursion voltage to zero and then releases the selection relays in proper sequence.
STOP AMPLIDYNE (pushbutton)		Causes the marginal checking amplidyne to stop.
	AMPLIDYNE CIR- CUIT BREAKER	On when an a-c circuit breaker on the amplidyne is open.
	CONTROL CIR- CUIT BREAKER	On when a circuit breaker is open on the —48-volt line used for marginal checking selection control circuits.
	SEQUENCE FAULT	On when an error has occurred in preparing to perform a marginal checking test.
	AMPLIDYNE SENSITROL	On when the amplidyne output is off limits as predetermined in the selection process; when illuminated, an automatic excursion off is initiated.
RESET AMPL SENSITROL (lever switch)		Resets the amplidyne voltage monitor relay.
MC DISP GEN (lever switch)		Places a dummy load on the display generator outputs to allow marginal checking of the display generators without using the indicator elements as loads.

TABLE 1-156. DUPLEX MARGINAL CHECKING CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR LIGHT	FUNCTION
SERVICE OPPOSITE DUPLEX SWITCH (lever switch)		Breaks the three —48-volt lines which go to the opposite duplex switch panel. This allows servicing of the connectors to the computer associated with the console being operated on the opposite duplex switch panel.
120V AC (circuit breaker)		This circuit breaker is in the 120-volt a-c line to the duplex maintenance console. This line supplies power for timer, buzzers, and the central scope and probes panel.

**When none or all of the CIRCUIT GROUP SELECTOR and/or LINE SELECTOR Switches are operated, all circuit groups and/or lines are selected.*

4.2.2 Satellite Mode *

When the satellite mode of operation is used, the equipment group, circuit group, and lines are selected by operating those controls on the marginal checking control panel. The excursion voltage magnitude and polarity are controlled by the operation of controls on a satellite unit located at a remote point. The satellite mode of operation permits the waveforms of the circuits under test to be observed during the test.

4.2.3 Calculator Mode *

The calculator mode of operation is used for auto-

matic marginal checking operations. Each marginal checking operational sequence is automatically initiated as required by the program. The program also selects the marginal checking group, circuit group, line, voltage, and the amplitude and polarity of the excursion. These selections are made by means of a marginal checking control word which is transferred from core memory to the live register of test memory. The excursion is then initiated by a *PER* (21)₈ instruction. The bit assignments and the corresponding selections are shown in table 1-157.

TABLE 1-157. MARGINAL CHECKING CONTROL WORD BIT ASSIGNMENTS

BIT	SELECTION	CODE	ACTION
LS	Change or start excursion.	1	Start excursion.
		0	Change excursion.
L1-L2	Restart after excursion applied.	00	Load from drums
		01	Continue from 00000 ₈ .
		10	Continue from 20000 ₈ .
		11	Load from card reader.
L3-L4	Restart after excursion removed.	Same as L1-L2	
L5-L6	Time duration of excursion.	00	Infinite
		01	3 seconds
		10	7 seconds
		11	30 seconds
L7	Polarity of excursion	0	Positive
		1	Negative
L8	Spare		

TABLE 1-157. MARGINAL CHECKING CONTROL WORD BIT ASSIGNMENTS (cont'd)

BIT	SELECTION	CODE	ACTION
L9-L12	Excursion magnitude	0000	0 volts
		0001	5
		0010	10
		0011	15
		0100	20
		0101	25
		0110	30
		0111	35
		1000	40
		1001	50
		1010	60
		1011	70
		1100	75
		1101	80
		1110	90
		1111	100
L13-L15	Voltage group selection	001	+250
		010	+150 CATHOD FOLLOWER
		011	+90
		100	-150 A FLIP FLOP
		101	-300
RS-R3	Marginal Checking Group	0001	1 Memory
		0010	2 Arithmetic
		0011	3 Program and control
		0100	4 IO Control
		0101	5 Drums
		0110	6 Displays
		0111	7 LRI, GFI, XTL common
		1000	8 Outputs
R4-R9	Circuit group	1001	Simplex
		10000	A
		010000	B
		001000	C

TABLE 1-157. MARGINAL CHECKING CONTROL WORD BIT ASSIGNMENTS (cont'd)

BIT	SELECTION	CODE	ACTION
R10-R15	Lines	000100	D
		000010	E
		000001	F
		100000	1
		010000	2
		001000	3
		000100	4
		000010	5
R10-R15	Simplex (RS-R3=1001)	000001	6
		100000	G
		010000	H
		001000	J
		000100	K
		000010	L
		000001	M

A duplex marginal checking excursion applied under calculator control is terminated either by the duration timers set by bits L5 and L6 of the control word or by a *PER* (22)₈ instruction. Similarly, a simplex excursion may be removed by a *PER* (23)₈ instruction. Programmed marginal checking is discussed in greater detail in Part 2, Chapter 10.

4.3 MODES OF OPERATION, SIMPLEX

Selection of the portion of the simplex equipment for marginal checking is somewhat different than the

selection of a portion of the duplex equipment for marginal checking. The simplex equipment contains 12 selections, referred to as circuit groups. It is possible to select these circuit groups in combinations. However, due to logical arrangement they will primarily be selected singly.

The simplex marginal checking control panel is shown in figure 1-230. Table 1-158 lists the functions of the controls and indicators on the marginal checking control panel.

TABLE 1-158. SIMPLEX MARGINAL CHECKING CONTROLS AND INDICATORS

CONTROL	INDICATOR	FUNCTION
START AMPLIDYNE (pushbutton)		Causes the marginal checking amplidyne to start.
	AMPLIDYNE READY	On when the amplidyne is running and a time delay has been allowed for the control circuits to stabilize.
	READY	On when the amplidyne is ready to be used for marginal checking but no circuit group has been selected.
EXCURSION CONTROL (pushbuttons)		Consists of a group of four interlocked pushbuttons which select the type of control exercised over the excursion.

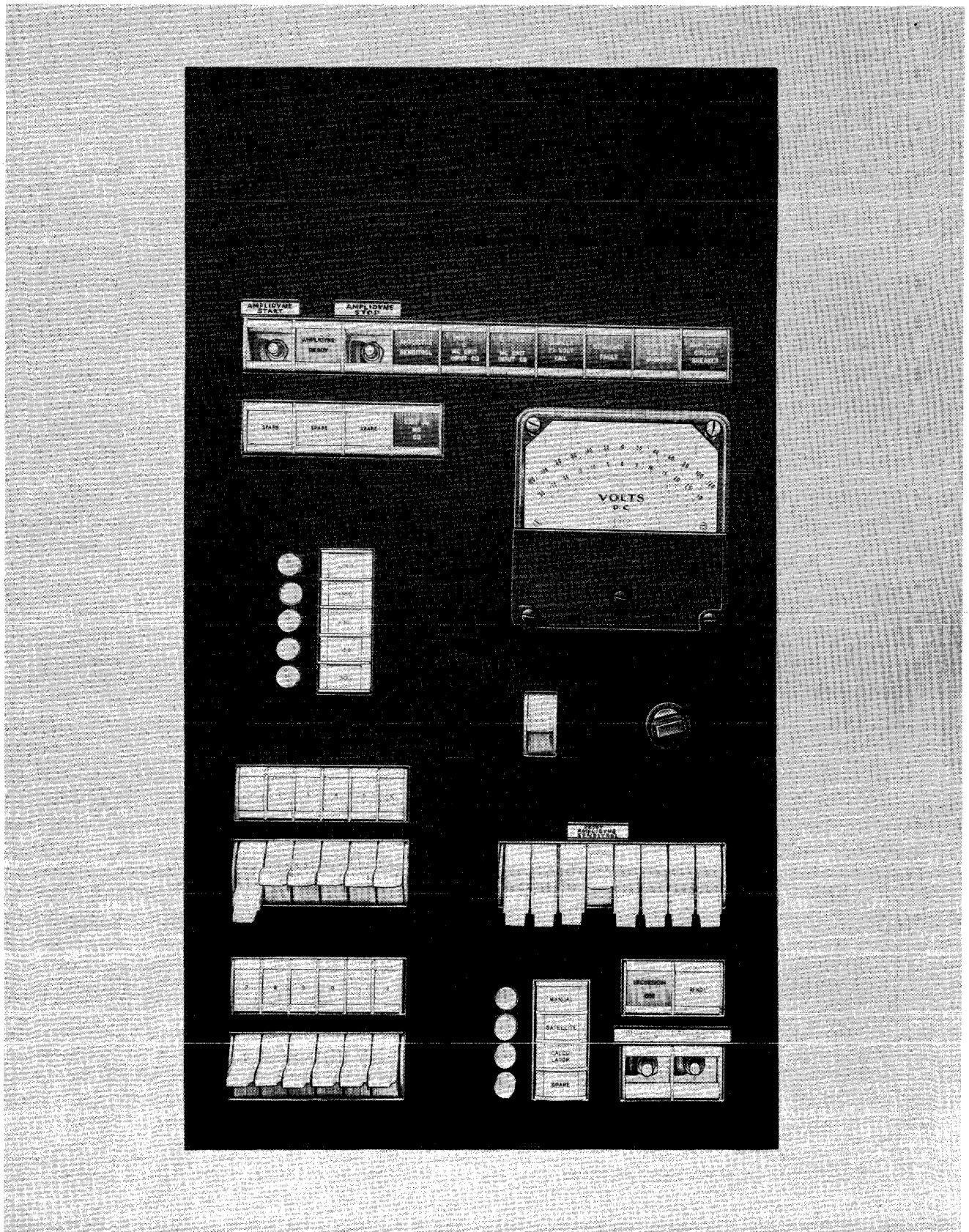


Figure 1-230. Simplex Marginal Checking Controls and Indicators

TABLE 1-158. SIMPLEX MARGINAL CHECKING CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR	FUNCTION
MANUAL (pushbutton)		When this switch is operated, the excursion is controlled by the MANUAL excursion control.
	MANUAL	On when the MANUAL pushbutton has been depressed.
SATELLITE (pushbutton)		When this switch is operated, the excursion is controlled by portable satellite units located at various remote points.
	SATELLITE	On when the SATELLITE pushbutton has been depressed.
CALCULATOR (pushbutton)		When this switch is operated, the excursion and the CIRCUIT GROUPS are selected by the test program.
	CALCULATOR	On when the CALCULATOR pushbutton has been depressed.
VOLTAGE SELECTOR (pushbuttons)		A group of five interlocked pushbuttons which select the voltage to be used for the marginal checking test when MANUAL or SATELLITE switches are operated.
+250		When operated, the excursion is applied to the +250-volt line.
	+250	On when the +250 switch has been operated.
+150		Same as +250 except that it refers to +150-volt line.
	+150	Same as +250 except that it refers to +150 switch.
+90		Same as +250 except that it refers to +90-volt line.
	+90	Same as +250 except that it refers to +90 switch.
-150		Same as +250 except that it refers to -150-volt line.
	-150	Same as +250 except that it refers to -150 switch.
-300		Same as +250 except that it refers to -300-volt line.
	-300	Same as +250 except that it refers to -300 switch.
CIRCUIT GROUP SE- LECTOR 1 through 12 (lever switches)		When the MANUAL or SATELLITE switches have been depressed, these lever switches select the circuit groups for marginal checking; these switches may be operated in any combination.
	CIRCUIT GROUP 1 through 12	Each of these lights is on when associated lever has been depressed.
START EXCURSION (pushbutton)		This switch is not effective until the READY light is on; if this light is on, depressing this switch causes an excursion to be applied to the marginal checking line as determined by the CIRCUIT GROUP SELECTOR and VOLTAGE SELECTOR; if this switch is depressed while an excursion is in process, the switch causes the existing marginal check operation to be cleared and then initiates the application of an excursion to the line as determined by the selection switches.
	EXCURSION ON	On when a line (or lines) has been selected for marginal checking.

TABLE 1-158. SIMPLEX MARGINAL CHECKING CONTROLS AND INDICATORS (cont'd)

CONTROL	INDICATOR	FUNCTION
	VOLTMETER	This meter indicates the amount and polarity of the excursion voltage.
VOLTMETER SCALE (lever switch)		This switch determines which scale of the voltmeter is used; normal position ± 120 v; with the switch operated, the scale is ± 30 v.
MANUAL EXCURSION (potentiometer)		This potentiometer controls the magnitude and polarity of the excursion voltage when in manual control.
STOP EXCURSION (pushbutton)		Causes the excursion voltage to return to zero and then releases the selection relays in proper sequence.
AMPLIDYNE STOP (pushbutton)		Causes the marginal checking amplidyne to stop.
	AMPLIDYNE CIR- CUIT BREAKER	On when there is an open circuit breaker in the amplidyne control d-c supplies.
	CONTROL CIRCUIT BREAKER	On when a marginal checking control circuit breaker has opened in the PCD unit —48-volt or +72-volt supplies.
	SEQUENCE FAULT	On when an error has occurred in preparing to perform a marginal checking test.
	SIMPLEX 72 VOLT FAIL	On when the voltage monitor relay on the output of the 72-volt supply has detected an off-limits condition.
	UNIT 56 D MC UNIT INPUT CB	On when a marginal checking input circuit breaker on unit 56 power supply D has opened.
	UNIT 56 C MC UNIT INPUT CB	On when a marginal checking input circuit breaker on unit 56 power supply C has opened.
	AMPLIDYNE SENSITROL	On when the amplidyne output is off limits as predetermined in the selection process; when illuminated, an automatic excursion off is initiated.
RESET AMPL SENSI- TROL (lever switch)		A momentary action lever switch; resets the voltage monitor relay which is on the output of the amplidyne.
	UNIT 58 MC CB	On when a circuit breaker has opened in marginal checking unit 58.

4.4 CODE ASSIGNMENTS**4.4.1 Marginal Checking**

The marginal checking group designation for the circuits on each block schematic is written beneath the logical reference number in the upper right-hand corner of the diagram. In some instances, the circuits on one block schematic fall into more than one marginal checking group. When this occurs, marginal checking group designations are listed beneath the logic number in numerical order. Asterisks are placed to the left of each additional marginal checking group designation; e.g.,

zero, one, and two asterisks are placed next to the first, second, and third designations, respectively.

4.4.2 Circuit Groups and Lines

The coding for circuit groups and lines is written in the lower left-hand corner of the individual circuit block symbol. For example, if circuit group A and line 6 are utilized to marginally check a flip-flop, the number A6 appears in the lower left-hand corner of the flip-flop symbol on the block schematic.

It may be necessary to marginally check a circuit with more than one marginal checking voltage. In this

case more than one circuit group and line are specified. When more than one circuit group and line are specified, additional designations appear directly beneath the first circuit group and line designation, but outside the circuit block symbol.

4.4.3 Indirect Checks

Occasionally, some circuits are checked indirectly. An indirectly checked circuit is a circuit that has no marginal checking line brought to it but is effectively checked by marginally checking the circuit at its input or output. An indirect check is indicated by placing the

letter M in the lower left-hand corner of the circuit block symbol.

4.5 MIMIC PANEL

A mimic panel is installed in the maintenance and programming room to present information on the status of the power and air conditioning systems. The information presented assists personnel in making rapid and logical decisions which help to maintain continuity of operation and aid in maintenance and troubleshooting. The mimic panel is located in the wall behind the duplex switching console. Table 1-159 lists the indicators and alarms and their functions.

TABLE 1-159. MIMIC PANEL INDICATORS

INDICATORS	ALARMS	FUNCTION
------------	--------	----------

INFORMATION NOT AVAILABLE

PART 2

PROGRAMMING APPLICATIONS

CHAPTER 1

OPERATIONAL PROGRAM ORGANIZATION

SECTION 1

INTRODUCTION

1.1 PURPOSE

The operation and capabilities of the systems within AN/FSQ-7 Combat Direction Central have been described in Part 1. This equipment has been designed to perform one major function: direction of the defense of an area against air attack. In order to perform this assignment, operations must be directed by programs written for that purpose. Part 2 presents a summary of the information pertinent to programming and some discussion of the types of programming that are required. In particular, this chapter describes the organization of the programs needed for performance of air defense functions by AN/FSQ-7 Combat Direction Central.

1.2 DIRECTION CENTER FUNCTIONS

1.2.1 General

The functions performed by AN/FSQ-7 Combat Direction Central and by the operating personnel associated with it are grouped into specific functions, which are described separately.

1.2.2 Air Surveillance

1.2.2.1 Overall

The air surveillance section of a direction center gathers information, processes it, and presents the current air situation for use by other operational sections of the direction center.

1.2.2.2 Radar Inputs and Mapping

In order to facilitate the tracking process, radar data is filtered to remove extraneous returns caused by ground clutter and cloud returns, interference, or jamming. They are also evaluated for purposes of locating defective equipment and keeping aware of radical changes in the tactical air situation. The program automatically performs some of these functions, while others are done manually.

Mapping is the manual rejection of data; this is accomplished by painting on the face of the mapping tube with a semi-opaque material. In the case of the long-range radars this is done at the radar site, while with the gap-filler radars it is done at the direction center. It is primarily intended to screen out areas where ground, sea, weather clutter, or jamming is great enough to prevent tracking. The mapping is performed by radar mapping operators under the supervision of radar mapping supervisors (MSL and MSG), who are responsible to the Air Surveillance Officer. The GFI mapping operators consoles are equipped with warning lights and audible alarms, which are used to call attention to unfavorable data situations. A mapping supervisor monitors the mapping operations of all the mappers under him at his console.

The mapping supervisors are also responsible for the insertion, by switches, of evaluations of the quality of the data from the sets. Four categories are possible: good (G), marginal (M), poor (P), and jammed (J).

Masking is the term applied to the process by which the returns from each radar are examined by the computer and either accepted or rejected. This is done primarily to save program time, by eliminating unnecessary processing of information from areas covered adequately by other radars, and to eliminate coverage outside the overlap line.

1.2.2.3 Track Detection and Initiation

The track detection and initiation section assists the computer in its automatic initiation function by taking advantage of the ability of a manual track initiator to work in areas of noise or high traffic density beyond the programmed ability of the computer, by assisting the computer in judgment on marginal cases, and by taking over the responsibility of initiation in certain areas when required by heavy load conditions.

The responsibility for track initiation lies with the initiation supervisor, who co-ordinates the activities of track initiators and also controls the areas in which automatic initiation is to take place. The latter is done by means of mapping on the area discriminator, a photo-electric pickup on a situation display console. Each track initiator is responsible for detection and initiation on tracks (excluding interceptors, normally) which lie in an assigned area and may confirm, as established, those tracks which the computer has designated as tentative. The initiation procedure consists of examination of the uncorrelated data on the display console, deciding on, designating (by light gun action), a specific radar return, and inserting speed and heading by switch action.

1.2.2.4 Track Monitoring

Track monitoring provides manual assistance to the automatic tracking program. To a considerable extent, the computer handles automatic tracking without any outside intervention. However, when tracking situations arise which are of a complexity beyond the programmed ability of the computer, monitoring is required. The program examines the most recent scans of track history and, according to certain established criteria, decides whether or not monitoring is required. Ambiguous data distributions, absence of data returns, entry of the track into an area of excessive noise or clutter, and merging of two tracks whose courses are nearly parallel are among the situations which may require the attention of track monitors. The subsector area is divided and allocated among the track monitors by the tracking officer. When the program detects tracking difficulty, it automatically assigns the track monitor on the basis of the area the track is in and also assigns to it a priority, based on such considerations as its location, identity, quality of tracking, etc. To alert the track monitor to his assignment, the track is forced on his display. He must take action on it or, if in doubt, refer it to his track supervisor. Lost tracks (i.e., those on which the tracking is too poor to warrant continued correlation) are assigned to the special track monitors.

Monitoring is not limited to requests concerning tracks in difficulty. Manual requests for monitoring of tracks may be made by either the tracking officer or a tracking supervisor. Normally, intercept directors perform the track monitoring function on interceptor tracks. However, they may request assistance from the track monitors.

1.2.2.5 Crosstelling

Air surveillance crosstelling between subsectors maintains continuous tracking on aircraft as they pass from one subsector into another. This requires the flow of information and transfer of tracking responsibility to the new subsector as the aircraft nears it. The program, monitored by the overlap technicians, will handle

the transfer of tracking responsibility between two subsectors. When the information originates from adjacent manual centers, airborne early-warning aircraft or picket vessels, it is received in the manual inputs section, punched on cards, and inserted into the computer. The overlap technicians manually establish tracking on the incoming track. The intercept directors are responsible for interceptor tracks and for the handover of interceptors to adjacent subsectors.

Weapons direction crosstelling is the transfer of track identification and assignment data on interceptors, targets, other tracks, and certain other data such as interception point information. The function alerts the receiving subsector (warning crosstelling), supplies necessary information to enable it to conduct interceptions (interception crosstelling), or enables it to relay information to antiaircraft defenses. Weapons direction crosstelling is generally done automatically by the program with the provision that the combat center, senior director, senior weapons director, or antiaircraft directors may specify additional tracks to be crosstold. In the case of an adjacent manual subsector, the crosstelling information is sent to the manual subsector by automatic teletype, is received from the manual subsector by voice telephone, and is entered into the computer by the manual data inputs (MDI) section.

1.2.2.6 Forward-telling

Forward-telling transmits to higher headquarters a summary of the subsector air situation and other information which is required to perform its assigned functions. For the most part, the transmission of this data will be automatic, requiring no action on the part of direction center personnel.

1.2.2.7 Height Finding

Height finding in the direction center primarily provides height data on all tracks being carried. A secondary function is to provide information on the tracks as to flight size, type of formation (extended primarily in azimuth, range, or elevation), and average range separation.

Height finding is handled automatically through data links. The computer generates requests for height-finding information, transmits them in accordance with a tactical priority scheme, and receives and uses directly the replies from the radar site. Provision is made for a manual procedure, in which case requests and replies for height information will be made by telephone, and data will be inserted into the computer via keyboard switches by the height technicians. Both modes may be in operation at the same time.

Supervision of all height-finding operations is handled by the height supervisor. Working under and with him are height technicians, who transmit voice requests and insert replies while on manual operation; they also

insert data received from sources such as weapons directors and antiaircraft directors at each radar site.

1.2.2.8 Identification

The goal of the identification section is to determine which of the non-interceptor tracks in the system are friendly aircraft. To accomplish this, various methods and techniques are used to correlate the actual tracks and track behavior with such information as is available on aircraft movements. The identification officer and his technicians are responsible for the identification and correlation of the activities of the section with other operations of the direction center. The classifications into which an aircraft may be assigned are pending, friendly, unknown, hostile, faker, round robin, special, and interceptor.

A principal method in identifying tracks is comparison with pre-filed plans which are inserted into the computer by the MDI section. Anticipated check points plotted from these flight plans are compared with actual radar track data; concurrence within certain error limits may result in a track's being identified as friendly. Both situation and digital displays of flight plan data are available for comparison with the track data.

Another method is the use of the multiple corridor identification system, under which transoceanic aircraft are assigned a specific corridor, turn, code word, and maneuver. Appearance of the aircraft in the specified corridor, plus execution of the check turn and/or answer of the code word, may result in the classification of the track as friendly. The check maneuver may be used as an added check in uncertain cases. Tracks originating in certain areas, called open or defense areas, may be designated friendly. The identification section also monitors tracks which have exhibited behavior indicating a possible need for reidentification.

1.2.2.9 Raid Forming

Raid forming is the function of summarizing and consolidating the air picture for the subsector commander and other supervisory personnel whose functions are concerned with overall planning and evaluation. Raid forming consists of consolidating individual tracks whose heading, speed, and proximity to each other indicate a number of aircraft acting together as a unit. When these tracks grouped together are hostile, unknown, faker, or pending, a raid is formed; a group is formed when the tracks consolidated are interceptor tracks. This operation is the responsibility of the air tactics officer with the assistance of the air tactics technician and is done with keyboard switches and light gun. Tracking by the program continues as before on individual tracks, but, for display purposes, special raid vectors are generated, the mean speed and the heading of all tracks within the raid are calculated, and tabular

information is displayed. Provision is made for the handling of raids and groups, such as adding tracks to raids, dropping tracks from raids, combining raids, etc.

1.2.3 Weapons Direction

1.2.3.1 Overall

Weapons direction is the name applied to the function which encompasses all activities associated with the committing and assigning of weapons and the subsequent execution of the attack or mission for which the weapons are assigned. The activities of the weapons direction section can be conveniently broken down into three general areas of operation: weapons assignment, intercept direction, and antiaircraft direction.

1.2.3.2 Weapons Assignment

Weapons assignment endeavors to ensure that sufficient weapons are available to meet any existing or potential threat, examines each threat, commits weapons against specific targets, and delegates responsibility to an intercept director and/or antiaircraft director for the execution of the mission. In addition to assigning aircraft on interception missions, weapons assignment designates aircraft for combat air patrol, return to base, deployment, and flight following.

The officers responsible for weapons assignment are the senior director, the senior weapons director, and the weapons directors. Under normal manning conditions, the senior director and one weapons director are on duty while, under alert conditions, the senior weapons director and other weapons directors are added. These personnel have access to situation and digital displays, keyboard switches, alarms, and telephone lines, which supply them with information necessary for making decisions and which afford them communication with the computer and with each other. In addition, each has a technician available to assist him.

In a typical sequence of operations, a weapons director decides to commit weapons against a target, scrambles interceptors, and assigns responsibility for the monitoring of the interception to an intercept director. On occasion, the senior director or the senior weapons director makes the decision to commit weapons against a target and, by switch action, assigns responsibility for the target to a weapons director.

1.2.3.3 Intercept Direction

Intercept direction implements the successful completion of assigned missions and is the responsibility of the intercept directors. Five intercept directors, each with an intercept technician, constitute a team under the supervision of a weapons director. Under normal manning conditions, one such team is on duty; under alert conditions, as many as four teams may be on duty.

The five types of missions which may be assigned to intercept directors are: interception, return-to-base,

combat air patrol, deployment, and flight following. Interception missions are divided into two types: normal, where interceptors are committed against hostile, unknown, faker, or pending tracks; and special, where they are vectored against friendly, special, or round robin tracks. Either type of interception may be performed by actual interceptor tracks or by specially designated interceptor tracks (noninterceptor tracks re-identified as interceptor tracks).

On a normal interception mission, a weapons director selects, assigns, and scrambles aircraft against a track and assigns responsibility for control of the mission to an intercept director. The intercept director will select and insert parameters, such as altitude, speed, and tactics, into the computer by keyboard switches. The computer will provide vectoring instructions for the interceptor pilot for the ground control phase of the mission, which will be transmitted to the pilot by data link. In the absence of the data link, the intercept director will transmit vectoring instructions by voice radio. Upon completion of the mission, ground control is resumed for the return-to-base phase.

Throughout the mission, the intercept director monitors the track on his situation and digital display, stands ready to take action to ensure the safety and effectiveness of the mission, and implements the handover of control of interceptors when the situation requires.

The computer will generate vectoring instructions for return-to-base and deployment missions. On flight following missions, intercept direction involves following the movement of the aircraft in question on the air situation display and, when possible, furnishing to the aircraft, by radio, any information which will assist in a safe flight.

1.2.3.4 Antiaircraft Direction

The antiaircraft direction section supplies information to, and co-ordinates the operations of, each of the antiaircraft defenses within the subsector. There will be an antiaircraft director and an antiaircraft director assistant for each of the antiaircraft operation centers.

The program, under control of, and monitored by, the antiaircraft direction section, selects tracks to be told to the antiaircraft operations center (AAOC), assembles the required information on these tracks, produces the messages, and transmits them to the AAOC. Affecting the selection of the tracks are the relative tactical importance of the tracks, the condition of operational control in effect, etc. By keyboard switch action, an antiaircraft director may add or delete tracks to be told to the AAOC.

An additional function performed by the program under the supervision of the antiaircraft direction section is the processing of replay messages from the AAOC. These contain fire unit operational status infor-

mation (for insertion by card) or engagement status information for pushbutton insertion.

1.2.4 Manual Data Inputs

The manual inputs section receives, processes, and inserts into the computer information which is not transmitted into it automatically and directly. This information is received by voice telephone or by manually operated teletype and punched in coded form into IBM cards, verified, and read into the computer. Manual inputs information includes flight plans, weather data, Ground Observer Corps reports, reports from picket vessels and from advance early-warning aircraft, cross-telling reports from adjacent manually operated direction centers, information on weapons status, and miscellaneous control parameters for the computer program.

The manual inputs supervisor has the overall responsibility for the operation of the section, which is composed of a number of manual inputs technicians who man the telephone and teletype stations, punch the cards, and insert the cards in the computer entry punches. The manual inputs supervisor has access to a digital display to aid him in monitoring the data input into the computer.

1.2.5 Weather

The weather section collects, processes, and furnishes to the computer and other sections in the direction center information on present and forecast weather conditions and winds aloft data. This weather data is presented in the form of briefings to the personnel concerned, as they come on duty, and answers to special requests, and by means of situation and digital displays. The winds aloft data is stored for use in intercept calculations.

A primary source of weather data is the terminal weather reports. These arrive at the weather station by teletype, are processed, screened, and relayed to the manual inputs section where the information is punched on cards and read into the computer. Other weather reports, consisting of significant changes in the weather, are transmitted by teletype directly to both the manual inputs section, into which they are inserted, and the weather station.

1.2.6 Subsector Command Post

The subsector Command Post exercises overall supervision, planning, and control of the air defense of a subsector and co-ordinates functions with adjacent subsectors and with higher headquarters. As an aid in performing its functions, the Command Post is provided with situation and digital displays. The summary situation display, photographed from a situation display tube at least every minute and projected onto a large screen in the Command Post, shows the overall nature

and broad outlines of the air situation and is supplemented in detail by various situation and digital displays.

1.2.7 Training and Battle Simulation

Training and battle simulation, in addition to providing training practice for operators, plans and carries out synthetic exercises and evaluates the performance of the man-machine combination during such exercises. These exercises are designed to prepare the operational personnel in a direction center, in peacetime, for combat conditions by offering experience in operating under heavy load conditions.

Prior to the synthetic system operation, the radar and height returns for a specified air situation are generated and written on magnetic tapes. During the operation of the exercise, the simulated radar inputs from the tape and actual radar data may be mixed. Simulated radar and IFF returns are manually specified for interceptors. Control of live fakers is provided. Other simulated data is inserted by the manual inputs section.

To evaluate efficiency and to diagnose weak spots, information received during the simulated exercises is recorded for later analysis. Pertinent information on each track in the system, unusual conditions arising and action taken, etc., is logged by writing on magnetic tapes or by photographing selected displays. In addition, audio tape records will be used to record telephone and radio communications.

1.2.8 Standby Operation

Standby operation ensures, as far as possible, that breakdown of the defense function will not occur. Switchover from one computer to the other can be accomplished by the flip of one switch, and transfer of control is accomplished usually within 15 seconds. Under normal conditions, while one computer carries out the normal air defense tasks, the other computer is occupied in several activities, all contributing to continuous operation of the direction center.

The standby computer receives data from the active computer, subjects it to examination, and stores it, thus performing a check on the active computer and also keeping current information in case of a switchover.

Indications of the reliability of the active computer are presented on digital display tubes in tabular form. This serves to facilitate corrections of malfunctions in the computer and to indicate the advisability of switching between computers.

1.2.9 Recording and Analysis

During the operation of the system, pertinent information is recorded on magnetic tape, selected situation and digital displays are photographed, and telephone and radio communications are recorded on audio tape. These records relieve the operators of logkeeping

and are used to evaluate efficiency, to diagnose faults, and to indicate areas requiring changes in procedure. Records may be kept permanently for future reference.

1.3 RESPONSE TIME

1.3.1 Introduction

The response time of the direction center (that is, the time required to react to a change in the environment) depends upon two factors: human response time and response time of the automatic functions performed by the computer. The response times for automatic functions are of two types:

- a. Response to operator action, which is the time elapsing between a switch insertion by an operator and the accomplishment of the desired result (e.g., the presentation of a requested display)
- b. Response to an external input; for instance, the time elapsed between a change in speed or heading of an aircraft and the calculation of a new velocity for the aircraft track.

It is important to keep response time to operator actions low in order to enable personnel to carry out their duties efficiently. Response to external inputs can sometimes be slower, depending on the relative effect of the input on the execution of the air defense function.

The computer performs the various automatic direction center functions under control of a number of programs or sequences of instructions. These programs, which are stored in the Drum System and core memory, operate in an established order, each one carrying out a particular part of the overall air defense function. These parts are called program functions and do not necessarily bear an exact one-to-one correspondence to the air defense functions. For example, one program function (switch interpretation) processes the switch insertions from all the operating stations in the direction center.

Response times both to external inputs and to operator actions are determined by the operation of the programs within the computer. Two factors are significant: the frequency or cyclical rate of operation of the programs and the sequence or order in which the programs are performed. The frequency of operation of each program is, to a certain degree, dependent upon load; under light load conditions, some calculations may be carried out at a slightly faster rate than if the system is under full load conditions. The sequence in which the programs are operated does not depend upon load; each program in the series performs its particular calculations as many times as necessary to process all items (e.g., tracks, radar returns) then present in the computer before the next program is initiated.

1.3.2 Timing and Sequencing of Programs

It is convenient to synchronize the operation of the series of programs in the computer with the situation display subsystem; this synchronization is accomplished by periodically sensing the state of the display cycle. (A display cycle is the time required to display all track data and radar data once; i.e., approximately 2.5 seconds.) The time between successive Display System synchronizations is 6 display cycles, or about 15 seconds; this time is known as the program frame synchronization time, and the programs which are scheduled to be performed during this time make up the program frame. Most programs are operated at least once in the frame.

If the length of time required to complete the operation of all the programs in a frame is less than 15 seconds, time-buffering (waiting until the end of the frame time) is used. If, under heavy load conditions, 15 seconds are not sufficient time for all the programs in the frame to operate, the frame time is extended to 7 display cycles (about 18 seconds), and system response times suffer slightly.

Some programs are operated three times during the frame. In order to equalize the time between successive operations of these programs and to prevent bunching of programs in the first part of the frame under light load conditions, time-buffering is also used at two points within the frame. The program operation is buffered only roughly at these points (to about 1 second), dividing the frame into three approximately equal parts (about 4-1/2 to 6 seconds long), called subframes. No program operates more frequently than once per subframe.

Although no calculation is done more often than every 5 seconds, the operation of equipment through which the computer receives and transmits information is not limited to this rate since, with the exception of the Display System, the operation of the auxiliary equipment is not synchronized with the computer programs. For example, the keyboard inputs can be read into the computer two, three, or more times per subframe, even though the program which interprets these insertions is operated only once per subframe. Similarly, messages to semiautomatic height-finders can be set to be transmitted with any desired time separation although priorities are calculated and messages prepared only once per frame.

1.4 DIRECTION CENTER PROGRAMS

1.4.1 General

The programs used with AN/FSQ-7 Combat Direction Central comprise four major classes, each containing other, shorter programs. These four programs (or classes) are:

- a. Operational active

- b. Operational standby
- c. Operational supporting
- d. Utility supporting.

Each class serves a particular function within the overall purpose of the equipment.

1.4.2 Operational Active Program

The operational active program executed by the active computer performs all data-processing operations of the air defense function, including preparation of reports on all aircraft within the subsector covered by the equipment, implementation of tactical decisions made on the basis of the information presented, and monitoring of all tactical actions and situations within the subsector. The operational active program contains a number of shorter programs which, taken together, perform the necessary air defense operations. These shorter programs are blocked out by program function rather than by air defense function; i.e., they are organized for convenience in writing rather than for correspondence to air defense functions. Their sequence of performance is controlled by a sequence selection program which is, in effect, the master active air defense program.

1.4.3 Operational Standby Program

The operational standby program executed by the standby computer consists of a master program and several short programs which, taken together, perform the following functions:

- a. Monitor the operation of both the standby and the active computers
- b. Maintain a summary of air defense data in the standby computer to reduce the time necessary for generation of a picture of the air situation in that computer, if it should be switched to active because of failure of the other computer
- c. Perform some keyboard message interpretation and display makeup for the duplex switching console
- d. Control the interleaving of maintenance or auxiliary data-processing programs by the standby computer with the programs which perform the other three listed functions.

1.4.4 Operational Supporting Program

The operational supporting program contains those programs which do not perform real time air defense functions; i.e., their functions need not meet the requirement, imposed upon the active and standby programs, that results be obtained in time to influence the situation from which the results were derived. Two general functions are performed by operational supporting programs: generation of simulated data for use in tests and training exercises and reduction of data recorded during active operation.

1.4.5 Utility Supporting Programs

Utility supporting programs have no direct connection with air defense data-processing. Instead, these programs are used in preparing, handling, checking out, and revising other programs. There are two general types of utility supporting programs: general utility programs, such as the compiler and the checker; and check-out utility programs, such as table simulation programs and the system checker. These latter two, the table simulation programs and the system checker, are designed to systematize and simplify parameters and system checkout of the operational active and standby programs.

1.4.6 Maintenance Programs

In addition to the four major programs and program types just described, there is a class of programs intended solely for maintenance of the equipment. In general, maintenance programs can be divided into three types: reliability, diagnostic, and auxiliary programs.

Reliability programs are those which are used to check the operation of specific portions of AN/FSQ-7 Combat Direction Central. With the performance of these checks, errors caused by circuit failure can be detected rapidly. Included in the error detection performed by reliability programs is the discovery of failures that may occur only under particular operating conditions, such as failures that appear at specific repetition rates and for certain combinations of bits. In order to discover these and other types of errors, reliability programs check logical operation, paths of information flow, timing, ability of equipment to function in all states, execution of instructions, and so on. Marginal checking is used with reliability programs to detect possible future trouble. Certain voltages in the equipment are varied by the program during program operation time. The magnitudes of these voltages can be varied within the operating limits of the circuits being checked or until these circuits fail.

Successful operation of a program without a failure indication during variations of marginal checking voltage within circuit operating limits indicates that the portion of the equipment checked will probably operate for an indeterminate period of time. However, checks of this kind do not provide a guarantee of prolonged operation, since components that do not exhibit deterioration discoverable by voltage variation procedures can fail suddenly. Reliability programs which are run in conjunction with marginal checking techniques are usually shorter than those designed to run without marginal checking. However, most reliability programs are capable of being run both with and without marginal checking.

Diagnostic programs are corrective maintenance programs which localize malfunctions to as small an area of the equipment as possible. In general, diagnostic programs are designed to isolate known failures, unlike reliability programs, which are designed to discover these failures. However, there is no clearly defined distinction between the two maintenance program types. Reliability programs can provide indication of the nature and location of a failure and may actually be used as diagnostic programs. Correspondingly, diagnostic programs may indicate that a given portion of the equipment is operating reliably.

Marginal checking techniques can be used with most diagnostic programs. However, those diagnostic programs designed to work exclusively with marginal checking voltage variations are less complex than other diagnostic programs. Failures that are observable only during specific marginal checking voltage variations used by a diagnostic program are localized to the group of circuits affected by the voltage variations used.

Auxiliary programs are used to generate signals in program-controlled equipment while adjustments or electronic tests are performed on that equipment. Thus, they are exercising programs without the checking features included in reliability or diagnostic programs.

SECTION 2

OPERATIONAL ACTIVE PROGRAM

2.1 PROGRAM DESIGN

2.1.1 Introduction

The operational active program comprises a number of shorter programs operating under the direction of an executive program (the sequence control program) to perform all calculations required for implementation of air defense direction. The individual programs within the operational active program, numbering about 35, vary in length from 500 to 4,000 instruction words; the entire operational active program is approximately 60,000 words in length. Since only a portion of the entire program can be held in core memory at any one time, all the programs making up the operational active program (except the sequence control program) are held on auxiliary memory drums C through H, whose write circuits are locked out of operation to prevent inadvertent erasure of the program. The sequence control program held in core memory loads each program just prior to its proper time of execution within the sequence of programs and turns over to it the data collected or processed by other programs in the sequence. This data is stored either in core memory or on one of the auxiliary memory fields in tabular form for easy access to specific items of data.

The individual programs making up the operational active program are organized primarily for convenience in programming rather than for a 1-to-1 correspondence to air defense functions. These air defense functions may be divided into two major blocks, air surveillance and weapons direction. The air surveillance function includes the compiling of all radar and nonradar target reports, the sorting of these reports for revision of track data messages or for initiation of new track data messages, and the identification of each track within the subsector. The weapons direction function includes evaluation of the threat posed by each hostile or unknown track, evaluation of counter measures available, and assignment and direction of weapons for destruction of threatening targets. The programs written to implement these functions do not correspond directly to these functions. Some programs may operate together but may be split into two programs to reduce their length and to allow writing two programs, each performing distinct functions, rather than one program performing two functions. For example, the radar input program, which converts radar polar co-ordinates to subsector rec-

tangular co-ordinates, and the correlation program, which sorts radar reports and assigns them to tracks or indicates that new tracks should be started, could be written as one long program. However, since their functions can be separated, it is more convenient to write two separate, short programs. In other cases, two short programs may be combined into one larger program for convenience in handling the block of instruction words making up that program, when transferring it to core memory for execution. Further, certain programs may contain subprograms which are used by other programs. For example, if one program contains a binary-to-decimal conversion routine, that routine may be called for and used by another program rather than having it duplicated in the second program.

Functionally, programs within the operational active program are of two types: central group programs and operational group programs. Operational group programs are those which perform separable automatic air defense functions; e.g., calculating interceptor vectoring commands or smoothed track positions and velocities. These programs are generally fairly long, mathematical or logical computation sequences which require access to a small number of tables of data processed by other programs and very little communication with other programs. Central group programs are those which perform bookkeeping (table makeup, for example), inter-program communication, and service functions for several or all air defense functions. For example, the program which makes up displays and the program which interprets manual data input keyboard messages are central group programs. The central group programs generally require a large amount of communication with other programs and must have access to a large number of data tables. The sequence control program, by definition, is a central group program.

Although no detailed descriptions of the programs within the operational active program can be given, the remainder of this section presents a general picture of the programs within the operational active program and their functions. A discussion of the timing of the operational active program is also provided.

2.1.2 Central Group Programs

2.1.2.1 Sequence Control Program

The sequence control program controls all transfers of program blocks from auxiliary memory to core mem-

ory for execution. In addition, in order to attain utmost efficiency in the use of the IO facilities in the Central Computer System, most IO operations are set up and initiated by the sequence control program. While a program is being executed, it senses the IO interlock periodically. When the IO interlock goes off, the program branches program control to the sequence control program, which initiates a new IO operation in accordance with the parameters indicated by the program requiring the IO operation. In this way, no internal operation time is lost in waiting to set up and initiate an IO operation while the IO interlock is on for a preceding IO operation.

The sequence control program maintains clock time in the computer; i.e., it establishes the basic time interval used for determination of approximate velocities by those programs which do not compute velocities with the precision made possible by the more lengthy calculation involving the real time clock. The sequence control program also performs certain bookkeeping functions, including the logging of running time for each other program and placing in the test register (and therefore displaying on the test register neons) a number identifying the program or program group being executed at any given time. Further, intercommunication with the standby computer via the writing of information on the intercommunication (own) field of the active computer is controlled by the sequence control program, as is the reading, if any, of the intercommunication (other) field.

2.1.2.2 Keyboard Message Interpretation Program Group

The keyboard message interpretation program group, which interprets operator actions as indicated by keyboard messages, is a central program group, since operator actions must be capable of affecting a large number of other programs. The keyboard interpretation program group examines the image of the manual input matrix generated in core memory as a result of reading the matrix in an IO operation. This program group also utilizes the light gun messages read from the manual input field. The keyboard messages are checked for completeness (the presence or absence of ACTION pushbutton bits or light gun bits) and for legality (whether the action requested is possible or allowed for the operator which requested it), and the actions requested are then carried out if they satisfy these two requirements. Legal actions involve making changes in data tables which are used by other programs and, in some cases, the generation of new data from the insertions. Display requests are handled by this program group by referring the type of display requested and the routing of that display to the requesting console to the programs which actually make up the

displays. Illegal actions are recognized by the keyboard message interpretation program and result in illegal action alarms (digital displays) at the requesting console.

2.1.2.3 Situation Display Program Group

The situation display program group makes up, routes, and monitors situation displays. (When necessary, the program also removes certain displays if they are invalid or have been replaced.) The situation display program group makes up special attention displays and other special displays when notified of requests for these displays (or requirements for these displays in alarm situations) by other programs, particularly the keyboard message interpretation program. The situation display program group also performs its own bookkeeping, including the calculation of priorities for the use of attention displays (vector messages used to call attention to other messages) and the calculation of the length of time for which attention displays are presented.

The track display program (one portion of the situation display group) makes up displays for all tracks (targets on which correlated radar data is available and used to generate a track data message). The functions of this program include changing the vectors, the positions, and the information symbols within each track data message as required by the motion of the target represented by that message. Pending tracks (targets on which a track has been initiated but not identified) have attention displays added to them by the track display program. In addition, the track display program contains all subprograms used in common by other programs in the situation display group.

The air surveillance situation display program makes up displays for monitoring of tracks which cannot be handled in a routine manner (crossing or splitting tracks), track history displays (vector messages wherein each vector represents an average of eight radar data messages on a particular target), and attention displays in conjunction with these two functions.

The weapons direction situation display program makes up weapons assignment displays, interception point displays (indications of the point of interception of a hostile track by an interceptor), and attention displays associated with these functions.

The situation display geography program makes up displays which present visual indications of salient geographical features within the subsector covered by the central. In addition to the geographical boundaries of the area, this program generates displays of radar sites, weapons centers, air bases, and target areas within the subsector. This program should operate only once to place on the display drum fields information necessary to produce the required geographic displays. The situation display geography program would need to be

repeated only if subsector boundaries or the locations of specified units were changed.

The miscellaneous situation display program makes up all other situation displays including special expanded displays and track history displays for the identification function within air surveillance.

2.1.2.4 Digital Display Program Group

~~The digital display program group prepares, directs, and changes all digital displays as required by other programs.~~ Requested digital displays are made up by the programs in this group; the requests are interpreted by the keyboard message interpretation program and presented to this program group for implementation of the requests. When alarm digital displays are required, the digital display program group also changes the appropriate bits in the core memory image of the warning light registers to cause a warning light indication to accompany the alarm digital display. The digital display program group includes the summary digital display program, which makes up all normally present summary digital displays, and the air surveillance, the weapons direction, and the miscellaneous digital display programs, which make up requested displays for the appropriate functions.

2.1.2.5 Input-Output Program Group

The input-output program group controls all input and output operations not specifically handled by other programs. The card input program processes all computer entry punch data which it reads from the manual data input field of the Drum System, checks the received data for consistency, and modifies the appropriate data tables in accordance with the received data. The crosstell (XTL) input program reads messages from the XTL drum field, extrapolating the position information within these messages over the time between their transmission and their activation in the receiving central, and modifying data tables to include this extrapolated information. The output makeup program prepares messages for transmission via the ground-to-air (G/A), ground-to-ground (G/G), and teletype (TTY) data links of the Output System. The information to be transmitted in G/A messages is furnished completely by another program; for G/G and TTY, the output makeup program is referred to data tables by those programs requesting the transmission of messages over these two links. The output makeup program may be split up into routines which can follow those programs requesting transmission of output messages to reduce the delay between the request and the transmission of messages.

One program of the input-output program group, the XTL output program, requests transmission of output messages. This program calculates the priorities for transmission and performs the bookkeeping involved in crosstelling to antiaircraft operations centers and to ad-

jacent centrals. In effect, the XTL output program receives requests for crosstelling from other programs, prepares a list in order of priority of tracks to be crosstold, and turns over the list to the output makeup program.

The recording and intercommunication program checks through all data tables for information to be recorded on magnetic tape for later analysis. (This tape-recorded information can likewise be used by the standby computer during start-over to supplement its stored data received via intercommunication.) This program also selects the information to be presented via the intercommunication field to the standby computer for its safe data storage. The actual transfers are controlled by the sequence control program.

2.1.2.6 Central Bookkeeping Program Group

The central bookkeeping program group performs much of the miscellaneous bookkeeping and service functions needed but not performed by other programs. The communications and bookkeeping program maintains summary counts of types of tracks, number of actions, etc., performs those tables changes which result from an automatic function rather than from a keyboard action and which are not made by the programs executing the automatic functions, and senses for certain types of alarm conditions whose detection causes this program to inform other programs that alarms are required at particular consoles. Some of the other functions performed by this program include: checking for conditions which lead to the dropping of a track (insufficient radar data on a track identified as friendly, for example); processing of tracks being dropped, which includes requesting special displays for those tracks being dropped; changing of summary counts and removal of data on those tracks from data tables; and calculation of speed and heading from rectangular velocity components.

The track sort program, another program of the central bookkeeping group, sorts tracks and uncorrelated radar data messages into geographical blocks to simplify the task of the correlation program (an operational group program) and checks for trouble tracks (crossing tracks or tracks leaving the subsector).

The new track processing program performs the necessary processing for newly initiated tracks. Its functions include the assignment of track numbers, initial identification as friendly or pending, priority for height determination (used in identification of the target), changing track summary counts, and notifying the appropriate display programs of the existence of new tracks to be displayed.

2.1.3 Operational Program Group

2.1.3.1 Tracking Program Group

The programs within the tracking group perform all functions necessary to the maintenance of a picture

of air movements within the subsector. The radar input program reads all incoming radar data from the Drum System, converts each radar report from polar co-ordinate form to rectangular co-ordinates based on the subsector reference point, masks out returns from radar sets which would provide greater than triple coverage, and approximates actual ground range from the slant-range measurements provided by radar.

The correlation program compares all data processed by the radar input program with tracks in the subsector in order to associate particular radar reports with these tracks and so obtain new reports on the positions of the targets represented by the tracks. Those radar reports which compare with tracks are considered correlated and are used to correct track positions and velocities. Those radar reports which do not compare with tracks are considered uncorrelated and may be used to initiate new tracks. Both types of radar reports are prepared by the correlation program for placement on the radar data message fields for delivery to the Display System.

The smooth and predict program analyzes correlated radar data for use in correcting track messages. Since there will usually be more than one radar report correlating with a single track, all the reports for a single track are examined by this program. If these reports are reasonably consistent (fall within a small area around the predicted position of the track), the program combines them into a weighted single report on the target and corrects the position and velocity of the track message accordingly. (The velocity of the track effectively defines the position predicted for the target when the next set of radar reports is analyzed.) If the radar reports which correlate with a given track are not consistent, these reports are turned over to the smoothing and trouble detection program.

The smoothing and trouble detection program attempts to do for tracks with inconsistent data what is done for other tracks by the smooth and predict program. Inconsistent radar reports may be caused by a target's changing its course or by a the splitting of a multiple target (a formation of aircraft) into two or more separate targets or formation. In the latter case, the smoothing and trouble detection program notifies the display programs of a trouble situation requiring monitoring at a display console. If a turn is detected, the smoothing constants for the turning track are changed to allow normal correlation without producing a trouble situation.

The automatic initiation program receives uncorrelated radar data messages, either those detected by the area discriminator or those referred to the program by keyboard and light gun actions. These uncorrelated radar messages are held until the next series of radar

reports is processed; correlation of the reports with the held uncorrelated reports of the previous series is attempted in this series. If this correlation is successful, a new track is indicated to the central bookkeeping program group for processing. This process of attempting correlation of new radar reports with previously uncorrelated reports is continuous, ensuring the generation of a track as soon as sufficient data on a target within the subsector is acquired.

The radar analysis program analyzes the data received from each radar set for signs of excessive noise, jamming, or equipment malfunction and notifies radar supervisory personnel of radar data quality from each set via the Warning Light System. Those radar sets which are defective or whose data quality is low have their status indicated by this program; in addition, this program prevents the data from these sets from being used in various automatic functions in the central.

2.1.3.2 Weapons Direction Program Group

The programs within the weapons direction group perform all functions necessary to the implementation of the tactical decisions made by operators at display consoles. The weapons assignment program calculates intercept points and times to go to interception for those tracks referred to it by the keyboard message interpretation programs or by the communications and bookkeeping program, selecting those air bases from which interceptors would have the shortest time to go to interception and notifying the display programs of the information to be displayed to weapons assignment personnel. The interception program calculates vectoring commands for interceptors (transmitted via G/A links) and intercept points for display to intercept direction personnel. This program also communicates, to the situation display program, any alarm conditions such as program-generated changes in intercept tactics or operator-requested interceptions which are impossible to perform as requested.

2.1.3.3 Miscellaneous Program Group

The miscellaneous program group within the operational program group performs certain operations for such functions as height determination, identification, raid-forming, and simulation (used for training purposes). The height priority program calculates the order in which height determinations shall be made or repeated on the tracks within the subsector. The results of this program are communicated to the output makeup program for preparation of height-request messages for G/G transmission. The height input program processes height-reply messages and modifies the appropriate data tables on the basis of the height information received.

The identification monitoring program extrapolates air movement data (flight plans filed by commercial

and military aircraft) for continuous use in correlation of radar data. Once a track has been established (initiated from air movement data and correlated radar data), this program checks for deviations of the track from its extrapolated air movement data position. If the track is too far from the extrapolated air movement data position, the program causes an alarm display to be generated. The identification monitoring program also performs other processing of air movement data, such as removal of terminating data (flight plans extrapolated beyond the subsector boundaries); in this situation, the program notifies the communication and book-keeping program that the data or extrapolated track is to be dropped.

The raid-forming program performs the operations necessary to present a summary or raid display of a number of hostile or unknown tracks designated by keyboard and light gun actions for grouping into a raid. These operations include the preparation of a raid box (a vector and tabular track message combined to describe the area, heading, and speed of the raid), extrapolation of existing raid boxes or group boxes (a group contains a number of interceptors grouped in the same manner as hostiles are grouped into a raid), checking for correlation of individual tracks with their raid or group, and correcting the raid or group box limits, if necessary.

The simulation program generates simulated radar data (for use in training operations), starting from a location and proceeding in a direction specified by keyboard messages. This program also prepares for simulated tracks based on simulated radar data and supplies simulated height replies to height requests on simulated tracks. Any other data which must be simulated to complete the information on simulated tracks is also supplied by the simulation program.

2.2 PROGRAM TIMING

2.2.1 General

In order to equalize program operation in either heavy or light air traffic, the operational active program is synchronized with Display System operation. Normally, the interval between successive synchroniza-

tions, known as frame synchronization time, is six display cycles, or 15.2 seconds. Under heavy load conditions, this interval can be lengthened.

Most of the programs in the operational active program are performed at least once within the interval of one frame synchronization time. When all these programs which were scheduled for performance have been executed, the computer is said to have completed one program frame, and the time required to complete all these programs is called the frame operating time. Since frame operating time may not match frame synchronization time, time-buffering is performed to start frame operating time in synchronism with frame synchronization time. If the frame operating time is less than the frame synchronization time, the program waits until synchronization is performed before starting the next frame. If operating time is greater than synchronization time, the synchronization time is extended for an extra display cycle, or to 17.8 seconds. (See fig. 2-1.)

Since no program operates more often than three times per frame, the frame is divided into three approximately equal parts, called subframes. Again, time-buffering is used to maintain subframe time between the limits of 4.5 and 6 seconds. Those programs which read the various input drum fields are scheduled once each subframe to prevent the loss of incoming data because of lack of storage space on these fields. Further, other programs are divided to operate once per subframe, to avoid premature completion of all programs in the frame under light load conditions, followed by a period of waiting until the next frame begins.

2.2.2 Program Classes

The programs which make up the operational active program are divided into three classes, A, B, and C, in accordance with their frequency of operation within a program frame. The A class programs are those which are operated once every subframe (three times per program frame). The sequence of their execution within a subframe is established in advance by the functions they perform and their relations to each other. The B class programs are those which are operated once per program frame. The order in which B class programs are executed is variable; a program of this

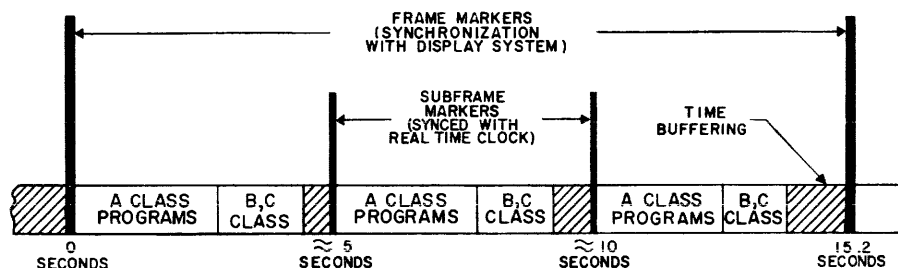


Figure 2-1. Program Frame Organization and Timing

class is designed to operate in any subframe, before or after any other B class program, or between any A class program. The C class programs are those programs which operate less than once per program frame. They are designed, as are B class programs, for operation between any other programs and anywhere with a program frame. However, C class programs are repeated only once per integral number of program frames. The various programs within the operational active program, their class, and their approximate length are given in table 2-1.

2.2.3 Program Sequence

Within a given program subframe, A class programs are performed in the following order:

- a. Simulation
- b. Crosstelling input
- c. Height input
- d. Keyboard message interpretation
- e. Radar input
- f. Correlation

TABLE 2-1. OPERATIONAL ACTIVE PROGRAMS

PROGRAM	CLASS	LENGTH (INSTRUCTIONS)	PROGRAM	CLASS	LENGTH (INSTRUCTIONS)
Sequence Control Program	Special	500	Recording and Intercommu- nication	A, B, C	2,000
Keyboard Message (Switch) Interpretation Group:			Central Bookkeeping Group:		
Air Surveillance Message Interpretation	A	4,000	Communications and Book- keeping	A, B	3,000
Weapons Direction Message Interpretation	A	4,000	New Track Processing	A	500
Command Post Message Interpretation	A	500	Track Sort	B	500
Miscellaneous	A	3,000	Tracking Group:		
Situation Display Makeup Group:			Radar Input	A	1,500
Track Display	A, B	3,000	Correlation	A	2,500
Air Surveillance SD	A	2,000	Smooth and Predict	A	500
Weapons Direction SD	A	2,000	Smoothing and Trouble De- tection	B	4,000
Miscellaneous SD	A, B, C	3,000	Automatic Initiation	B	500
SD Geography	C*	1,000*	Radar Analysis	B	1,000
Digital Display Makeup Group:			Weapons Direction Group:		
Air Surveillance DD	A	1,000	Interception	B	2,000
Weapons Direction DD	A	2,000	Weapons Assignment	A	1,500
Miscellaneous DD	A	2,000	Miscellaneous Group:		
Summary DD	A	2,000	Height Priorities	B	1,000
Input-Output Group:			Height Input	A	500
Card Input	B	2,000	Identification Monitoring	B	1,000
Crosstell Input	A	1,000	Raid Forming	C	1,000
Crosstell Output	A, B	2,000	Simulation	A	2,000
Output Makeup	A, B	1,000	Total		60,000*

*The SD geography program is run only once unless changes are required; its length is not included in the total program length.

- g. Smooth and predict
- h. New track processing
- i. Communication and bookkeeping
- j. Crosstell output
- k. Output makeup
- l. Weapons assignment
- m. Situation display makeup
- n. Digital display makeup
- o. Recording and intercommunication.

As stated previously, the sequence of execution of B and C class programs is arbitrarily chosen for convenience and may vary from frame to frame.

2.2.4 Effect of Program Sequence on Response Times

The fact that the radar inputs program is an A class program indicates that the various radar input fields (LRI and GFI) are read into the Central Computer System once every 5 seconds. With this rate of reading, the drum fields will not become filled; if these drum fields were filled, new radar data would be lost.

The position of a track data message as displayed at SD consoles is changed once each frame although new positions and velocities are calculated for each track every subframe. The displays are not changed every 5 seconds since to do so would blur them. Similarly, since a radar data drum field is held for the writing of new radar data messages for an entire program frame, RD message displays change only once every 15 seconds. Thus, the RD present messages (displayed brightly in contrast to the dim RD past messages) being displayed at any one time show data collected during the last program frame; they are a maximum of 20 seconds old when first displayed.

The keyboard message interpretation programs and most display makeup programs are A class programs, giving a response time for operator display requests of from 5 to 10 seconds. Normally, the manual input matrix is read every 2.5 seconds (although the memory images of the matrix are interpreted only once every 5 seconds). Under heavy load conditions, reading frequency may drop to once every 3 seconds or slower. Therefore, an operator must leave a keyboard message

set up for at least 3 seconds after completing the message (by ACTION pushbutton or light gun action) to ensure correct reading of the message.

Successful correlation of radar data to establish a new track requires radar data in several successive program frames. Therefore, the time taken between the initial detection of a new target and the automatic initiation of a track on this target varies between 1 and 2 minutes. (This time may be shortened if the radar reports correlate with air movement data such as flight plans, early-warning data, GOC reports, etc.) Once a new track is established, it is presented to the identification section in from 5 to 10 seconds. If the track is identified as hostile, unknown, or faker (thus requiring defensive action), it is referred to the senior director or senior weapons director in from 5 to 10 seconds.

Certain program functions allow synchronized display of information to various operators at display consoles. A digital display cycle is initiated each subframe immediately after operation of the digital display makeup programs. At almost the same time, the core memory image of the warning lights registers is written into the Warning Light System, thus synchronizing alarm displays on DD tubes with the lighting of alarm neons and the sounding of audible alarms. Similarly, the *Operate* (area discriminator) instruction is given just after the release of a new RD field for display, ensuring that the automatic initiation program obtains the latest RD present messages for initiation of tracks.

If heavy load conditions necessitate the lengthening of program frame time, those programs which assume a constant time (one frame or subframe) since their last operation in performing their calculations (for example, calculating velocities in miles per subframe by subtracting positions in the last subframe from present position) will introduce errors of magnitudes proportional to the lengthening of frame time. This type of error can be eliminated by referring to the sequence control program for correction factors or by making the programs which use approximate times more complex. Neither possibility is too desirable since it lengthens program execution time. Further, corrections of time errors may not be necessary, since the calculations are repeated often enough to prevent accumulation of errors over a long period of time.

SECTION 3

OPERATIONAL STANDBY PROGRAM

3.1 INTRODUCTION

The operational standby program, unlike the operational active program, performs two functions, air defense and maintenance. The requirements of these two functions conflict with each other in several ways, forcing a compromise in the operational standby program which can only partially satisfy both requirements and which varies with the relative importance of the two requirements at any given time.

3.2 AIR DEFENSE REQUIREMENTS

3.2.1 General

The functions of the standby computer needed to meet air defense requirements can be divided into four categories:

- a. Startover
- b. Safe data storage
- c. Active computer monitoring
- d. Standby computer monitoring.

3.2.2 Startover

The startover function is performed by the standby computer as it is switched to active status either at a scheduled time or as a result of malfunctions in the active computer. The startover program utilizes summary air defense data supplied by the active computer, extrapolating this data over the interval between deactivation of the active computer and activation of the standby computer, allowing the now active computer to pick up the air defense function smoothly and quickly. (An active computer may require startover without a switching operation if it is shut down and then turned on again without activating the standby computer.)

3.2.3 Safe Data Storage

Smooth startover of the standby computer requires the maintenance of stored summary air defense data. The active computer must periodically select data and supply it to the standby computer via the intercommunication field of the active Drum System. The standby computer must ready the data, check it for reliability, and then store it. Since the standby computer must read the drum field before the active computer writes new data on it, the operation of the standby computer is of necessity synchronized with the active computer. The degree of limitation on standby computer activities imposed by this synchronization is dependent upon the

volume and rate of transfer of summary air defense data. The limitation can be relaxed somewhat if summary air defense data is maintained by the active computer on magnetic tape. In the event of a switchover, the tape can be transferred to a tape drive unit of the now active computer and used by it for startover.

3.2.4 Active Computer Monitoring

The standby computer can perform some monitoring of active computer operation to ensure more rapid detection of malfunctions in the active computer. Certain alarm conditions in one computer are communicated to the other computer for this purpose. In addition, the summary air defense data supplied by the active computer must be checked for self-consistency before it is used in safe data storage. Inconsistencies in this data can be presented by the standby computer as indications of malfunctions in the active computer.

The standby computer may be used in long-term analysis of active computer operation, since the task of post-problem analysis of data (reduction of data recorded during the execution of air defense operations) is performed by the standby computer. The point-by-point monitoring of the active computer by the standby computer in ways other than the ones just described will be minimized because the standby computer will not always be functioning in support of the active computer.

3.2.5 Standby Computer Monitoring

The major responsibility of the standby computer is the monitoring and maintenance of its own reliability.

3.3 MAINTENANCE REQUIREMENTS

3.3.1 General

The standby computer is used in maintaining itself and in maintaining equipment external to the central. In general, maintenance of the equipment within the central is done by reliability and diagnostic programs. Once the Central Computer System of the standby computer is found to be reliable, it is used to test the other systems within the standby computer. In addition, simplex equipment is tested by the standby computer by switching the simplex units to be tested into standby status. Finally, the standby computer can be used to test the equipment at P sites, for example, by routing the signals from the P sites to the standby computer. While some of the maintenance routines may be interleaved

with the routines used by the standby computer in performing its air defense functions, many others (including those which may destroy information stored on drums) cannot be performed along with air defense routines. Therefore, the standby computer must be assigned at least two different modes of operation, one mode in which it performs air defense functions and another mode in which no air defense functions can be performed.

3.3.2 Standby Modes of Operation

3.3.2.1 Duplex Mode

In the duplex mode of operation, the standby computer is ready to assume the air defense load after switchover. In this mode, air defense and maintenance programs are interleaved, and the standby computer is synchronized with the active computer. Both the maintenance and the air defense programs operated in this mode are short enough to permit interleaving.

3.3.2.2 Simplex Mode

When the standby computer is either inoperative and undergoing corrective maintenance or executing maintenance programs which are destructive or do not allow interleaving with air defense programs, the standby computer is in simplex mode. If switchover is necessary when the standby computer is in simplex mode, the startover operation will be much longer and much less smooth than a switchover starting with the standby computer in duplex mode.

3.3.3 Operational Standby Program, Duplex Mode

3.3.3.1 Sequence Control Program

A sequence control program (not the same as that used in the operational active program) is used to control the interleaving of operational and maintenance programs in the standby computer. The standby sequence control program maintains the necessary synchronization between the standby and active computers

for the purposes of intercommunication of summary air defense data.

3.3.3.2 Duplex Switching Console Programs

Included within the operational standby program are programs which make up digital displays and which interpret keyboard messages for the duplex switching console. (Refer to Part 1, Ch 9.) The displays presented on the standby half of this console give indications of the reliability of the standby computer and of its readiness to assume active status should switchover be necessary.

3.3.3.3 Active Computer Monitoring Program

This program monitors the alarm indications of the other computer and checks the consistency of the summary air defense data supplied by the active computer.

3.3.3.4 Safe Data Storage Program

The data passed by the active computer monitoring program is taken by the safe data storage program and stored in appropriate data tables on the auxiliary memory drums (A or B).

3.3.3.5 Startover Program

The startover program is activated only if the standby computer is switched to active. However, this program must be included within the operational standby program. When activated, the startover program uses the tables prepared and maintained by the safe data storage program to extrapolate an up-to-date picture of air movements within the subsector. The operational active program can then take over, to maintain and correct this extrapolated picture. The now active computer thus assumes the active air defense function with the least possible time loss for switchover. One of the first operations of the startover program is the loading and initiation of the active sequence control program (if this program is not already in core memory during standby operation).

CHAPTER 2

PROGRAMMING DATA

SECTION 1

INTRODUCTION

1.1 GENERAL

This chapter presents data necessary for an understanding of the action performed by each instruction. The information provided includes:

- a. All the codes used in instructions
- b. Descriptions of all instructions, both legal and illegal
- c. Descriptions of instruction sequencing, arithmetic indexing, and IO processes, including the effects of illegal actions
- d. Rules for programming, including special rules and restrictions caused by difficulties with logic or timing.

Table 2-2 lists all legal instructions by class with their mnemonic and octonary codes. The octonary code corresponds to three adjacent octonary divisions of the instruction word. The first octonary digit corresponds to bit positions L4 through L6, the second to positions L7 through L9, and the third, if present, to positions L10 through L12, of which only L10 is properly part of the operation code.

1.2 NUMBERS AND ARITHMETIC

1.2.1 Representation of Numbers

A number in the Central Computer System can be considered in several forms:

- a. As a binary fraction of 15 or 31 bits prefixed by a sign bit
- b. As a binary fraction of 16 bits with the sign understood
- c. As a binary integer, usually of 16 bits, with the sign understood.

The arithmetic circuits of the Central Computer System treat 32-bit words as pairs of binary fractions. Normally, a number (half-word) consists of a sign bit and a binary fraction of 15 numerical bits. Its range is limited to between $-1+2^{-15}$ and $+1-2^{-15}$ inclusive, and it is expressed to the nearest multiple of 2^{-15} . A sign bit of 0 indicates a positive number whose magnitude is the binary fraction expressed by the 15 remaining

TABLE 2-2. INSTRUCTION CODES

CLASS	INSTRUCTION NAME	OPERATION CODE	
		MNEMONIC	OCTONARY
Miscellaneous	<i>Program Stop</i>	HLT	000
	<i>Extract</i>	ETR	004
	<i>Operate</i>	PER	01-
	<i>Clear and Subtract Word Counter</i>	CSW	020
	<i>Shift Left and Round</i>	SLR	Q24
	<i>Load B Register</i>	LDB	030
Add	<i>Clear and Add</i>	CAD	100
	<i>Add</i>	ADD	104
	<i>Twin and Add</i>	TAD	110
	<i>Add B Registers to Accumulators</i>	ADB	114
	<i>Clear and Subtract</i>	CSU	130
	<i>Subtract</i>	SUB	134
	<i>Twin and Subtract</i>	TSU	140
	<i>Clear and Add Magnitude</i>	CAM	160
Multiply	<i>Difference Magnitudes</i>	DIM	164
	<i>Multiply</i>	MUL	250
	<i>Twin and Multiply</i>	TMU	254
	<i>Divide</i>	DVD	260
	<i>Twin and Divide</i>	TDV	264
Store	<i>Store</i>	FST	324
	<i>Left Store</i>	LST	330
	<i>Right Store</i>	RST	334
	<i>Store Address</i>	STA	340
	<i>Add One</i>	AOR	344
	<i>Exchange</i>	ECH	350
	<i>Deposit</i>	DEP	360

TABLE 2–2. INSTRUCTION CODES (cont'd)

CLASS	INSTRUCTION NAME	OPERATION CODE	
		MNEMONIC	OCTONARY
Shift	<i>Shift Left</i>	DSL	400
	<i>Shift Right</i>	DSR	404
	<i>Shift Accumulators Left</i>	ASL	420
	<i>Shift Accumulators Right</i>	ASR	424
	<i>Left Element Shift Right</i>	LSR	440
	<i>Right Element Shift Right</i>	RSR	444
	<i>Cycle Left</i>	DCL	460
	<i>Cycle Accumulators Left</i>	FCL	470
Branch	<i>Branch and Index Sense</i>	BPX	51-
		BSN	52-
	<i>Branch on Zero</i>	BFZ	540
	<i>Branch on Minus</i>	BFM	544
	<i>Branch on Left Minus</i>	BLM	550
	<i>Branch on Right Minus</i>	BRM	554
IO	<i>Load Input-Output Address Counter</i>	LDC	600
	<i>Select Drum</i>	SDR	61-
	<i>Select</i>	SEL	62-
	<i>Read</i>	RDS	670
	<i>Write</i>	WRT	674
Reset	<i>Reset Index Register</i>	XIN	754
	<i>Reset Index Register from Right Accumulator</i>	XAC	764
	<i>Add Index Register</i>	ADX	770

bits. A negative number is simply the 1's complement (including sign) of the corresponding positive number.

A product or dividend is expressed by a sign bit followed by a 31-bit fraction (following the usual rule of signs for multiplication). A quotient is initially a 16-bit binary fraction whose sign bit is in another register (accumulator).

Numbers in counters and address values in other registers are thought of as binary integers (usually of 16 bits) with a positive sign understood. However, if arithmetic processes are performed on them, they are treated as binary fractions.

The method chosen for representing numbers results in two fractional representations for zero: $+0$ (binary 0.000 000 000 000) and -0 (binary 1.111 111 111 111). These two representations are numerically equivalent, but certain manipulations such as indexing or branching may require that the programmer know which form a zero has taken.

On the other hand, the integer representation of zero has only one form (binary 0.000 000 000 000). A word consisting of 16 binary 1's taken as a fraction is -0 , but as an integer it is a large positive number (65,535 decimal) with the sign understood.

1.2.2 Addition and Subtraction

Addition or subtraction (the addition of the 1's complement) is carried out by straightforward addition of two binary fractions (including the sign bit.) Any carryout of the sign bit position is added to the least significant bit. For this method of addition-subtraction, a value of zero resulting from a legal addition or subtraction is usually in the form of -0 . The two exceptions are $(+0) + (+0)$ and $(-0) - (-0)$.

1.2.3 Overflow

Any illegal addition or subtraction in the arithmetic element (one which would yield a result not in the range $-1+2^{-15}$ to $+1-2^{-15}$ inclusive) results in the generation of an overflow condition. The operations which can generate overflows are *SLR*, *ADD*, *TAD*, *ADB*, *SUB*, *TSU*, and *AOR*. Any overflow in the left or right accumulators unconditionally sets the corresponding overflow indicator whose setting can be used to cause a response to an overflow by a *Sense* instruction. The response of the Central Computer System to an overflow condition can also be determined jointly by the instruction which caused the overflow and by the setting of certain switches on the duplex maintenance console. An overflow may or may not result in an alarm. The overflow alarm control bits in L14 through L15 of the instruction word determine which accumulator can generate overflows. See table 2–3.

TABLE 2–3. OVERFLOW ALARM CONTROL

OCTONARY CODE (L14, L15)	ACCUMULATORS WHICH CAN GENERATE OVERFLOW ALARMS
0	Neither Accumulator
1	Right Accumulator Only
2	Left Accumulator Only
3	Either Left or Right Accumulator

The action taken when an overflow alarm occurs is then determined by the settings of the OVERFLOW (inactive)-ACTIVE switch and the STOP-BRANCH switch on the duplex maintenance console.

TABLE 2-4. USEFUL ILLEGAL DIVISIONS

DIVIDEND		DIVISOR (REGISTER \times)	REMAINDER (ACCUMULATOR)	QUOTIENT (B REGISTER)
(ACCUMULATOR)	(B REGISTER)			
+A	+O	+A	0.77777	1.77777
+A	+O	-A	1.00000	0.00000
+A	B	+	B	Complement of +A

1.2.4 Division

A division operation, to be legal, must result in a quotient of magnitude less than one. However, there is no circuit which ensures that a division operation will be a legal one, nor is there any indication of an illegal division. It is therefore the responsibility of the programmer to avoid unintentional illegal divisions. However, certain illegal division operations may occasionally be useful, as shown in table 2-4.

The remainder in a division process is multiplied by 2^{16} and left in the accumulators.

1.3 SEQUENCE OF INSTRUCTION EXECUTION

1.3.1 Program Counter

The memory address in the program counter at any moment is the one from which the next instruction is normally taken and executed. Immediately after an address has been transferred from the program counter to the address counter, the contents of the program counter are increased by 1. This forms the new address from which the succeeding instruction is usually taken. However, an active *Branch* instruction will substitute, in the program counter, a new address (the right half-word of the *Branch* instruction) which becomes the address of the next instruction. A branch initiated by an alarm condition causes an automatic branch to register 20010₈ in test memory.

The program counter is a 16-bit register; however, only the right 13 bits (R3 through R15) are connected as a counter. The counter therefore counts modulo 8,192₁₀, or 20000₈. If a *Branch* instruction is not present, the program cannot continue from one type of internal memory to another. Specifically, the execution of an instruction (not a *Branch* instruction) at 17777₈ will be followed by the instruction in register 0; the instruction at 37777₈ (interpreted as 20017₈) is followed by the one at 20000₈.

1.3.2 Indexing

The process of indexing (modifying) an instruction consists of adding to the right half-word the con-

tents of the index register designated by the contents of L1 through L3 of the instruction word. This is done in an adder which operates in the same manner as the arithmetic element adders. Although an overflow can occur in the indexing process, no alarm or other overflow indication is generated. With the exception of *Branch* instructions, all instructions whose right half-words refer to memory addresses are indexible. In addition, three other instructions are indexable: *SDR*, *RDS*, and *WRT*.

There are eight possible numbers for index register selection (0 through 7). Numbers 1, 2, 4, and 5 correspond to physical index registers, while number 3 corresponds to the right accumulator. Numbers 0, 6, and 7 specify nonexistent registers which behave, as far as the process of indexing instructions is concerned, as if they each contained +0. The contents of the physical index registers (1, 2, 4, and 5) can be set to specific values or reduced by specific decrements through the use of the *XIN*, *XAC*, and *BPX* instructions.

An index register whose contents have been reduced to zero through the use of the *BPX* instruction contains -0. If this value is used to modify an instruction whose right half-word is zero, the result in the right half-word is -0, which is not interpreted in the same manner as +0. Memory address -0 is interpreted as the clock register; drum address -0 is interpreted as drum address 3777₈; and *RDS* or *WRT* -0 call for a transfer of 65,535₁₀ words.

1.4 INTERNAL MEMORY ADDRESSING

1.4.1 General

When the right half-word of an instruction is interpreted as a memory address, the address applies to one of the Central Computer System internal memory registers, as shown in table 2-5.

In addition to the normal addresses, any other address expressible with the 16 bits of the right half-word selects some register in one of the memory units, as shown in table 2-6.

TABLE 2–5. SELECTION OF INTERNAL MEMORY REGISTERS

MEMORY UNIT	NUMBER OF REGISTERS	NORMAL ADDRESSES
Core Memory (2 units)	8,192 ₁₀	0 through 17777 ₈
Test Memory	16 ₁₀	20000 through 20017 ₈
Clock Register	1	60000 ₈

TABLE 2–6. SELECTION OF INTERNAL MEMORY UNITS

CONTENTS OF R1 THROUGH R3 (OCTONARY)	MEMORY UNIT SELECTED	BITS SPECIFYING REGISTER IN SELECTED UNIT
0 or 4	Core Memory No. 1	R4 through R15
1 or 5	Core Memory No. 2	R4 through R15
2 or 3	Test Memory	R12 through R15
6 or 7	Clock Register	None

The selections for core memory are valid only if the core memory assignment switch is set to NORMAL or if the Central Computer System is in the operate mode. If the Central Computer System is in the test mode and the core memory assignment switch is on REVERSED, the roles of unit 1 and unit 2 are interchanged. In all cases, the contents of RS have no significance in a memory address.

1.4.2 Core Memory

Core memory units 1 and 2 each consist of 4,096 registers, making available a total of 8,192 registers. Each core memory register has provisions for 32 bits plus a parity bit. The addressing of the registers in the two units is successive, ranging from 0 to 17777₈.

When the core memory assignment switch is on NORMAL, registers 0 through 7777₈ are in core memory unit 1, and 10000 through 17777₈ are in unit 2. When the switch is on REVERSED and the TEST-OPERATE switch is on TEST, the roles of units 1 and 2 are reversed; i.e., the registers are renumbered 10000 through 17777₈ in unit 1 and 0 through 7777₈ in unit 2.

1.4.3 Test Memory

Test memory consists of 16 plugboard registers, 2 toggle switch registers, and a flip-flop register (test register), of which only 16 can be used at any time. Each register consists of 32 bits with no provisions for a parity bit.

The plugboard contains four assignment hubs associated with each test memory address. The hub labels and their meanings are as follows:

- P designates a plugboard register.
- A designates toggle switch register A.
- B designates toggle switch register B.
- L designates the test register.

When a hub is plugged, the designated unit is connected to the associated test memory address (provided the test memory assignment switch is on UNASSIGNED). If more than one hub is plugged at a given address, the Central Computer System interprets that address as containing a word which is the logical sum of the words in the registers connected to that address. The plugging of these hubs determines only which test memory registers are to be read at a given address. Any instruction in the store class which selects any address in test memory, or a block transfer into test memory, writes in the test register whether or not the test register is assigned to any test memory addresses.

When the test memory assignment switch is on ASSIGNED, the plugging of hubs for test memory addresses 20000, 20001, and 20017₈ are superseded by the following assignments:

- 20000₈ designates toggle switch register A.
- 20001₈ designates toggle switch register B.
- 20017₈ designates the test register.

This assignment does not affect the assignments of these or any other registers to the remaining test memory addresses 20002 through 20016₈.

The contents of the test register specify a number of parameters in connection with program-controlled marginal checking. These parameters include:

- Marginal checking group selection
- Voltage selection
- Circuit group selection
- Line selection
- Excursion control
- Polarity of excursion
- Magnitude of excursion
- Duration of excursion.

To complete an interlocking circuit permitting a computer to be switched to active status, a test memory plugboard must have a jumper between the rightmost hubs of the top and bottom rows. A plugboard of this

type is called an operational plugboard and contains operational information rather than test information.

1.4.4 Clock Register

The clock register is a 16-bit counter whose contents are increased by 1 every 1/32 second (31.25 milliseconds) while computer power is on. End-carries produced by the highest order bit position (RS) of the clock register occur every 2,048 seconds (34.133+ minutes) of continuous counting but are not used.

When an instruction calls for an operation involving the clock register, the contents of the clock register

are transferred to the right memory buffer register. The left memory buffer register then contains +0. Writing in the clock register, whose address is 60000₈, has no effect, since no provisions exist for the transfer of information into the clock register. Similarly, the contents of the clock register cannot be transferred to the operation and address register. Therefore, if the program executes a branch of control to the clock register, it is interpreted as an *HLT* instruction. An IO transfer where the *LDC* instruction specifies the clock register results in the transfer of words of +0 to the specified IO unit.

SECTION 2 INSTRUCTIONS

2.1 GENERAL

This section contains a brief statement of the action taken by the execution of each instruction (legal and illegal). These actions are also summarized in table 2-7. The contents of the registers affected by the instruction are described; those registers not specified are unchanged by that instruction. The execution times of all the instructions described are based on a 6.0-microsecond memory cycle. In each case where the exact time is indeterminate, a minimum and a maximum time are given.

The characteristics of each instruction are also given; i.e., whether it is indexable, whether it can generate an overflow condition, or whether it is delayed in its execution if the IO interlock is on. The phrase technically indexable that characterizes some instructions means that the right half-word of the instruction is indexed, but that this indexing is meaningless since the right half-word is not used. In the description of those instructions which are indexable, the term $x + in$ refers to the original contents of the address half of an instruction (x) to which are added the contents of the specified index register (in).

2.2 MISCELLANEOUS CLASS

2.2.1 Program Stop (HLT) Instruction

This instruction, which cannot be executed unless the IO interlock is off, adds 1 to the program counter and stops Central Computer System operation. Execution time of this instruction is 12 microseconds. The operation code is 000.

2.2.2 Extract (ETR) Instruction

This instruction causes each bit position of the word (mask) in register $x + in$ which contains a 0 to set to 0 the corresponding bit in the accumulators. The original contents of register $x + in$ remain in the A register. The octonary operation code for this instruction, which is indexable, is 004. Execution time of the ETR instruction is 12 microseconds.

2.2.3 Operate (PER) Instruction

This instruction activates the unit specified by the operate unit code. (See table 2-8.) If the operate unit code specifies a nonexistent unit, no action is taken. Execution time of the PER instruction is 12 microseconds.

TABLE 2-8. OPERATE UNIT CODES

OCTONARY CODE	UNIT	ACTION
Card Machines and Tapes		
51	Printer Operate Hub No. 1	Energize
52	Printer Operate Hub No. 2	Energize
53	Printer Operate Hub No. 3	Energize
54	Printer Operate Hub No. 4	Energize
55	Printer Operate Hub No. 5	Energize
56	Printer Operate Hub No. 6	Energize
57	Printer Operate Hub No. 7	Energize
60	Printer Operate Hub No. 8	Energize
61	Printer Operate Hub No. 9	Energize
62	Printer Operate Hub No. 10	Energize
73	Punch	Set up to punch columns 17-32 of card image into columns 1-16 of card.
74	Punch	Set up to gang punch columns 1-16.
70	Selected Tape Unit	Backspace.
71	Selected Tape Unit	Rewind.
67	Selected Tape Unit	Set prepared.
72	Selected Tape Unit	Write end of file.
Displays and Associated Equipment		
17	Area Discriminator No. 1 (Spare)	Turn on for next display cycle.
20	Area Discriminator No. 2 (Track initiation)	Turn on for next display cycle.
35	Digital Display	Start at slot No. 0.
76	Scan Counter	Set to 0.
77	Scan Counter	Increase by 1.
31	Situation Display Camera	Start Mode No. 1.
32	Situation Display Camera	Start Mode No. 2.

TABLE 2-8. OPERATE UNIT CODES (cont'd)

OCTONARY CODE	UNIT	ACTION
Indicators		
01	Condition Light No. 1	Turn on.
02	Condition Light No. 2	Turn on.
03	Condition Light No. 3	Turn on.
04	Condition Light No. 4	Turn on.
10	Intercommunication Indicator No. 1	Turn on.
11	Intercommunication Indicator No. 2	Turn on.
12	Intercommunication Indicator No. 3	Turn on.
13	Intercommunication Indicator No. 4	Turn on.
Input System Testing		
63	Pattern Generator	Start GFI continuous pattern; start LRI and XTL pattern.
64	Pattern Generator, GFI Section	Connect.
65	GFI Azimuth	Initiate azimuth signals.
66	GFI Target	Initiate target signals.
Miscellaneous Controls		
75	IO Address counter	Lock at current address until IO interlock is cleared.
27	IO Interlock	Turn off.
21	Marginal Checking	Start excursion.
22	Marginal Checking	Stop duplex excursion.
23	Marginal Checking	Stop simplex excursion.

2.2.4 Clear and Subtract Word Counter (CSW) Instruction

This instruction transfers the contents of the IO word counter to the right accumulator. Execution time of the CSW instruction is 6 microseconds. The octonary operation code is 020.

2.2.5 Shift Left and Round (SLR) Instruction

This instruction shifts the number in the left accumulator and left B register and the number in the right accumulator and right B register (with the exception of the sign bits of the accumulators) n bit positions to the left. The bit positions thus made vacant are filled with the accumulator sign bits, and the bits shifted left out of bit position 1 of the accumulators are discarded. The results are rounded off to 15 bits in each accumulator.

Roundoff of a positive number is accomplished by adding 1 to the contents of bit position 15 of the accumulators if the sign bit of the corresponding B register contains a 1. Roundoff of a negative number is accomplished by subtracting 1 from the contents of bit position 15 of the accumulators if the sign bit of the corresponding B register contains a 1. Completion of this instruction leaves positive zero in the A and B registers. The octonary operation code for the SLR instruction is 024. Execution time is dependent upon the value of n . If $n = 0$, execution time is 6 microseconds. If $n = 1$, minimum execution time is 6.5 microseconds, and the maximum is 7.0 microseconds. If $n \geq 2$, the minimum execution time is $\frac{n + 12}{2}$ microseconds, and the maxi-

mum is $\frac{n + 14}{2}$ microseconds. This instruction may cause overflow.

2.2.6 Load B Register (LDB) Instruction

This instruction replaces the contents of the B register with the contents of register $x + in$. Execution time of this instruction, which is indexable, is 12 microseconds. The octonary operation code is 030.

2.3 ADD CLASS**2.3.1 Clear and Add (CAD) Instruction**

This instruction replaces the contents of the accumulators with the contents of register $x + in$ and places +0 in the A registers. Execution time of this instruction, which is indexable, is 12 microseconds. The octonary operation code is 100.

2.3.2 Add (ADD) Instruction

This instruction adds the numbers in the left and right halves of register $x + in$ to the numbers in the left and right accumulators, respectively. The results are left in the accumulators, and positive zero is placed in the A registers. Execution time of this instruction, which is indexable, is 12 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 104.

2.3.3 Twin and Add (TAD) Instruction

This instruction adds the number in the left half of register $x + in$ to the numbers in the left and right accumulators. The results are left in the accumulators, and

positive zero is placed in the A registers. Execution time of the *TAD* instruction, which is indexable, is 12 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 110.

2.3.4 Add B Registers to Accumulators (ADB) Instruction

This instruction adds the numbers in the left and right B registers to the numbers in the left and right accumulators, respectively. The results are left in the accumulators, and positive zero is placed in the A registers. Execution time of the *ADB* instruction, which is technically indexable, is 12 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 114.

2.3.5 Clear and Subtract (CSU) Instruction

This instruction replaces the contents of the accumulators with the complement of the number in register $x + in$ and places positive zero in the A registers. Execution time of the *CSU* instruction, which is indexable, is 12 microseconds. The octonary operation code is 130.

2.3.6 Subtract (SUB) Instruction

This instruction subtracts the numbers in the left and right halves of register $x + in$ from the numbers in the left and right accumulators, respectively. The results are left in the accumulators, and positive zero is placed in the A registers. Execution time of the *SUB* instruction, which is indexable, is 12 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 134.

2.3.7 Twin and Subtract (TSU) Instruction

This instruction subtracts the number in the left half of register $x + in$ from the numbers in the left and right accumulators. The results are left in the accumulators, and positive zero is placed in the A registers. Execution time of the *TSU* instruction, which is indexable, is 12 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 140.

2.3.8 Clear and Add Magnitudes (CAM) Instruction

This instruction replaces the contents of the accumulators with the positive magnitudes of the contents of the left and right halves of register $x + in$, and positive zero is placed in the A registers. Execution time of the *CAM* instruction, which is indexable, is 12 microseconds. The octonary operation code is 160.

2.3.9 Difference Magnitudes (DIM) Instruction

This instruction subtracts the positive magnitudes of the numbers in the left and right halves of register $x + in$ from the positive magnitudes of the numbers in the left and right accumulators, respectively. The results are left in the accumulators while the original contents of the accumulators are placed in the B registers. Posi-

tive zero is placed in the A registers. Execution time of the *DIM* instruction, which is indexable, is 12 microseconds. The octonary operation code is 164.

2.4 MULTIPLY CLASS

2.4.1 Multiply (MUL) Instruction

This instruction multiplies the numbers in the left and right accumulators by the numbers in the left and right halves of register $x + in$, respectively, leaving the results in the accumulators and B registers. The final contents of bit position 15 of the left and right B registers are identical to the contents of the sign bit positions of the left and right accumulators, respectively. The magnitudes of the numbers in the left and right halves of register $x + in$ are placed in the A registers. Execution time of the *MUL* instruction, which is indexable, is 17 ± 0.5 microseconds. The octonary operation code is 250.

2.4.2 Twin and Multiply (TMU) Instruction

This instruction multiplies the numbers in the left and right accumulators by the number in the left half of register $x + in$, leaving the results in the accumulators and B registers. The final contents of bit position 15 of the left and right B registers are identical to the contents of the sign bit positions of the left and right accumulators, respectively. The magnitude of the number in the left half of register $x + in$ is placed in the right A register. Execution time of the *TMU* instruction, which is indexable, is 17 ± 0.5 microseconds. The octonary operation code is 254.

2.4.3 Divide (DVD) Instruction

This instruction divides the number in the left accumulator and left B register and the number in the right accumulator and right B register by the numbers in the left and right halves of register $x + in$, respectively. The 16 magnitude bits of each quotient are left in the B registers, and the remainders (times 2^{15}) are left in the accumulators. The signs of the remainders are also the signs of their respective quotients. If, after the division is completed, bit position 15 of the quotient in the right B register contains a 1, the complement of the number in the right half of register $x + in$ remains in the right A register. If the B15 bit position contains a 0, the number (rather than its complement) remains in the right A register. Execution time of the *DVD* instruction, which is indexable, is 51 microseconds minimum and 52 microseconds maximum. The octonary operation code is 260.

2.4.4 Twin and Divide (TDV) Instruction

This instruction divides the number in the left accumulator and left B register and the number in the right accumulator and right B register by the number in the left half of register $x + in$. The 16 magnitude bits of each quotient are left in the B registers, and the re-

mainders (times 2^{15}) are left in the accumulators. The signs of the remainders are also the signs of their respective quotients. If, after the division is completed, bit position 15 of the quotient in the right B register contains a 1, the complement of the number in the left half of register $x + in$ remains in the right A register. If the B15 bit position contains a 0, the number (rather than its complement) remains in the right A register. Execution time of the *TDV* instruction, which is indexable, is 51 microseconds minimum and 52 microseconds maximum. The octonary operation code is 264.

2.5 STORE CLASS

2.5.1 Store (FST) Instruction

This instruction replaces the contents of register $x + in$ with the contents of the accumulators. The contents of the accumulators remain unchanged. Execution time of the *FST* instruction, which is indexable, is 12 microseconds. The octonary operation code is 324.

2.5.2 Left Store (LST) Instruction

This instruction replaces the left half-word of register $x + in$ with the contents of the left accumulators. The contents of the accumulators remain unchanged. The original contents of register $x + in$ remain in the A registers. Execution time of the *LST* instruction, which is indexable, is 18 microseconds. The octonary operation code is 330.

2.5.3 Right Store (RST) Instruction

This instruction replaces the right half-word of register $x + in$ with the contents of the right A register. The contents of the accumulators remain unchanged. The original contents of register $x + in$ remain in the A registers. Execution time of the *RST* instruction, which is indexable, is 18 microseconds. The octonary operation code is 334.

2.5.4 Store Address (STA) Instruction

This instruction replaces the right half-word in register $x + in$ with the contents of the right A register. Execution time of the *STA* instruction, which is indexable, is 18 microseconds. The octonary operation code is 340.

2.5.5 Add One (AOR) Instruction

This instruction adds the number 0.00001_8 to the right half-word in register $x + in$. The results are also left in the right accumulator. The original contents of register $x + in$ remain in the A registers. Execution time of the *AOR* instruction, which is indexable, is 18 microseconds. Overflow may result from execution of this instruction. The octonary operation code is 344.

2.5.6 Exchange (ECH) Instruction

This instruction exchanges the contents of the accumulators with the contents of register $x + in$. The original contents of register $x + in$ remain in the A register.

Execution time of the *ECH* instruction, which is indexable, is 18 microseconds. The octonary operation code is 350.

2.5.7 Deposit (DEP) Instruction

This instruction replaces the contents of each bit position of register $x + in$ with the contents of its corresponding bit position in the accumulator if the same bit position of the word (mask) in the B registers contains a 1. The final contents of the accumulators are identical with the new contents of register $x + in$. In effect, those bit positions of the B registers that contain a 0 cause the duplication of the corresponding bit positions of register $x + in$ in the accumulators. Those bit positions of the B registers that contain a 1 cause the duplication of the corresponding bit positions of the accumulators in register $x + in$. The A registers will contain the logical sum of the contents of the B registers and the original contents of register $x + in$. Execution time of the *DEP* instruction, which is indexable, is 18 microseconds. The octonary operation code is 360.

2.6 SHIFT CLASS

2.6.1 Shift Left (DSL) Instruction

This instruction shifts the number in the left accumulator and left B register and the number in the right accumulator and right B register (with the exception of the contents of the sign bit positions of the accumulators) n places to the left. The bit positions thus made vacant are filled with the accumulator sign bits, and the bits shifted left out of bit position 1 of the accumulators are lost. Execution time of the *DSL* instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 400.

2.6.2 Shift Right (DSR) Instruction

This instruction shifts the number in the left accumulator and left B register and the number in the right accumulator and right B register (with the exception of the contents of the sign bit positions of the accumulators) n places to the right. The bit positions thus made vacant are filled with the accumulator sign bits, and the bits shifted right out of bit position 15 of the B registers are lost. Execution time of the *DSR* instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 404.

2.6.3 Shift Accumulators Left (ASL) Instruction

This instruction shifts the numbers in the left and right accumulators (with the exception of the contents of the sign bit positions) n places to the left. The bit positions thus made vacant are filled with the accumulator sign bits, and the bits shifted left out of bit position 1 of the accumulators are lost. Execution time of the ASL instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 420.

2.6.4 Shift Accumulators Right (ASR) Instruction

This instruction shifts the numbers in the left and right accumulators (with the exception of the contents of the sign bit positions) n places to the right. The bit positions thus made vacant are filled with the accumulator sign bits, and the bits shifted right out of bit position 15 of the accumulators are lost. Execution time of the ASR instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 424.

2.6.5 Left Element Shift Right (LSR) Instruction

This instruction shifts the number in the left accumulator and left B register (with the exception of the sign bit position of the left accumulator) n places to the right. The bit positions thus made vacant are filled with the sign bit of the left accumulator, and the bits shifted right out of bit position 15 of the left B register are lost. Execution time of the LSR instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 440.

2.6.6 Right Element Shift Right (RSR) Instruction

This instruction shifts the number in the right accumulator and right B register (with the exception of the contents of the sign bit position of the right accumulator) n places to the right. The bit positions thus made vacant are filled with the sign bit of the right accumulator, and the bits shifted right out of bit position 15 of the right B register are lost. Execution time of the RSR instruction depends upon the value of n . If n is less than

7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 444.

2.6.7 Cycle Left (DCL) Instruction

This instruction shifts the entire contents of the left accumulator and left B register and the entire contents of the right accumulator and right B register n places to the left. The bits shifted left out of the sign bit position of each accumulator are reinserted into bit position 15 of the associated B register, and no bits are lost. Execution time of the DCL instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 460.

2.6.8 Cycle Accumulators Left (FCL) Instruction

This instruction shifts the entire contents of the left and right accumulators n places to the left. The bits shifted left out of the sign bit position of each accumulator are reinserted into bit position 15 of the other accumulator, and no bits are lost. Execution time of the FCL instruction depends upon the value of n . If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum execution time is $\frac{n+9}{2}$ microseconds. The octonary operation code is 470.

2.7 BRANCH CLASS

2.7.1 Branch and Index (BPX) Instruction

If the contents of the specified index register (1, 2, 4, or 5) are positive, the contents of the program counter (which contains an address one greater than the address of the BPX instruction) are transferred to the right A register, and the right half-word of the BPX instruction is loaded into the program counter. The index interval (L10 through L15) is subtracted from the contents of the specified index register, and the results are left in the index register. If the contents of the specified index register are negative, no branch is performed, and the right A register is cleared. If index register 0 or 3 is specified (unconditional branch), the contents of the program counter (which contains an address one greater than the address of the BPX instruction) are transferred to the right A register, and the right half-word of the BPX instruction is loaded into the program counter. If index register 6 or 7 is specified, no branch

takes place, and the right A register is cleared. Execution time of the *BPX* instruction is 6.0 microseconds. The octonary operation code is 51-.

2.7.2 Sense (*BSN*) Instruction

This instruction senses the unit specified by the sense code (L10 through L15). (See table 2-9.) If the condition for branching is satisfied, the contents of the program counter (which contains an address one greater

than the address of the *BSN* instruction) are transferred to the right A register, and the right half-word of the *BSN* instruction is loaded into the program counter. If the condition for branching is not satisfied, or if a non-existent unit is specified, no branch can take place, and the right A register is cleared. Execution time of the *BSN* instruction is 12 microseconds. The octonary operation code is 52-.

TABLE 2-9. SENSE CODES

OCTONARY CODE	UNIT	CONDITION FOR BRANCH	BSN TURNS UNIT OFF
Alarms			
50	Nonsearch Comparison Error Indicator (Too Many Words Tagged for 1 Burst)	On	Yes
51	Output Drum Parity Error Indicator	On	Yes
52	Illegal Output Section or Address Indicator	On	Yes
53	Defective Output Transmission Indicator	On	Yes
12	Left Overflow Indicator	On	Yes
13	Right Overflow Indicator	On	Yes
15	Memory Parity Error Indicator	On	Yes
16	Drum Parity Error Indicator	On	Yes
17	Tape Parity Error Indicator	On	Yes
33	Output Alarm Indicator	On	Yes
41	Other Computer Alarm No. 1 Indicator	On	Yes
42	Other Computer Alarm No. 2 Indicator	On	Yes
Duplex Facilities			
30	Duplex Switch	Active Position	No
43	Other Computer Intercommunication Indicator No. 1	On	Yes
44	Other Computer Intercommunication Indicator No. 2	On	Yes
45	Other Computer Intercommunication Indicator No. 3	On	Yes
46	Other Computer Intercommunication Indicator No. 4	On	Yes
IO Units and Controls			
01	Condition Light No. 1	On	Yes
02	Condition Light No. 2	On	Yes
03	Condition Light No. 3	On	Yes
04	Condition Light No. 4	On	Yes

TABLE 2-9. SENSE CODES (cont'd)

OCTONARY CODE	UNIT	CONDITION FOR BRANCH	BSN TURNS UNIT OFF
37	Display System	Displaying Track Data	No
14	IO Interlock	On	No
31	Printer Sense Hub No. 1	Energized	No
32	Printer Sense Hub No. 2	Energized	No
10	Selected Tape Unit	Not Prepared	No
11	Selected IO Unit	Not Ready	No
35	Situation Display Camera	Taking a Picture	No
Input System Testing			
34	GFI Range Signal	Initiated	Yes
47	GFI North or Azimuth Signal	Initiated	Yes
Miscellaneous			
20	Marginal Checking Voltage	In Excursion Cycle	No
21	Sense Switch No. 1	On	No
22	Sense Switch No. 2	On	No
23	Sense Switch No. 3	On	No
24	Sense Switch No. 4	On	No

2.7.3 Branch on Zero (BFZ) Instruction

This instruction examines the contents of the accumulators; if both are zero (any combination of +0 and/or -0), the contents of the program counter (which contains an address one greater than the address of the *BFZ* instruction) are transferred to the right A register, and the right half-word of the *BFZ* instruction is loaded into the program counter. If the numbers in the accumulators are not zero, the right A register is cleared. Execution time of the *BFZ* instruction is 12 microseconds. The octonary operation code is 540.

2.7.4 Branch on Minus (BFM) Instruction

This instruction examines the numbers in the accumulators; if both are negative, the contents of the program counter (which contains an address one greater than the address of the *BFM* instruction) are transferred to the right A register, and the right half-word of the *BFM* instruction is loaded into the program counter. If the numbers in the accumulators are not negative, the right A register is cleared. Execution time of the *BFM* instruction is 6 microseconds. The octonary operation code is 544.

2.7.5 Branch on Left Minus (BLM) Instruction

This instruction examines the number in the left accumulator; if it is negative, the contents of the program counter (which contains an address one greater than the address of the *BLM* instruction) are transferred to the right A register, and the right half-word of the *BLM* instruction is loaded into the program counter. If the number in the left accumulator is not negative, the right A register is cleared. Execution time of the *BLM* instruction is 6.0 microseconds. The octonary operation code is 550.

2.7.6 Branch on Right Minus (BRM) Instruction

This instruction examines the number in the right accumulator; if it is negative, the contents of the program counter (which contains an address one greater than the address of the *BRM* instruction) are transferred to the right A register, and the right half-word of the *BRM* instruction is loaded into the program counter. Execution time of the *BRM* instruction is 6.0 microseconds. The octonary operation code is 554.

2.8 IO CLASS**2.8.1 Load Input-Output Address Counter (LDC) Instruction**

The right half-word of this instruction (suitably modified if an index register is specified) is placed in the IO address counter. The right half-word contains the first internal memory address from or to which a word is transferred during the IO transfer. This instruction cannot be executed until the IO interlock is off. Execution time of the LDC instruction, which is indexable, is 6.0 microseconds. The octonary operation code is 600.

2.8.2 Select Drum (SDR) Instruction

This instruction selects the group, drum, field, and

mode for an IO transfer involving the Drum System and deselects the previously selected drum field or IO unit. (See table 210.) If a drum field using the address mode is chosen, the right half-word (R5 through R15) specifies the address of the first drum register to be used. If a field using the status identity mode is chosen, the right half-word contains the identify bits which are designated for comparison with the identity bits within a message in the selected field. This instruction cannot be executed until the IO interlock is cleared. The SDR instruction is indexable when a field is chosen which uses the address or status identity modes; it is technically indexable when a field is chosen which uses the status mode. Execution time for the SDR instruction is 12 microseconds. The octonary operation code is 61-.

TABLE 2-10. GROUP, DRUM, FIELD, AND MODE SELECTION CODES

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	02	Auxiliary Memory No. 1	Address	Read, Write
0	03	Auxiliary Memory No. 2	Address	Read, Write
0	04	Auxiliary Memory No. 3	Address	Read, Write
0	05	Auxiliary Memory No. 4	Address	Read, Write
0	06	Auxiliary Memory No. 5	Address	Read, Write
0	07	Auxiliary Memory No. 6	Address	Read, Write
0	10	Auxiliary Memory No. 7	Address	Read, Write
0	11	Auxiliary Memory No. 8	Address	Read, Write
0	12	Auxiliary Memory No. 9	Address	Read, Write
0	13	Auxiliary Memory No. 10	Address	Read, Write
0	14	Auxiliary Memory No. 11	Address	Read, Write
0	15	Auxiliary Memory No. 12	Address	Read, Write
1	41	Auxiliary Memory No. 13	Address	Read, Write
1	42	Auxiliary Memory No. 14	Address	Read, Write*
1	43	Auxiliary Memory No. 15	Address	Read, Write*
1	44	Auxiliary Memory No. 16	Address	Read, Write*
1	45	Auxiliary Memory No. 17	Address	Read, Write*
1	46	Auxiliary Memory No. 18	Address	Read, Write*
1	51	Auxiliary Memory No. 19	Address	Read, Write*
1	52	Auxiliary Memory No. 20	Address	Read, Write*
1	53	Auxiliary Memory No. 21	Address	Read, Write*
1	54	Auxiliary Memory No. 22	Address	Read, Write*

*Write only if write interlock switch on duplex maintenance console is unlocked.

TABLE 2-10. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
1	55	Auxiliary Memory No. 23	Address	Read, Write*
1	56	Auxiliary Memory No. 24	Address	Read, Write*
1	61	Auxiliary Memory No. 25	Address	Read, Write*
1	62	Auxiliary Memory No. 26	Address	Read, Write*
1	63	Auxiliary Memory No. 27	Address	Read, Write*
1	64	Auxiliary Memory No. 28	Address	Read, Write*
1	65	Auxiliary Memory No. 29	Address	Read, Write*
1	66	Auxiliary Memory No. 30	Address	Read, Write*
1	71	Auxiliary Memory No. 31	Address	Read, Write*
1	72	Auxiliary Memory No. 32	Address	Read, Write*
1	73	Auxiliary Memory No. 33	Address	Read, Write*
1	74	Auxiliary Memory No. 34	Address	Read, Write*
1	75	Auxiliary Memory No. 35	Address	Read, Write*
1	76	Auxiliary Memory No. 36	Address	Read, Write*
1	02	Auxiliary Memory No. 37	Address	Read, Write*
1	03	Auxiliary Memory No. 38	Address	Read, Write*
1	04	Auxiliary Memory No. 39	Address	Read, Write*
1	05	Auxiliary Memory No. 40	Address	Read, Write*
1	06	Auxiliary Memory No. 41	Address	Read, Write*
1	07	Auxiliary Memory No. 42	Address	Read, Write*
1	10	Auxiliary Memory No. 43	Address	Read, Write*
1	11	Auxiliary Memory No. 44	Address	Read, Write*
1	12	Auxiliary Memory No. 45	Address	Read, Write*
1	13	Auxiliary Memory No. 46	Address	Read, Write*
1	14	Auxiliary Memory No. 47	Address	Read, Write*
1	15	Auxiliary Memory No. 48	Address	Read, Write*
0	24	Crosstelling Input	Status	Read, Test Write
0	25	Crosstelling Input	Identity (R11-R15)	Read
0	40	Crosstelling Marker (Single Channel)	Address	Write (Contents of LS only)
0	27	Digital Display	Address	Read, Write
0	17	Digital Display	Identity (R14-R15)	Test Read

*Write only if write interlock switch on duplex maintenance console is unlocked.

TABLE 2-10. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	32	Gap-Filler Input	Status	Read, Test Write
0	33	Gap-Filler Input	Identity (R11-R15)	Read
0	16	Intercommunication (Other)	Address	Read
0	26	Intercommunication (Own)	Address	Read, Write
0	76	Intercommunication (Own)	Address	Test Read
0	34	Long-Range Radar Input No. 1	Status	Read, Test Write
0	35	Long-Range Radar Input No. 1	Identity (R12-R15)	Read
0	50	Long-Range Radar Input No. 1	Identity (R7-R15)	Read
0	36	Long-Range Radar Input No. 2	Status	Read, Test Write
0	37	Long-Range Radar Input No. 2	Identity (R12-R15)	Read
0	51	Long-Range Radar Input No. 2	Identity (R7-R15)	Read
0	22	Manual Input	Status	Read, Test Write
0	23	Manual Input	Identity (R14-R15)	Read
0	30	Output Buffer Odd	Status	Write
0	31	Output Buffer Even	Status	Write
0	30	Output Buffer	Identity (R14-R15)	Test Read
0	31	Output Buffer	Status	Test Read
0	60	Radar Data No. 1	Address	Read, Write
0	61	Radar Data No. 2	Address	Read, Write
0	62	Radar Data No. 3	Address	Read, Write
0	63	Radar Data No. 4	Address	Read, Write
0	64	Radar Data No. 5	Address	Read, Write
0	65	Radar Data No. 6	Address	Read, Write
0	66	Radar Data No. 7	Address	Read, Write

TABLE 2-10. GROUP, DRUM, FIELD, AND MODE SELECTION CODES (cont'd)

OCTONARY CODE R1, L10-15		DRUM FIELD	MODE	OPERATIONS
0	67	Radar Data No. 8	Address	Read, Write
0	70	Radar Data No. 9	Address	Read, Write
0	47	Situation Display	Identity (R5-R10)	Test Read
0	20	Spare No. 1	Address	Read, Write
0	21	Spare No. 2	Address	Read, Write
0	41	Track Display No. 1	Address	Read, Write
0	42	Track Display No. 2	Address	
0	43	Track Display No. 3	Address	Read, Write
0	44	Track Display No. 4	Address	Read, Write
0	45	Track Display No. 5	Address	Read, Write
0	46	Track Display No. 6	Address	Read, Write

2.8.3 Select (SEL) Instruction

This instruction selects a specific IO unit for a subsequent IO operation and deselects any previously selected drum field or IO unit. (See table 2-11.) This

instruction cannot be executed until the IO interlock is cleared. Execution time of the *SEL* instruction, which is technically indexable, is 12 microseconds. The octonary operation code is 62-.

TABLE 2-11. IO UNIT SELECTION CODES

OCTONARY CODE	UNIT	OPERATION
21	Burst Time Counter and G/A Elapsed Time Counter	Read
02	Card Punch	Write
01	Card Reader	Read
04	IO Register (Containing Positive Zero)	Read
03	Line Printer	Write
11	Magnetic Tape Unit No. 1	Read, Write
12	Magnetic Tape Unit No. 2	Read, Write
13	Magnetic Tape Unit No. 3	Read, Write
14	Magnetic Tape Unit No. 4	Read, Write
15	Magnetic Tape Unit No. 5	Read, Write
16	Magnetic Tape Unit No. 6	Read, Write
06	Manual Input Matrix	Write
10	Warning Light System	Read

2.8.4 Read (RDS) Instruction

This instruction initiates the transfer of *n* words (specified by the right half-word of the instruction and suitably modified if an index register is designated) from an IO unit which has been selected previously and is ready to operate. The words are transferred from the selected IO unit to a block of consecutive registers in core memory, starting at the address contained in the IO address counter. This instruction turns on the IO interlock (which must be off before the instruction can be executed). Execution time of the *RDS* instruction, which is indexable, is 6.0 microseconds. The octonary operation code is 670.

2.8.5 Write (WRT) Instruction

This instruction initiates the transfer of *n* words (specified by the right half-word of the instruction and suitably modified if an index register is designated) to an IO unit which has been selected previously and is ready to operate. The words are transferred to the selected IO unit from a block of consecutive registers in internal memory, starting with the address contained in the IO address counter. This instruction turns on the IO interlock (which must be off before the instruction can be executed). Execution time of the *WRT* instruction, which is indexable, is 6.0 microseconds. The octonary operation code is 674.

2.9 RESET CLASS

2.9.1 Reset Index Register (XIN) Instruction

This instruction replaces the contents of the specified index register with the contents of the right half-word of the instruction. The right accumulator cannot be used as an index register with this instruction. Execution time of the *XIN* instruction is 6.0 microseconds. The octonary operation code is 754.

2.9.2 Reset Index Register from Right Accumulator (XAC) Instruction

This instruction replaces the contents of the specified index register with the contents of the right accumulator. The right accumulator cannot be used as an index register with this instruction. Execution time of the *XAC* instruction is 6.0 microseconds. The octonary operation code is 764.

2.9.3 Add Index Register (ADX) Instruction

This instruction adds the contents of the specified index register to the right half-word of the instruction, leaving the results in the right A register. Overflow is possible upon execution of this instruction, but it does not affect the overflow alarm or overflow sense units. Execution time of the *ADX* instruction is 6.0 microseconds. The octonary operation code is 770.

2.10 ILLEGAL INSTRUCTIONS

2.10.1 General

Since 7 bits (L4 through L10) are utilized to specify an instruction, 128 possible combinations exist. Only 53 of these codes are utilized, but each of the remaining 75 codes will cause the Central Computer System to take some action. Since L4 through L6 are used to specify the instruction class and eight classes are used, there is no unused class code. However, within a given class, the number of unused variations ranges from a maximum of 13 (decimal) in the reset class to a minimum of 7 (decimal) in the add class. The action which is caused by the unused variations depends on the sequence of commands which are conditioned by class; thus, within a given class, all unused variations give the same result. The fact that these unused variations are called illegal instructions does not preclude them from being used as useful instructions. However, current assembly programs do not interpret them as instructions. They may be loaded into core memory as binary constants and so arranged that they will be interpreted as instructions.

2.10.2 Illegal Instruction, Type 0

Octonary operation codes 034, 040, 044, 050, 060, 064, 070, and 074 are illegal instructions (type 0). If any of these instructions is decoded, the Central Computer System performs no operations for 6.0 microseconds.

2.10.3 Illegal Instruction, Type 1

Octonary operation codes 120, 124, 144, 150, 154, 170, and 174 are illegal instructions (type 1). If any of these instructions is decoded, the number in the left half of register $x + in$ is added to the number in the left accumulator. The results remain in the left accumulator, and the A register is cleared. These instructions may cause overflow. Execution time of type 1 illegal instructions, which are indexable, is 12 microseconds.

2.10.4 Illegal Instruction, Type 2

Octonary operation codes 200, 204, 210, 214, 220, 230, 234, 240, 244, 270, and 274 are illegal instructions (type 2). Execution of any of these instructions causes the number in the left accumulator and left B register to be complemented if the original contents of R10 through R15 of the instruction (before possible modification by an index register) are 00_8 or 01_8 , and if bit position LS of register $x + in$ contains a 1.

If the content of R10 through R15 of the instruction (before possible modification by an index register) is neither 00_8 nor 01_8 , the program counter is stepped, and the Central Computer System comes to a stop. Any IO operation in progress at the time will be completed. If LS of register $x + in$ is a 0, no action is taken. Execution time of type 2 illegal instructions, which are indexable, is 12 microseconds.

2.10.5 Illegal Instruction, Type 3

Octonary operation codes 300, 304, 310, 314, 320, 354, 364, 370, and 374 are illegal instructions (type 3). Execution of any of these instructions places positive zero into register $x + in$. Execution time of type 3 illegal instructions, which are indexable, is 18 microseconds.

2.10.6 Illegal Instruction, Type 4

Octonary operation codes 410, 414, 430, 434, 450, 454, 464, and 474 are illegal instructions (type 4). Upon execution of any of these instructions, no action is taken by the Central Computer System for a period of time determined by the value of n , contained in the right half-word. If n is less than 7, execution time is 6.0 microseconds. If $n \geq 7$, minimum execution time is $\frac{n+7}{2}$ microseconds, and maximum $15 \frac{n+9}{2}$ microseconds.

2.10.7 Illegal Instruction, Type 5

Octonary operation codes 500, 504, 530, 534, 560, 564, 570, and 574 are illegal instructions (type 5). Execution of any of these instructions places positive zero into the A registers. Execution time of type 5 illegal instructions is 6.0 microseconds.

2.10.8 Illegal Instruction, Type 6

Octonary operation codes 604, 630, 634, 640, 644, 650, 654, 660, and 664 are illegal instructions (type 6). Any of these instructions, which cannot be executed until the IO interlock is off, causes the Central Computer System to take no action for a period of 6.0 microseconds.

2.10.9 Illegal Instruction, Type 7

Octonary operation codes 700, 704, 710, 714, 720, 724, 730, 734, 740, 744, 750, 760, and 774 are illegal instructions (type 7). Execution of any of these instructions causes the Central Computer System to take no action for a period of 6.0 microseconds.

SECTION 3

OTHER PERTINENT DATA

3.1 IO PROCESS

3.1.1 IO Transfers

All transfers of words involving a specific IO unit are performed as block transfers to or from internal memory. Only one block transfer can be in progress at any time. A block transfer is initiated by an *RDS* or *WRT* instruction. The transfer then proceeds at a rate determined primarily by the IO unit involved. The program normally does not wait for the transfer to take place, except to stop momentarily to relinquish a memory cycle whenever one is needed by the IO unit to deliver a word to or receive a word from internal memory. Normally, an IO transfer is programmed by three instructions, as shown in table 2-12.

TABLE 2-12. PROGRAMMING AN IO TRANSFER

INSTRUCTION	ACTION
1. <i>SEL</i> (<i>u</i>)	Selects the IO Unit.
or	
<i>SDR</i> (<i>u</i>)	Selects drum group, drum, field and mode.
2. <i>LDC</i> <i>x</i>	Specifies the starting address, <i>x</i> , of the block in internal memory.
3. <i>RDS</i> (<i>i</i>) <i>n</i>	Initiates the reading of <i>n</i> words from the selected IO unit, where (<i>i</i>) specifies the interleave factor.
or	
<i>WRT</i> (<i>i</i>) <i>n</i>	Transfers <i>n</i> words to the selected IO unit, where (<i>i</i>) specifies the interleave factor.

It is of no consequence whether *LDC* precedes or follows *SEL* or *SDR*, but *SEL* or *SDR* and *LDC* (if given) must precede the *RDS* or *WRT* instruction. Any number of instructions (except IO instructions or *HLT*) may intervene between instructions which specify an IO transfer. The *SEL* or *SDR* instruction may be omitted if the most recently executed *SEL* or *SDR* instruction has already selected the IO unit and the mode desired for this transfer and if the most recently executed *RDS* or *WRT* instruction specified no interleave. The *LDC* instruction may be omitted if the IO address counter already contains the desired address.

The number of words that are normally transferred in a single IO transfer is the integer specified by all 16 bits of the right half-word of the *RDS* or *WRT* instruction (suitably modified if an index register is specified). To illustrate, a right half-word (after indexing, if any) which is 1.77777_8 (equal to -0) calls for the transfer of 177777_8 or $65,535_{10}$ words. In reading from a drum in the status or status identity modes (or in reading a magnetic tape record), if less than *n* words is on the selected drum (or tape), the transfer is limited to reading all the available words. Similarly, in writing on a drum in the status mode, if the number of empty registers is less than *n*, the transfer is limited to filling the empty drum registers. By examining the contents of the IO word counter after the transfer is completed, the program can determine how many words were actually transferred.

3.1.2 IO Interlock

The IO interlock is on whenever an IO transfer is in progress. It is turned on by the *RDS* or *WRT* instruction, which initiates the transfer, and is turned off when all *n* words have been transferred (the IO word counter goes to $+0$) or when no more words can be transferred (the selected IO unit sends a disconnect signal). The IO interlock is never turned off automatically, and the program hangs up on the next IO instruction if any of the following conditions exists:

- a. A nonexistent unit is selected.
- b. The selected unit is not turned on.
- c. The selected unit is not ready.
- d. The selected unit is not prepared.
- e. The *RDS* or *WRT* operation is not legal for the selected unit.

The IO interlock can be sensed at any time by *BSN* (14) *x*, which will branch if the IO interlock is on. An *Operate* instruction, *PER* (27), is provided which clears the IO interlock regardless of whether an IO transfer is in progress. However, this instruction is intended for use only when the Central Computer System is in a condition which could never result in the IO interlock's being cleared automatically. Clearing the IO interlock by the *PER* (27) instruction while an IO transfer is in progress may result in actions both undesirable and unpredictable.

3.1.3 IO Word Counter

At any instant during normal IO transfers, the IO word counter holds a quantity (properly a 16-bit integer with negative sign understood) related to the number of words already transferred. If all n words have been transferred, the IO word counter contains $+0$. If t words have been transferred, where $t < n$, the contents of the counter are a negative quantity equal to $(-n + t + 1)$. The number of words already transferred, t , can be found by adding to this quantity the value of $(n-1)$. When all words but one have been transferred ($t = n-1$), the contents of the IO word counter equal -0 . However, if a transfer is interrupted by an alarm, the IO word counter cannot be used to determine the number of words transferred.

3.1.4 IO Address Counter

At any instant, the IO address counter holds the internal memory address to or from which the next word of an IO transfer is to be transferred. The IO address counter is normally indexed by 1 automatically for each word transferred. Therefore, at the end of a transfer, the counter usually holds an address one greater than the internal memory address in which the last word was stored (or from which it was taken). There are two exceptions: at the end of a block transfer to a drum, the address in the counter is three greater than the last address of the block; if the IO address counter is locked, the address in the counter is one greater than when the recent *Operate* (lock address counter) instruction, *PER* (75)₈, was executed.

Occasionally it is desirable to execute a transfer to or from a single register in internal memory. This is done by preceding the *RDS* or *WRT* instruction by a *PER* (75) instruction, which locks the IO address counter so that it cannot be changed. The IO address counter is unlocked automatically the next time the IO interlock is cleared. The *PER* (75) instruction must not be given while the IO interlock is on.

Only the right 14 bits of the IO address counter are connected as a counter; hence, it is a counter modulo 20000₈ words (which is rarely practical), the transfer involves either core memory and test memory (alternately) or only the clock register.

3.1.5 Interleave

Reading from and writing on drums, using the address mode, can be modified by interleave; that is, every 8th, 16th, or 64th drum register can be read or written instead of consecutive registers. The corresponding block of words in internal memory always is in consecutive registers (or, if the IO address counter is locked, in a single register). The interleave pattern is specified by the contents of L13 through L15 of the *RDS* or *WRT* instruction, which initiates the transfer. (See table 2–13.) The use of other codes for interleave results in unpredictable actions.

TABLE 2–13. INTERLEAVE CODE

OCTONARY CODE (L13-L15)	INTERLEAVE PATTERN
0	Consecutive Registers
1	Every 8th Register
2	Every 16th Register
4	Every 64th Register

The interleaving increment will be 1 greater than 8, 16, or 64 when it encloses the boundary between addresses 2,047₁₀ and 0. For example, reading every 8th register starting with 0 would result in reading registers 0, 8, 16, . . . 8n, . . . 2040, 1, 9, 17, etc. Since the programmer deals primarily in octonary numbers, table 2–14 is included for ease in octonary-to-decimal conversion.

TABLE 2–14. OCTONARY-DECIMAL INTEGER CONVERSION

0	1	2	3	4	5	6	7		
0000	0000	0001	0002	0003	0004	0005	0006	0007	0000
0010	0008	0009	0010	0011	0012	0013	0014	0015	to
0020	0016	0017	0018	0019	0020	0021	0022	0023	0377
0030	0024	0025	0026	0027	0028	0029	0030	0031	(Octonary)
0040	0032	0033	0034	0035	0036	0037	0038	0039	(Decimal)
0050	0040	0041	0042	0043	0044	0045	0046	0047	
0060	0048	0049	0050	0051	0052	0053	0054	0055	
0070	0056	0057	0058	0059	0060	0061	0062	0063	
0100	0064	0065	0066	0067	0068	0069	0070	0071	
0110	0072	0073	0074	0075	0076	0077	0078	0079	

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
0120	0080	0081	0082	0083	0084	0085	0086	0087		
0130	0088	0089	0090	0091	0082	0093	0094	0095		
0140	0096	0097	0098	0099	0100	0101	0102	0103		
0150	0104	0105	0106	0107	0108	0109	0110	0111		
0160	0112	0113	0114	0115	0116	0117	0118	0119		
0170	0120	0121	0122	0123	0124	0125	0126	0127		
0200	0128	0129	0130	0131	0132	0133	0134	0135		
0210	0136	0137	0138	0139	0140	0141	0142	0143		
0220	0144	0145	0146	0147	0148	0149	0150	0151		
0230	0152	0153	0154	0155	0156	0157	0158	0159		
0240	0160	0161	0162	0163	0164	0165	0166	0167		
0250	0168	0169	0170	0171	0172	0173	0174	0175		
0260	0176	0177	0178	0179	0180	0181	0182	0183		
0270	0184	0185	0186	0187	0188	0189	0190	0191		
0300	0192	0193	0194	0195	0196	0197	0198	0199		
0310	0200	0201	0202	0203	0204	0205	0206	0207		
0320	0208	0209	0210	0211	0212	0213	0214	0215		
0330	0216	0217	0218	0219	0220	0221	0222	0223		
0340	0224	0225	0226	0227	0228	0229	0230	0231		
0350	0232	0233	0234	0235	0236	0237	0238	0239		
0360	0240	0241	0242	0243	0244	0245	0246	0247		
0370	0248	0249	0250	0251	0252	0253	0254	0255		
0400	0256	0257	0258	0259	0260	0261	0262	0263	0400	0256
0410	0264	0265	0266	0267	0268	0269	0270	0271	to	to
0420	0272	0273	0274	0275	0276	0277	0278	0279	0777	0511
0430	0280	0281	0282	0283	0284	0285	0286	0287	(Octonary)	(Decimal)
0440	0288	0289	0290	0291	0292	0293	0294	0295		
0450	0296	0297	0298	0299	0300	0301	0302	0303		
0460	0304	0305	0306	0307	0308	0309	0310	0311		
0470	0312	0313	0314	0315	0316	0317	0318	0319		
0500	0320	0321	0322	0323	0324	0325	0326	0327		
0510	0328	0329	0330	0331	0332	0333	0334	0335		
0520	0336	0337	0338	0339	0340	0341	0342	0343		
0530	0344	0345	0346	0347	0348	0349	0350	0351		
0540	0352	0353	0354	0355	0356	0357	0358	0359		
0550	0360	0361	0362	0363	0364	0365	0366	0367		
0560	0368	0369	0370	0371	0372	0373	0374	0375		
0570	0376	0377	0378	0379	0380	0381	0382	0383		
0600	0384	0385	0386	0387	0388	0389	0390	0391		
0610	0392	0393	0394	0395	0396	0397	0398	0399		
0620	0400	0401	0402	0403	0404	0405	0406	0407		
0630	0408	0409	0410	0411	0412	0413	0414	0415		
0640	0416	0417	0418	0419	0420	0421	0422	0423		
0650	0424	0425	0426	0427	0428	0429	0430	0431		
0660	0432	0433	0434	0435	0436	0437	0438	0439		
0670	0440	0441	0442	0443	0444	0445	0446	0447		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
0700	0448	0449	0450	0451	0452	0453	0454	0455	1000	0512
0710	0456	0457	0458	0459	0460	0461	0462	0463	to	to
0720	0464	0465	0466	0467	0468	0469	0470	0471	1377	0767
0730	0472	0473	0474	0475	0476	0477	0478	0479	(Octonary)	(Decimal)
0740	0480	0481	0482	0483	0484	0485	0486	0487		
0750	0488	0489	0490	0491	0492	0493	0494	0495		
0760	0496	0497	0498	0499	0500	0501	0502	0503		
0770	0504	0505	0506	0507	0508	0509	0510	0511		
1000	0512	0513	0514	0515	0516	0517	0518	0519		
1010	0520	0521	0522	0523	0524	0525	0526	0527		
1020	0528	0529	0530	0531	0532	0533	0534	0535		
1030	0536	0537	0538	0539	0540	0541	0542	0543		
1040	0544	0545	0546	0547	0548	0549	0550	0551		
1050	0552	0553	0554	0555	0556	0557	0558	0559		
1060	0560	0561	0562	0563	0564	0565	0566	0567		
1070	0568	0569	0570	0571	0572	0573	0574	0575		
1100	0576	0577	0578	0579	0580	0581	0582	0583		
1110	0584	0585	0586	0587	0588	0589	0590	0591		
1120	0592	0593	0594	0595	0596	0597	0598	0599		
1130	0600	0601	0602	0603	0604	0605	0606	0607		
1140	0608	0609	0610	0611	0612	0613	0614	0615		
1150	0616	0617	0618	0619	0620	0621	0622	0623		
1160	0624	0625	0626	0627	0628	0629	0630	0631		
1170	0632	0633	0634	0635	0636	0637	0638	0639		
1200	0640	0641	0642	0643	0644	0645	0646	0647		
1210	0648	0649	0650	0651	0652	0653	0654	0655		
1220	0656	0657	0658	0659	0660	0661	0662	0663		
1230	0664	0665	0666	0667	0668	0669	0670	0671		
1240	0672	0673	0674	0675	0676	0677	0678	0679		
1250	0680	0681	0682	0683	0684	0685	0686	0687		
1260	0688	0689	0690	0691	0692	0693	0694	0695		
1270	0696	0697	0698	0699	0700	0701	0702	0703		
1300	0704	0705	0706	0707	0708	0709	0710	0711		
1310	0712	0713	0714	0715	0716	0717	0718	0719		
1320	0720	0721	0722	0723	0724	0725	0726	0727		
1330	0728	0729	0730	0731	0732	0733	0734	0735		
1340	0736	0737	0738	0739	0740	0741	0742	0743		
1350	0744	0745	0746	0747	0748	0749	0750	0751		
1360	0752	0753	0754	0755	0756	0757	0758	0759		
1370	0760	0761	0762	0763	0764	0765	0766	0767		
1400	0768	0769	0770	0771	0772	0773	0774	0775	1400	0768
1410	0776	0777	0778	0779	0780	0781	0782	0783	to	to
1420	0784	0785	0786	0787	0788	0789	0790	0791	1777	1023
1430	0792	0793	0794	0795	0796	0797	0798	0799	(Octonary)	(Decimal)
1440	0800	0801	0802	0803	0804	0805	0806	0807		
1450	0808	0809	0810	0811	0812	0813	0814	0815		
1460	0816	0817	0818	0819	0820	0821	0822	0823		
1470	0824	0825	0826	0827	0828	0829	0830	0831		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
1500	0832	0833	0834	0835	0836	0837	0838	0839		
1510	0840	0841	0842	0843	0844	0845	0846	0847		
1420	0848	0849	0850	0851	0852	0853	0854	0855		
1530	0856	0857	0858	0859	0860	0861	0862	0863		
1540	0864	0865	0866	0867	0868	0869	0870	0871		
1550	0872	0873	0874	0875	0876	0877	0878	0879		
1560	0880	0881	0882	0883	0884	0885	0886	0887		
1570	0888	0889	0890	0891	0892	8093	0894	0895		
1600	0896	0897	0898	0899	0900	9001	0902	0903		
1610	0904	0905	0906	0907	0908	0909	0910	0911		
1620	0912	0913	0914	0915	0916	0917	0918	0919		
1630	0920	0921	0922	0923	0924	0925	0926	0927		
1640	0928	0929	0930	0931	0932	0933	0934	0935		
1650	0936	0937	0938	0939	0940	0941	0942	0943		
1660	0944	0945	0946	0947	0948	0949	0950	0951		
1670	0952	0953	0954	0955	0956	0957	0958	0959		
1700	0960	0961	0962	0963	0964	0965	0966	0967		
1710	0968	0969	0970	0971	0972	0973	0974	0975		
1720	0976	0977	0978	0979	0980	0981	0982	0983		
1730	0984	0985	0986	0987	0988	0989	0990	0991		
1740	0992	0993	0994	0995	0996	0997	0998	0999		
1750	1000	1001	1002	1003	1004	1005	1006	1007		
1760	1008	1009	1010	1011	1012	1013	1014	1015		
1770	1016	1017	1018	1019	1020	1021	1022	1023		
2000	1024	1025	1026	1027	1028	1029	1030	1031	2000	1024
2010	1032	1033	1034	1035	1036	1037	1038	1039	to	to
2020	1040	1041	1042	1043	1044	1045	1046	1047	2377	1279
2030	1048	1049	1050	1051	1052	1053	1054	1055	(Octonary)	(Decimal)
2040	1056	1057	1058	1059	1060	1061	1062	1063		
2050	1064	1065	1066	1067	1068	1069	1070	1071		
2060	1072	1073	1074	1075	1076	1077	1078	1079		
2070	1080	1081	1082	1083	1084	1085	1086	1087		
2100	1088	1089	1090	1091	1092	1093	1094	1095		
2110	1096	1097	1098	1099	1100	1101	1102	1103		
2120	1104	1105	1106	1107	1108	1109	1110	1111		
2130	1112	1113	1114	1115	1116	1117	1118	1119		
2140	1120	1121	1122	1123	1124	1125	1126	1127		
2150	1128	1129	1130	1131	1132	1133	1134	1135		
2160	1136	1137	1138	1139	1140	1141	1142	1143		
2170	1144	1145	1146	1147	1148	1149	1150	1151		
2200	1152	1153	1154	1155	1156	1157	1158	1159		
2210	1160	1161	1162	1163	1164	1165	1166	1167		
2220	1168	1169	1170	1171	1172	1173	1174	1175		
2230	1176	1177	1178	1179	1180	1181	1182	1183		
2240	1184	1185	1186	1187	1188	1189	1190	1191		
2250	1192	1193	1194	1195	1196	1197	1198	1199		
2260	1200	1201	1202	1203	1204	1205	1206	1207		
2270	1208	1209	1210	1211	1212	1213	1214	1215		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
2300	1216	1217	1218	1219	1220	1221	1222	1223		
2310	1224	1225	1226	1227	1228	1229	1230	1231		
2320	1232	1233	1234	1235	1236	1237	1238	1239		
2330	1240	1241	1242	1243	1244	1245	1246	1247		
2340	1248	1249	1250	1251	1252	1253	1254	1255		
2350	1256	1257	1258	1259	1260	1261	1262	1263		
2360	1264	1265	1266	1267	1268	1269	1270	1271		
2370	1272	1273	1274	1275	1276	1277	1278	1279		
2400	1280	1281	1282	1283	1284	1285	1286	1287	2400	1280
2410	1288	1289	1290	1291	1292	1293	1294	1295	to	to
2420	1296	1297	1298	1299	1300	1301	1302	1303	2777	1535
2430	1304	1305	1306	1307	1308	1309	1310	1311	(Octonary)	(Decimal)
2440	1312	1313	1314	1315	1316	1317	1318	1319		
2450	1320	1321	1322	1323	1324	1325	1326	1327		
2460	1328	1329	1330	1331	1332	1333	1334	1335		
2470	1336	1337	1338	1339	1340	1341	1342	1343		
2500	1344	1345	1346	1347	1348	1349	1350	1351		
2510	1352	1353	1354	1355	1356	1357	1358	1359		
2520	1360	1361	1362	1363	1364	1365	1366	1367		
2530	1368	1369	1370	1371	1372	1373	1374	1375		
2540	1376	1377	1378	1379	1380	1381	1382	1383		
2550	1384	1385	1386	1387	1388	1389	1390	1391		
2560	1392	1393	1394	1395	1396	1397	1398	1399		
2570	1400	1401	1402	1403	1404	1405	1406	1407		
2600	1408	1409	1410	1411	1412	1413	1414	1415		
2610	1416	1417	1418	1419	1420	1421	1422	1423		
2620	1424	1425	1426	1427	1428	1429	1430	1431		
2630	1432	1433	1434	1435	1436	1437	1438	1439		
2640	1440	1441	1442	1443	1444	1445	1446	1447		
2650	1448	1449	1450	1451	1452	1453	1454	1455		
2660	1456	1457	1458	1459	1460	1461	1462	1463		
2670	1464	1465	1466	1467	1468	1469	1470	1471		
2700	1472	1473	1474	1475	1476	1477	1478	1479		
2710	1480	1481	1482	1483	1484	1485	1486	1487		
2720	1488	1489	1490	1491	1492	1493	1494	1495		
2730	1496	1497	1498	1499	1500	1501	1502	1503		
2740	1504	1505	1506	1507	1508	1509	1510	1511		
2750	1512	1513	1514	1515	1516	1517	1518	1519		
2760	1520	1521	1522	1523	1524	1525	1526	1527		
2770	1528	1529	1530	1531	1532	1533	1534	1535		
3000	1536	1437	1538	1539	1540	1541	1542	1543	3000	1536
3010	1544	1545	1546	1447	1548	1549	1550	1551	to	to
3020	1552	1553	1554	1555	1556	1557	1558	1559	3377	1791
3030	1560	1561	1562	1563	1564	1565	1566	1567	(Octonary)	(Decimal)
3040	1568	1569	1670	1571	1572	1573	1574	1575		
3050	1576	1577	1578	1579	1580	1581	1582	1583		
3060	1584	1585	1586	1587	1588	1589	1590	1591		
3070	1592	1593	1594	1595	1596	1597	1598	1599		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
3100	1600	1601	1602	1603	1604	1605	1606	1607		
3110	1608	1609	1610	1611	1612	1613	1614	1615		
3120	1616	1617	1618	1619	1620	1621	1622	1623		
3130	1624	1625	1626	1627	1628	1629	1630	1631		
3140	1632	1633	1634	1635	1636	1637	1638	1639		
3150	1640	1641	1642	1643	1644	1645	1646	1647		
3160	1648	1649	1650	1651	1652	1653	1654	1655		
3170	1656	1657	1658	1659	1660	1661	1662	1663		
3200	1664	1665	1666	1667	1668	1669	1670	1671		
3210	1672	1673	1674	1675	1676	1677	1678	1679		
3220	1680	1681	1682	1683	1684	1685	1686	1687		
3230	1688	1689	1690	1691	1692	1693	1694	1695		
3240	1696	1697	1698	1699	1700	1701	1702	1703		
3250	1704	1705	1706	1707	1708	1709	1710	1711		
3260	1712	1713	1714	1715	1716	1717	1718	1719		
3270	1720	1721	1722	1723	1724	1725	1726	1727		
3300	1728	1729	1730	1731	1732	1733	1734	1735		
3310	1736	1737	1738	1739	1740	1741	1742	1743		
3320	1744	1745	1746	1747	1748	1749	1750	1751		
3330	1752	1753	1754	1755	1756	1757	1758	1759		
3340	1760	1761	1762	1763	1764	1765	1766	1767		
3350	1768	1769	1770	1771	1772	1773	1774	1775		
3360	1776	1777	1778	1779	1780	1781	1782	1783		
3370	1784	1785	1786	1787	1788	1789	1790	1791		
3400	1792	1793	1794	1795	1796	1797	1798	1799	3400	1792
3410	1800	1801	1802	1803	1804	1805	1806	1807	to	to
3420	1808	1809	1810	1811	1812	1813	1814	1815	3777	2047
3430	1816	1817	1818	1819	1820	1821	1822	1823	(Octonary)	(Decimal)
3440	1824	1825	1826	1827	1828	1829	1830	1831		
3450	1832	1833	1834	1835	1836	1837	1838	1839		
3460	1840	1841	1842	1843	1844	1845	1846	1847		
3470	1848	1849	1850	1851	1852	1853	1854	1855		
3500	1856	1857	1858	1859	1860	1861	1862	1863		
3510	1864	1865	1866	1867	1868	1869	1870	1871		
3520	1872	1873	1874	1875	1876	1877	1878	1879		
3530	1880	1881	1882	1883	1884	1885	1886	1887		
3540	1888	1889	1890	1891	1892	1893	1894	1895		
3550	1896	1897	1898	1899	1900	1901	1902	1903		
3560	1904	1905	1906	1907	1908	1909	1910	1911		
3570	1912	1913	1914	1915	1916	1817	1918	1919		
3600	1920	1921	1922	1923	1924	1925	1926	1927		
3610	1928	1929	1930	1931	1932	1933	1934	1935		
3620	1936	1937	1938	1939	1940	1941	1942	1943		
3630	1944	1945	1946	1947	1948	1949	1950	1951		
3640	1952	1953	1954	1955	1956	1957	1958	1959		
3650	1960	1961	1962	1963	1964	1965	1966	1967		
3660	1968	1969	1670	1971	1972	1973	1974	1975		
3670	1976	1977	1978	1979	1980	1981	1982	1983		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
3700	1984	1985	1986	1987	1988	1989	1990	1991		
3710	1992	1993	1994	1995	1996	1997	1998	1999		
3720	2000	2001	2002	2003	2004	2005	2006	2007		
3730	2008	2009	2010	2011	2012	2013	2014	2015		
3740	2016	2017	2018	2019	2020	2021	2022	2023		
3750	2024	2025	2026	2027	2028	2029	2030	2031		
3760	2032	2033	2034	2035	2036	2037	2038	2039		
3770	2040	2041	2042	2043	2044	2045	2046	2047		
4000	2048	2049	2050	2051	2052	2053	2054	2055	4000	2048
4010	2056	2057	2058	2059	2060	2061	2062	2063	to	to
4020	2064	2065	2066	2067	2068	2069	2070	2071	4377	2303
4030	2072	2073	2074	2075	2076	2077	2078	2079	(Octonary)	(Decimal)
4040	2080	2081	2082	2083	2084	2085	2086	2087		
4050	2088	2089	2090	2091	2092	2093	2094	2095		
4060	2096	2097	2098	2099	2100	2101	2102	2103		
4070	2104	2105	2106	2107	2108	2109	2110	2111		
4100	2112	2113	2114	2115	2116	2117	2118	2119		
4110	2120	2121	2122	2123	2124	2125	2126	2127		
4120	2128	2129	2130	2131	2132	2133	2134	2135		
4130	2136	2137	2138	2139	2140	2141	2142	2143		
4140	2144	2145	2146	2147	2148	2149	2150	2151		
4150	2152	2153	2154	2155	2156	2157	2158	2159		
4160	2160	2161	2162	2163	2164	2165	2166	2167		
4170	2168	2169	2170	2171	2172	2173	2174	2175		
4200	2176	2177	2178	2179	2180	2181	2182	2183		
4210	2184	2185	2186	2187	2188	2189	2190	2191		
4220	2192	2193	2194	2195	2196	2197	2198	2199		
4230	2200	2201	2202	2203	2204	2205	2206	2207		
4240	2208	2209	2210	2211	2212	2213	2214	2215		
4250	2216	2217	2218	2219	2220	2221	2222	2223		
4260	2224	2225	2226	2227	2228	2229	2230	2231		
4270	2232	2233	2234	2235	2236	2237	2238	2239		
4300	2240	2241	2242	2243	2244	2245	2246	2247		
4310	2248	2249	2250	2251	2252	2253	2254	2255		
4320	2256	2257	2258	2259	2260	2261	2262	2263		
4330	2264	2265	2266	2267	2268	2269	2270	2271		
4340	2272	2273	2274	2275	2276	2277	2278	2279		
4350	2280	2281	2282	2283	2284	2285	2286	2287		
4360	2288	2289	2290	2291	2292	2293	2294	2295		
4370	2296	2297	2298	2299	2300	2301	2302	2303		
4400	2304	2305	2306	2307	2308	2309	2310	2311	4400	2304
4410	2312	2313	2314	2315	2316	2317	2318	2319	to	to
4420	2320	2321	2322	2323	2324	2325	2326	2327	4777	2559
4430	2328	2329	2330	2331	2332	2333	2334	2335	(Octonary)	(Decimal)
4440	2336	2337	2338	2339	2340	2341	2342	2343		
4450	2344	2345	2346	2347	2348	2349	2350	2351		
4460	2352	2353	2354	2355	2356	2357	2358	2359		
4470	2360	2361	2362	2363	2364	2365	2366	2367		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
4500	2368	2369	2370	2371	2372	2373	2374	2375		
4510	2376	2377	2378	2379	2380	2381	2382	2383		
4520	2384	2385	2386	2387	2388	2389	2390	2391		
4530	2392	2393	2394	2395	2396	2397	2398	2399		
4540	2400	2401	2402	2403	2404	2405	2406	2407		
4550	2408	2409	2410	2411	2412	2413	2414	2415		
4560	2416	2417	2418	2419	2420	2421	2422	2423		
4570	2424	2425	2426	2427	2428	2429	2430	2431		
4600	2432	2433	2434	2435	2436	2437	2438	2439		
4610	2440	2441	2442	2443	2444	2445	2446	2447		
4620	2448	2449	2450	2451	2452	2453	2454	2455		
4630	2456	2457	2458	2459	2460	2461	2462	2563		
4640	2464	2465	2466	2467	2468	2469	2470	2471		
4650	2472	2473	2474	2475	2476	2477	2478	2479		
4660	2480	2481	2482	2483	2484	2485	2486	2487		
4670	2488	2489	2490	2491	2492	2493	2494	2495		
4700	2496	2497	2498	2499	2500	2501	2502	2503		
4710	2504	2505	2506	2507	2508	2509	2510	2511		
4720	2512	2513	2514	2516	2516	2517	2518	2519		
4730	2520	2521	2522	2523	2524	2525	2526	2527		
4740	2528	2529	2530	2531	2532	2533	2534	2535		
4750	2536	2537	2538	2539	2540	2541	2542	2543		
4760	2544	2545	2546	2547	2548	2549	2550	2551		
4770	2552	2553	2554	2555	2556	2557	2558	2559		
5000	2660	2661	2662	2663	2664	2665	2666	2667	5000	2660
5010	2568	2569	2570	2571	2572	2573	2574	2575	to	to
5020	2576	2577	2578	2579	2580	2581	2582	2583	5377	2815
5030	2584	2585	2586	2587	2588	2589	2590	2591	(Octonary)	(Decimal)
5040	2592	2593	2594	2595	2596	2597	2598	2599		
5050	2600	2601	2602	2603	2604	2605	2606	2607		
5060	2608	2609	2610	2611	2612	2613	2614	2615		
5070	2616	2617	2618	2619	2620	2621	2622	2623		
5100	2624	2625	2626	2627	2628	2629	2630	2631		
5110	2632	2633	2634	2635	2636	2637	2638	2639		
5120	2640	2641	2642	2643	2644	2645	2646	2647		
5130	2648	2649	2650	2651	2652	2653	2654	2655		
5140	2656	2657	2658	2659	2660	2661	2652	2663		
5150	2664	2665	2666	2667	2668	2669	2670	2671		
5160	2672	2673	2674	2675	2676	2677	2678	2679		
5170	2680	2681	2682	2683	2684	2685	2686	2687		
5200	2688	2689	2690	2691	2692	2693	2694	2695		
5210	2696	2697	2698	2699	2700	2701	2702	2703		
5220	2704	2705	2706	2707	2708	2709	2710	2711		
5230	2712	2713	2714	2715	2716	2717	2718	2719		
5240	2720	2721	2722	2723	2724	2725	2726	2727		
5250	2728	2729	2730	2731	2732	2733	2734	2735		
5260	2736	2737	2738	2739	2740	2741	2742	2743		
5270	2744	2745	2746	2747	2748	2749	2750	2751		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
5300	2752	2753	2754	2755	2756	2757	2758	2759		
5310	2760	2761	2762	2763	2764	2765	2766	2767		
5320	2768	2769	2770	2771	2772	2773	2774	2775		
5330	2776	2777	2778	2779	2780	2781	2782	2783		
5340	2784	2785	2786	2787	2788	2789	2790	2791		
5350	2792	2793	2794	2795	2796	2797	2798	2799		
5360	2800	2801	2802	2803	2804	2805	2806	2807		
5370	2808	2809	2810	2811	2812	2813	2814	2815		
5400	2816	2817	2818	2819	2820	2821	2822	2823	5400	2816
5410	2824	2825	2826	2827	2828	2829	2830	2831	to	to
5420	2832	2833	2834	2835	2836	2837	2838	2839	5777	3071
5430	2840	2841	2842	2843	2844	2845	2846	2847	(Octonary)	(Decimal)
5440	2848	2849	2850	2851	2852	2853	2854	2855		
5450	2856	2857	2858	2859	2860	2861	2862	2863		
5460	2864	2865	2866	2867	2868	2869	2870	2871		
5470	2872	2873	2874	2875	2876	2877	2878	2879		
5500	2880	2881	2882	2883	2884	2885	2886	2887		
5510	2888	2889	2890	2891	2892	2893	2894	2895		
5520	2896	2897	2898	2899	2900	2901	2902	2903		
5530	2904	2905	2906	2907	2908	2909	2910	2911		
5540	2912	2913	2914	2915	2916	2917	2918	2919		
5550	2920	2921	2922	2923	2924	2925	2926	2927		
5560	2928	2929	2930	2931	2932	2933	2934	2935		
5570	2936	2937	2938	2939	2940	2941	2942	2943		
5600	2944	2945	2946	2947	2948	2949	2950	2951		
5610	2952	2953	2954	2955	2956	2957	2958	2959		
5620	2960	2961	2962	2963	2964	2965	2966	2967		
5630	2968	2969	2970	2971	2972	2973	2974	2975		
5640	2976	2977	2978	2979	2980	2981	2982	2983		
5650	2984	2985	2986	2987	2988	2989	2990	2991		
5660	2992	2993	2994	2995	2996	2997	2998	2999		
5670	3000	3001	3002	3003	3004	3005	3006	3007		
5700	3008	3009	3010	3011	3012	3013	3014	3015		
5710	3016	3017	3018	3019	3020	3021	3022	3023		
5720	3024	3025	3026	3027	3028	3029	3030	3031		
5730	3032	3033	3034	3035	3036	3037	3038	3039		
5740	3040	3041	3042	3043	3044	3045	3046	3047		
5750	3048	3049	3050	3051	3052	3053	3054	3055		
5760	3056	3057	3058	3059	3060	3061	3062	3063		
5770	3064	3065	3066	3067	3068	3069	3070	3071		
6000	3072	3073	3074	3075	3076	3077	3078	3079	6000	3072
6010	3080	3081	3082	3083	3084	3085	3086	3087	to	to
6020	3088	3089	3090	3091	3092	3093	3094	3095	6377	3327
6030	3096	3097	3098	3099	3100	3101	3102	3103	(Octonary)	(Decimal)
6040	3104	3105	3106	3107	3108	3109	3110	3111		
6050	3112	3113	3114	3115	3116	3117	3118	3119		
6060	3120	3121	3122	3123	3124	3125	3126	3127		
6070	3128	3129	3130	3131	3132	3133	3134	3135		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
6100	3136	3137	3138	3139	3140	3141	3142	3143		
6110	3144	3145	3146	3147	3148	3149	3150	3151		
6120	3152	3153	3154	3155	3156	3157	3158	3159		
6130	3160	3161	3162	3163	3164	3165	3166	3167		
6140	3168	3169	3170	3171	3172	3173	3174	3175		
6150	3176	3177	3178	3179	3180	3181	3182	3183		
6160	3184	3185	3186	3187	3188	3189	3190	3191		
6170	3192	3193	3194	3195	3196	3197	3198	3199		
6200	3200	3201	3202	3203	3204	3205	3206	3207		
6210	3208	3209	3210	3211	3212	3213	3214	3215		
6220	3216	3217	3218	3219	3220	3221	3222	3223		
6230	3224	3225	3226	3227	3228	3229	3230	3231		
6240	3232	3233	3234	3235	3236	3237	3238	3239		
6250	3240	3241	3242	3243	3244	3245	3246	3247		
6260	3248	3249	3250	3251	3252	3253	3254	3255		
6270	3256	3257	3258	3259	3260	3261	3262	3263		
6300	3264	3265	3266	3267	3268	3269	3270	3271		
6310	3272	3273	3274	3275	3276	3277	3278	3279		
6320	3280	3281	3282	3283	3284	3285	3286	3287		
6330	3288	3289	3290	3291	3292	3293	3294	3295		
6340	3296	3297	3298	3299	3300	3301	3302	3303		
6350	3304	3305	3306	3307	3308	3309	3310	3311		
6360	3312	3313	3314	3315	3316	3317	3318	3319		
6370	3320	3321	3322	3323	3324	3325	3326	3327		
6400	3328	3329	3330	3331	3332	3333	3334	3335	6400	3328
6410	3336	3337	3338	3339	3340	3341	3342	3343	to	to
6420	3344	3345	3346	3347	3348	3349	3350	3351	6777	3583
6430	3352	3353	3354	3355	3356	3357	3358	3359	(Octonary)	(Decimal)
6440	3360	3361	3362	3363	3364	3365	3366	3367		
6450	3368	3369	3370	3371	3372	3373	3374	3375		
6460	3376	3377	3378	3379	3380	3381	3382	3383		
6470	3384	3385	3386	3387	3388	3389	3390	3391		
6500	3392	3393	3394	3395	3396	3397	3398	3399		
6510	3500	3401	3402	3403	3404	3405	3406	3407		
6520	3408	3409	3410	3411	3412	3413	3414	3415		
6530	3416	3417	3418	3419	3420	3421	3422	3423		
6540	3424	3425	3426	3427	3428	3429	3430	3431		
6550	3432	3433	3434	3435	3436	3437	3438	3439		
6560	3440	3441	3442	3443	3444	3445	3446	3447		
6570	3448	3449	3450	3451	3452	3453	3454	3455		
6600	3456	3457	3458	3459	3460	3461	3462	3463		
6610	3464	3465	3466	3467	3468	3469	3470	3471		
6620	3472	3473	3474	3475	3476	3577	3478	3479		
6630	3480	3481	3482	3483	3484	3485	3486	3487		
6640	3488	3489	3490	3491	3492	3493	3494	3495		
6650	3496	3497	3498	3499	3500	3501	3502	3503		
6660	3404	3505	3506	3507	3508	3509	3510	3511		
6670	3512	3513	3514	3515	3516	3517	3518	3519		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7		
6700	3520	3521	3522	3523	3524	3525	3526	3527		
6710	3528	3529	3530	3531	3532	3533	3534	3535		
6720	3536	3537	3537	3539	3540	3541	3542	3543		
6730	3544	3545	3546	3547	3548	3549	3550	3551		
6740	3552	3553	3554	3555	3556	3557	3558	3559		
6750	3560	3561	3562	3563	3564	3565	3566	3567		
6760	3568	3569	3570	3571	3572	3573	3574	3575		
6770	3576	3577	3578	3579	3580	3581	3582	3583		
7000	3584	3585	3586	3587	3588	3589	3590	3591	7000	3584
7010	3592	3593	3594	3595	3596	3597	3598	3599	to	to
7020	3600	3601	3602	3603	3604	3605	3606	3607	7377	3839
7030	3608	3609	3610	3611	3612	3613	3614	3615	(Octonary)	(Decimal)
7040	3616	3617	3618	3619	3620	3621	3622	3623		
7050	3624	3625	3626	3627	3628	3629	3630	3631		
7060	3632	3633	3634	3635	3636	3637	3638	3639		
7070	3640	3641	3642	3643	3644	3645	3646	3647		
7100	3648	3649	3650	3651	3652	3653	3654	3655		
7110	3656	3657	3658	3659	3660	3661	3662	3663		
7120	3664	3665	3666	3667	3668	3669	3670	3671		
7130	3672	3673	3674	3675	3673	3677	3678	3679		
7140	3680	3681	3682	3683	3684	3685	3686	3687		
7150	3688	3689	3690	3691	3692	3693	3694	3695		
7160	3696	3697	3698	3699	3700	3701	3702	3703		
7170	3704	3705	3706	3707	3708	3709	3710	3711		
7200	3712	3713	3714	3715	3716	3717	3718	3719		
7210	3720	3721	3722	3723	3724	3725	3726	3727		
7220	3728	3729	3730	3731	3732	3733	3734	3735		
7230	3736	3737	3738	3739	3740	3741	3742	3743		
7240	3744	3745	3746	3747	3748	3749	3750	3751		
7250	3752	3753	3754	3755	3756	3757	3758	3759		
7260	3760	3761	3762	3763	3764	3765	3766	3767		
7270	3768	3769	3770	3771	3772	3773	3774	3775		
7300	3776	3777	3778	3779	3780	3781	3782	3783		
7310	3784	3785	3786	3787	3788	3789	3790	3791		
7320	3792	3793	3794	3795	3796	3797	3798	3799		
7330	3800	3801	3802	3803	3804	3805	3806	3807		
7340	3808	3809	3810	3811	3812	3813	3814	3815		
7350	3816	3817	3818	3819	3820	3821	3822	3823		
7360	3824	3825	3826	3827	3828	3829	3830	3831		
7270	3832	3833	3834	3835	3836	3837	3838	3839		
7400	3840	3841	3842	3843	3844	3845	3846	3847	7400	3840
7410	3848	3849	3850	3851	3852	3853	3854	3855	to	to
7420	3856	3857	3858	3859	3860	3861	3862	3863	7777	4095
7430	3864	3865	3866	3867	3868	3869	3870	3871	(Octonary)	(Decimal)
7440	3872	3873	3874	3875	3876	3877	3878	3879		
7450	3880	3881	3882	3883	3884	3885	3886	3887		
7460	3888	3889	3890	3891	3892	3893	3894	3895		
7470	3896	3897	3898	3899	3900	3901	3902	3903		

TABLE 2-14. OCTONARY-DECIMAL INTEGER CONVERSION (cont'd)

	0	1	2	3	4	5	6	7
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

3.2 ALARMS AND CHECKS**3.2.1 Alarm Actions**

Some program errors which can be detected by circuits and most equipment conditions which could disturb the proper execution of the program are reported as alarm conditions by visual and audible indications on the duplex maintenance console. In addition to these indications, certain conditions can directly influence the program in one or both of two ways:

- By turning on a sense unit (flip-flop) which can be sensed by a BSN instruction
- By generating an alarm signal which causes the program to stop or branch to register 20010₈ (test memory), depending upon the settings of certain switches.

The conditions that turn on an associated sense unit which can be sensed are:

- Memory parity error
- Drum parity error
- Tape parity error
- Overflow in left accumulator
- Overflow in right accumulator

f. Output alarm caused by at least one of the following conditions:

- Parity error in word sent from output drum to Output System
- Error in message transmitted by Output System
- Illegal section address
- Illegal register address
- Too many words tagged for one burst

g. Other computer alarm No. 1, caused by at least one of the following conditions in the other computer:

- Memory parity error
- Overflow in left or right accumulator

h. Other computer alarm No. 2 caused by at least one of the following conditions in the other computer:

- Drum parity error
- Tape parity error

i. Switching computer to active status.

The computers to which these items apply and the associated sense codes are listed in table 2-15.

TABLE 2-15. ALARMS WHICH MAY INFLUENCE PROGRAM

ALARM NAME	CONDITION	SENSE CODE	SIGNAL TO ALARM CONTROL SWITCH	
Memory Parity	Memory Parity Error	15 ₈	Yes	
Drum Parity	Drum Parity Error	16 ₈	Yes	
Tape Parity	Tape Parity Error	17 ₈	Yes	
Left Overflow	Overflow in Left Accumulator	12 ₈	Conditional*	
Right Overflow	Overflow in Right Accumulator	13 ₈		
Output	Output Drum Parity Error	51 ₈	33 ₈ ***	No
	Error in Transmitted Message	53 ₈		
	Illegal Section Address	52 ₈		
	Illegal Register Address			
	Too Many Words Tagged for One Burst (Drum Comparison Error)	50 ₈		
Other Computer Alarm No. 1	Memory Parity Error	41 ₈	No	
	Overflow in Left Accumulator			
	Overflow in Right Accumulator			
Other Computer Alarm No. 2	Drum Parity Error	42 ₈	No	
	Tape Parity Error			
Duplex Switch	Computer Switch to Active Status	30 ₈ **	Yes	

*Alarm signal conditional on overflow alarm control bits; sense units always respond.

**Sense unit not turned off by BSN instruction.

***Sense code 33₈ determines whether any of the listed conditions occurred.

The conditions which generate an alarm signal causing the program to stop or to branch are:

- Memory parity error
- Drum parity error
- Tape parity error
- Overflow in left or right accumulators, provided corresponding overflow alarm control bits are 1's
- Switching computer to active status.

A switch is associated with each of the five alarm signals just listed. If one of these switches is in the INACTIVE position, the corresponding alarm signal is ignored. If the switch is in the ACTIVE position, the resulting action is determined by a common STOP-BRANCH switch. However, since there is only one STOP-BRANCH switch, all alarms whose switches are set to ACTIVE must result in the same action.

If the STOP-BRANCH switch is in the STOP position, an alarm signal causes the following actions:

- The program is stopped immediately, and the current instruction may not be completed.
- Any IO transfer in progress is interrupted. The internal memory unit involved is effectively disconnected, and the IO address counter stops. Except in the case of a tape unit, the IO unit tries to complete the transfer, and the IO word counter continues to count. Words read from an IO unit are simply discarded; however, if words are read by status from a drum, they are permanently lost. If the transfer is to a drum, the remainder of the block of words after the alarm are words of +0. If the transfer is to or from a tape unit, a memory parity error or tape parity error immediately stops the transfer of words. The IO word counter is stopped, and the tape unit is deselected. If the tape unit is in

the read mode, it comes to rest immediately (not at the end of a record).

If the STOP-BRANCH switch is in the BRANCH position, an alarm signal causes the following actions:

- a. The program is stopped immediately, and the current instruction may not be completed.
- b. The contents of the program counter are transferred to the right A register. The contents are one greater than the address of the instruction being executed when the alarm occurred.
- c. Any IO transfer in progress is stopped immediately, the IO unit is stopped and deselected (a tape unit being read stops at the end of a record), and the IO interlock is cleared.
- d. All registers and controls (except the right A register, the IO word counter, and alarm indicators and alarm sense units) are cleared.
- e. The program continues with the instruction in register 20010₈.

3.2.2 Parity Checking

A parity error results from the reading of a word of incorrect parity from a unit which possesses a parity check facility. Correct parity of a 33-bit word (a 32-bit computer word plus a single parity bit) is defined as the sum of the 1's in the word being odd.

Correct parity bits are generated automatically when a block of words is transferred into core memory from a unit not possessing parity check facilities. However, correct parity bits are not generated when transferring a block of words from test memory to any parity-checked unit. When a word is transferred from one parity-checked unit to another, the parity of the word is unchanged regardless of whether it is correct or not. The only means of restoring correct parity to a word of wrong parity are:

- a. To get the word into the accumulators and to execute an *FST* instruction (although the word probably will continue to contain defective information)
- b. To store another word on top of the defective word, either by a block transfer or a *Store* class instruction.

The Central Computer System can perform parity checks on words received from or supplied to the following units to detect transfer errors caused by equipment malfunction:

- a. Core memory
- b. Magnetic tapes
- c. Auxiliary memory drum fields
- d. Crosstelling input drum fields
- e. Digital display field
- f. Gap-filler input drum field

- g. Intercommunication drum fields
- h. Long-range radar input drum fields
- i. Manual data input drum field
- j. Output buffer drum fields
- k. Spare drum fields
- l. Track display drum fields.

Parity checking of the gap-filler input and manual data input fields is done only for test purposes and does not automatically occur during normal computer operations. The long-range radar does have facilities for parity checking telephone-line messages. In addition, words read from output buffer drum fields on the OD side are parity-checked by the Output System. No other units or registers provide for parity checking.

A misleading situation can arise in connection with the generation of a parity alarm. A parity alarm may occur when a word is read from core memory by an instruction even if the instruction does not use the memory word. Instructions which read a word from core memory, check the parity, but do not use the word are: *HLT*, *PER*, *ADB*, *SEL*, and *SDR*. The fact that *ADB*, *SEL*, and *SDR* are indexable may also be of consequence.

3.3 MISCELLANEOUS OPERATING CONTROLS

3.3.1 Automatic Branch to Zero

If the BRANCH NORMAL-BRANCH TO ZERO MEMORY switch is in the BRANCH TO ZERO MEMORY position, every branch instruction which is active and which has the contents of LS equal to 0 causes the program to branch to memory address 0. As in a regular branch operation, the original contents of the program counter are transferred to the right A register. This automatic branch provision makes possible the high-speed tracing of a program by causing the program to branch to a trace routine on every active branch instruction. No branch to 0 occurs for branch instructions in the trace routine itself since each branch instruction, in this case, is exempted by having the contents of LS equal to 1. When the COMPUTER OPERATE-TEST switch is in the OPERATE position, automatic branching does not occur.

3.3.2 Test and Operate Modes

The Central Computer System has two basic modes of operation: test and operate. The COMPUTER OPERATE-TEST switch on the duplex maintenance console normally selects the mode. When a computer is switched to active status by the duplex switch, its COMPUTER OPERATE-TEST switch can be overridden to put the Central Computer System in the operate mode. The Central Computer System is switched to the operate mode during the execution of programs other than those concerned with testing and maintenance of the equip-

ment. In the operate mode, most of the facilities which could interfere with normal Central Computer System operation are disabled. This includes pushbuttons and switches used for testing and maintenance, the marginal checking controls, and other circuits incorporated exclusively for test purposes.

3.3.3 Duplex Switching

Duplex switching is accomplished at one of two duplex switching consoles by the depression of a push-button indicating which computer (A or B) is to be made active. The duplex switching operation normally cannot take place until the following conditions are true of the computer chosen to become active:

- a. Power is on all units.
- b. Special jumper is in test memory plugboard.
- c. Magnetic tape is not conducting hazardous test.

If it is necessary to make a computer active when all the above conditions are not met, this can be done while the INTERLOCK BYPASS switch is held. A computer in active status normally overrides the COMPUTER

OPERATE-TEST switch and puts the Central Computer System in the operate mode. However, if the Central Computer System in the operate mode needs to be put in the test mode (in order to perform marginal checking, for example), this can be done with the TEST OPERATE OVERRIDE BYPASS switch.

Switching a computer to active status performs the following actions:

- a. Sends a duplex switch alarm to the associated DUPLEX SWITCH (inactive)-ACTIVE switch on the duplex maintenance console
- b. Clears the output buffer drum
- c. Turns on the duplex switch active sense unit in the active computer, and turns off the corresponding sense unit in the standby computer. (Each sense unit is sensed by a BSN (30)₈ x instruction which does not turn off the sense unit.)
- d. Sounds an audible alarm throughout the entire computer area.

CHAPTER 3

PROGRAMMING TECHNIQUES

SECTION 1

GENERAL

1.1 INTRODUCTION

This chapter presents some of the considerations necessary in preparing programs, whether operational or maintenance, for AN/FSQ-7 Combat Direction Central. The basic techniques involved in program preparation have already been described. (Refer to Part 1, Ch 3, Sec 5.) However, Section 1 of this chapter presents a discussion of techniques generally used in preparing programs. Section 2 presents a discussion of the fundamentals of scaling in fixed-point digital computers, of which the Central Computer System is one.

1.2 ALTERNATE ROUTINE SELECTION

1.2.1 Operational Programs

When a library of routines is available for the preparation of an operational program, it is usually necessary to select one routine out of several which perform the same function. (It may also be necessary to examine each available routine in order to decide whether a special routine is needed for the program being prepared.) The criteria used to select one routine out of several similar routines (or to decide to write a new routine) are varied, depending upon the requirements of the using program.

The criteria used in selecting routines exist as opposable pairs. The criterion of program execution time must usually be balanced against the amount of memory space required for storage of the routine in question. For example, conversion of co-ordinates from polar to rectangular, or from rectangular to polar, form can be done quite rapidly by using table look-up routines. However, table look-up routines require a large amount of memory space to store trigonometric or inverse trigonometric tables. On the other hand, conversion routines can be written to generate trigonometric functions from angular measurements, and vice versa; these routines require less memory space than do the table look-up routines but take more time in execution.

A third criterion, precision, must be balanced against the time-space criteria just mentioned. For example, a table look-up routine for co-ordinate conversion can attain high precision at the expense of memory space; the higher precision requires more entries in

the table of trigonometric (or inverse trigonometric) functions and, hence, more memory space. On the other hand, the function generation routine can attain higher precision at the expense of execution time. In general, function generation requires a successive approximation approach to the correct result; for higher precision, a greater number of approximations and, therefore, more time for calculations is required. Further, the degree of complexity of routines usually varies directly with the precision attained by them.

In addition to all these criteria, the choice of a routine for a particular function may be influenced by the possibility of using that same routine to perform a similar function at another point in the program. If the two functions are similar enough, the same routine can be used in both places, either as an iterative routine or as a close subroutine. If these alternatives are not possible, it may be desirable to adapt a standard routine (by inserting a program-set parameter) for use in both places in the main program. If this is impossible, it may be desirable to write a new routine which is adaptable for both uses. In general, the use of two routines where one can be utilized is to be avoided in operational programs.

1.2.2 Maintenance Programs

Perhaps the overriding criterion in the preparation of maintenance programs is clarity. Maintenance programs are used as tools by personnel who are not themselves programmers; they must be able to understand the purpose and operation of a maintenance program if they are to use it intelligently. Therefore, if a choice exists between a test routine which is relatively complex and another which, although longer, is relatively straightforward in its approach to the test to be performed, the straightforward program is preferable (assuming that it does not make the overall maintenance program impossibly long). Particular attention must be given to the comments and descriptions which accompany all written maintenance programs, for these accomplishments and descriptions can greatly facilitate the understanding of maintenance programs by personnel who use them.

1.3 CHECKING

The accuracy (validity of result as opposed to precision) of any program result is dependent upon several factors. First, the data on which the program operates must be correct if the result is to be correct. The validation of air defense data is performed partially by facilities included in the Input System and supplemented by data analysis performed by the Central Computer System. Second, the program which performs data-processing must itself have been validated before being put into use, to ensure that the program does what it is expected to do with the data presented to it. Finally, the operation of the computing equipment must be validated before the accuracy of the program result can be accepted. This last consideration is partially satisfied by the maintenance operations performed on a computer while it is in standby status and by the alarm detection facilities built into the equipment. It can be further satisfied by including within a data-processing program certain routines which check the validity of the results of preceding routines within the program. The amount of checking which can be incorporated in an operational program is, of course, limited by the time and space which can be devoted to checking. Further, checking should be incorporated into a critical program, one whose results are used by, and would influence, a large number of other programs but not necessarily into a program whose results can be monitored in some other way and whose errors may not be critical.

A mathematical program can include checks on intermediate results by duplicating calculations on one

item of data and comparing the two results so obtained. Another type of check might involve repeating a calculation with the complement of an item of data and testing for zero produced by adding the result of the first calculation to the result of the second. A third type of check is possible if the function to be performed can be done by two different methods; both methods are used, and the results are compared. Finally, known data can be introduced for processing and the result compared with the pre-determined correct result.

The exact action taken when a check reveals an error depends upon the significance of the information to the air defense function. It may be sufficient to maintain a record of the number of errors per unit time. If this error rate exceeds some predetermined limit, an alarm can be generated or information on the type of error can be displayed at the duplex switching console. In general, it is desirable to have the detection of an error cause the program to repeat the block of operations within which the error occurred. If the error is caused by a transient equipment failure, the second pass should produce a valid result, and the only time lost is that taken to repeat the block. If the error is repeated on the second pass, some other action is required.

No specific directions can be given as to the extent to which checking should be included in any given program or the action to be taken upon detection of an error. Both these factors can be determined only by weighing the significance of a particular program within the air defense function against the time and space required to include checks and take actions on them.

SECTION 2

SCALING

2.1 PURPOSE AND DEFINITION

The term scaling is defined as placing the binary point in a binary number so that powers of 2 are made to correspond to each binary position of the number. In effect, placing the binary point defines the scale by which the binary number measures magnitude.

Scaling a problem more generally connotes initially defining the position of the binary point in each of the input numbers on the basis of the bounds on their magnitudes, following the behavior of the point in all the computational steps (i.e., the position of the point resulting from multiplication and division, again taking into account the bounds on the magnitudes of the intermediate answers) and, finally, knowing the position of the binary point in each of the final results. Fixed point scaling generally requires complete and accurate information about the bounds on the magnitudes of all numbers, input, intermediate, and output, that come into the computation.

A careful scaling analysis is necessary to make the most efficient use of the 15 bits in the binary number (to minimize the presence of leading zeros, those between the binary point and the first 1 bit, thus increasing precision) and, at the same time, to provide for the largest numbers without the occurrence of overflow (loss of the most significant bits in a number).

2.2 NOTATION

It is convenient to write the symbols for the numbers in a problem in a form which explicitly states the position of the binary point, as:

$$x = 2^q x$$

where

x is the true value

x is a 15-bit fraction (point before the first bit), the scaled form of x

2_q is the scale factor, generally the smallest integral power of 2 which is greater than the maximum value of x .

Conventionally, all binary numbers are considered as fraction. The numbers in a problem are expressed as fraction multiplied by scale factors. The 15 bits in a binary half-word can be considered in two ways. With the binary point at the extreme left, it is x , the scaled fraction. With the point q places over from the left, it is x , the true value. In going from x to x , or vice versa, the point moves but the binary bits remain the same.

2.3 CONSIDERATIONS

Neither addition nor subtraction changes the position of the point. However, both in multiplication and division, special consideration must be given to the place of the point in the product and in the quotient. The engineering design of the Central Computer System is such that the following rules apply.

Multiplying two 15-bit fractions yields a product which is a 31-bit fraction. The rule, stated more generally, is: If q is the number of binary places to the left of the point in the multiplier and q' is the number of places to the left of the point in the multiplicand, then the 31-bit product has $q + q'$ places to the left of the point. Actually, the 31st bit in the product is not relevant (always a true value zero) since the product of two 15-bit terms cannot exceed 30 bits.

In division, the divisor must be larger than the dividend. If so, a 31-bit fractional dividend divided by a 15-bit fractional divisor yields a 16-bit fraction for a quotient. Generally, the dividend is a 15-bit fraction; the 31-bit register space is used to shift the dividend to the right the number of places required to make it smaller than the divisor.

The accumulator sign, after division, is the sign of the quotient. The point in the remainder is the same as it was originally in the dividend, i.e., the 15 bits of the remainder correspond to the last 15 bits in the dividend (the 17th through 31st places to the right of the point). However, the position of the remainder (in the accumulators) indicates that the remainder is scaled up by 2^{15} . Similarly, the quotient (in the B registers) is scaled down by 2^{-15} . These scale factors introduced by position can be eliminated by a shift operation. Ignoring the scale factors due to position, the general rule for the point in division is: if q is the number of binary places to the left of the point in the dividend and q' is the number of places to the left of the point in the divisor, then the 16-bit quotient has $q - q'$ places to the left of the point. If $q - q'$ is a negative integer, then that number of zeros (or ones in the case of a negative quotient) must be inserted between the 16 bits in the B register quotient and the binary point; i.e., the quotient has $16 + |q - q'|$ bits to the right of the point.

The Central Computer System can thus divide correctly any pair of 15-bit terms (except where the

divisor is zero) even though the 15 bits in the divisor must exceed the first 15 bits in the 31-bit dividend.

2.4 PROCEDURE

The steps to follow in scaling are the following:

1. Ascertain the bounds on the absolute values of numbers.
2. Set up the scaling relationship between true numbers and scaled fractions by determining the required scale factor 2^a in the general equation,

$$x = 2^a x.$$

3. By substitution, obtain the scaled value formula from the true value formula, and write the program directly from the former. The scale factors which do not cancel specify the required shift operations.

2.5 EXAMPLES

2.5.1 Multiplication Scaling

Consider the equation $xy = z$ with the following bounds on the absolute values of the terms:

$$/x/ < 250$$

$$/y/ < 50$$

$$/z/ < 4,000$$

It frequently occurs that additional information is available which puts a bound on $/z/$ which is less than the calculated bound (maximum x times maximum y). In this case, advantage can be taken of the fact that $/z/$ has an effective bound of 4,000 in place of 12,500.

Next, the scale factors are determined:

$$x = 2^8 x \quad (2^8 = 256)$$

$$y = 2^6 y \quad (2^6 = 64)$$

$$z = 2^{12} z \quad (2^{12} = 4096)$$

Note that, in each case, the scale factor is the smallest integral power of 2, which is greater than the absolute bound. In effect, the point is placed in the half-word so that no more binary places are carried to the right of the point than are necessary to represent the absolute bound.

Finally, the scaled equation is written:

$$xy = z$$

$$2^8 x 2^6 y = 2^{12} z$$

This scaled equation reduces to

$$2^2 xy = z$$

Coding directly in terms of the reduced equation, the routine for multiplication is written:

CAD L (x)

MUL L (y)

SLR 2

FST L (z)

In the first two steps, two fractions are multiplied to generate a 31-bit fractional product. Because the effective bound of $/z/$ is 4,000, not 12,500, two leading binary zeros will always appear in the product. The left shift of 2 eliminates these zeros and, at the same time, brings two additional bits of precision from the B register. The last step stores a 15-bit rounded fraction as the product, z .

Logically, the 15 bits in the core memory register containing the product can be considered in two ways. With the binary point at the extreme left, it is z , the scaled fraction. With the point 12 places over from the left, it is z , the true answer. The scaling relationship

$$z = 2^{12} z$$

explicitly states the equivalence of the two views.

2.5.2 Division Scaling

Consider the equation $\frac{x}{y} = z$ with the following absolute bounds on the terms:

$$/x/ < 1000$$

$$2 < /y/ < 50$$

$$/z/ < 200$$

The lower bound as well as the upper bound must be known for all divisors. The bounds on $/x/$ and $/y/$ imply an absolute bound on $/z/$ of 500. As it occurs frequently, a lower figure of 200 is given as a usable bound in this example. Advantage is taken of this in scaling to achieve greater precision to the right of the point.

The scale factors for the numbers are determined as follows:

$$x = 2^{10} x \quad (2^{10} = 1024)$$

$$y = 2^6 y \quad (2^6 = 64)$$

$$z = 2^8 z \quad (2^8 = 256)$$

Finally, the scaled equation is written as follows:

$$\frac{x}{y} = z$$

$$\frac{2^{10} x}{2^6 y} = 2^8 z$$

This equation reduces to

$$\frac{2^{-4} x}{y} = z$$

The division operation is then coded as follows:

CAD L (\bar{x})

DSR 4

DVD L (\bar{y})

SLR 17₈

FST L (\bar{z})

The right shift of 4 derived from the bounds on x, y, and z sets up the machine division so that the divisor exceeds the dividend. The fraction $2^{-4} \bar{x}$ is then divided by the fraction \bar{y} to obtain the fractional quotient \bar{z} . It has been assumed that the original bits present in the B register introduce an insignificant inaccuracy. If necessary, the B register can be cleared before the division operation proper.

2.5.3 Accumulation Scaling

2.5.3.1 General Procedure

Numbers which are added or subtracted in the accumulator obviously must have the same scale. However, in order to prevent an overflow in the summing process, it is not enough to scale the final sum according to its bound, but generally (unless special shift operations are included) the final sum must be scaled by the largest bound which applies to any element in the sum (any partial sum generated in the process of summing).

In order to scale the sum

$$A = \sum_{i=1}^n a_i$$

the following is usually done, given the bounds:

$$/A/ < A'$$

$$/a_i/ < a'_i \quad i=1, 2, \dots, n$$

Select the largest bound from A' , a'_1 , a'_2 , \dots , a'_n , and know the bound for any partial sum of a_i terms. The larger of these two bounds, say A'' , is then used as the effective bound to scale both the sum A and the elements a_i . This procedure is generally used in place of carrying separate scale factors which require shifts to align binary points for addition. The latter procedure may be used if the maximum precision attainable for the a_i terms must be carried through the summation for use in later calculations.

2.5.3.2 Simple Addition

Consider the equation

$$A = a_1 + a_2 + a_3 + a_4$$

given that:

$$/a_1/ < 200$$

$$/a_2/ < 300$$

$$/a_3/ < 600$$

$$/a_4/ < 700$$

$$/A/ < 1000$$

$$/\text{partial sum}/ < 1000$$

The scale factors are determined as follows:

$$a_1 = 2^{10} \bar{a}_1 \quad (2^{10} = 1024)$$

$$a_2 = 2^{10} \bar{a}_2$$

$$a_3 = 2^{10} \bar{a}_3$$

$$a_4 = 2^{10} \bar{a}_4$$

$$A = 2^{10} \bar{A}$$

The routine is as follows:

CAD L (\bar{a}_1)

ADD L (\bar{a}_2)

ADD L (\bar{a}_3)

ADD L (\bar{a}_4)

FST L (\bar{A})

2.5.3.3 Compound Multiplication and Division

Consider the equation

$$A = \frac{a_1 a_2 a_3}{a_4 a_5}$$

given that

$$/a_1/ < 100$$

$$/a_2/ < 50$$

$$/a_3/ < 70$$

$$5 < /a_4/ < 40$$

$$10 < /a_5/ < 20$$

The implied maximum bound for A is

$$/A/ < 7,000$$

The terms are scaled

$$a_1 = 2^7 \bar{a}_1 \quad (2^7 = 128)$$

$$a_2 = 2^6 \bar{a}_2 \quad (2^6 = 64)$$

$$a_3 = 2^7 \bar{a}_3 \quad (2^7 = 128)$$

$$a_4 = 2^6 \bar{a}_4 \quad (2^6 = 64)$$

$$a_5 = 2^5 \bar{a}_5 \quad (2^5 = 32)$$

Specifying the order of computation to be

$$\frac{a_1}{a_4}, \left(\frac{a_1}{a_4} \right) a_2, \left(\frac{a_1 a_2}{a_4} \right), \left(\frac{a_1 a_2}{a_4 a_5} \right) a_3$$

The intermediate terms are scaled

$$\left| \frac{a_1}{a_4} \right| < 20 \quad \frac{a_1}{a_4} = 2^5 \left(\frac{\overline{a_1}}{a_4} \right)$$

$$\left| \frac{a_1 a_2}{a_4} \right| < 1000 \quad \frac{a_1 a_2}{a_4} = 2^{10} \left(\frac{\overline{a_1 a_2}}{a_4} \right)$$

$$\left| \frac{a_1 a_2}{a_4 a_5} \right| < 100 \quad \frac{a_1 a_2}{a_4 a_5} = 2^7 \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right)$$

Converting into fractional form

$$2^5 \left(\frac{\overline{a_1}}{a_4} \right) = \frac{2^7 \overline{a_1}}{2^6 a_4} \text{ reduces to } \left(\frac{\overline{a_1}}{a_4} \right) = \frac{2^{-4} \overline{a_1}}{a_4}$$

$$2^{10} \left(\frac{\overline{a_1 a_2}}{a_4} \right) = 2^5 \left(\frac{\overline{a_1}}{a_4} \right) 2^6 \overline{a_2} \text{ reduces to } \left(\frac{\overline{a_1 a_2}}{a_4} \right) = 2^1 \left(\frac{\overline{a_1}}{a_4} \right) \overline{a_2}$$

$$2^7 \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right) = \frac{2^{10} \left(\frac{\overline{a_1 a_2}}{a_4} \right)}{2^5 a_5} \text{ reduces to } \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right) = \frac{2^{-2} \left(\frac{\overline{a_1 a_2}}{a_4} \right)}{a_5}$$

Finally

$$2^{13} \overline{A} = 2^7 \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right) 2^7 \overline{a_3} \text{ reduces to } \overline{A} = 2^1 \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right) \overline{a_3}$$

Coding directly from the fraction equations for each intermediate result of calculation gives:

$$\left(\frac{\overline{a_1}}{a_4} \right), \left(\frac{\overline{a_1 a_2}}{a_4} \right), \left(\frac{\overline{a_1 a_2}}{a_4 a_5} \right), \text{ and } \overline{A}$$

The resulting routine is:

CAD	L	($\overline{a_1}$)
DSR		4
DVD	L	($\overline{a_4}$)
SLR		17 ₈
MUL	L	($\overline{a_2}$)
DSR		1
DVD	L	($\overline{a_5}$)
SLR		17 ₈
MUL	L	($\overline{a_3}$)
SLR		1
FST	L	(\overline{A})

Note how the DSR 1 instruction combines two shifts into one operation.

2.5.3.4 Function Evaluation

Consider the equation

$$P = a + bx + cx^2$$

Given the bounds

$$\begin{aligned} /a/ &< 2 \\ /b/ &< 2 \\ /c/ &< 2 \\ /x/ &< 4 \\ /P/ &< 30 \end{aligned}$$

To reduce computational steps, the equation is put into the form

$$P = a + x(b + cx)$$

which specifies, as the order of computation,

$$cx, b + cx, x(b + cx), a + x(b + cx)$$

Both c and x are used directly, in multiplication and may therefore be scaled according to their own bounds:

$$\begin{aligned} /c/ &< 2 & c &= 2^1 \overline{c} \\ /x/ &< 4 & x &= 2^2 \overline{x} \end{aligned}$$

The term cx is added to b. Hence, the scaling for b, cx, and their sum is determined by the bound

$$/b + cx/ < 10$$

Accordingly

$$\begin{aligned} b &= 2^4 \overline{b} \\ cx &= 2^4 \overline{cx} \\ b + cx &= 2^4 \overline{b + cx} \end{aligned}$$

Finally, noting that P, a and x(b + cx) are involved in the final addition and should have the same scaling, their bounds are determined:

$$\begin{aligned} /P/ &< 30 \\ /a/ &< 2 \\ /x(b + cx)/ &< 32 \end{aligned}$$

Hence, these terms are scaled:

$$\begin{aligned} P &= 2^5 \overline{P} \\ a &= 2^5 \overline{a} \\ x(b + cx) &= 2^5 [\overline{x(b + cx)}] \end{aligned}$$

The fractional forms include:

$$2^4 \overline{cx} = 2^1 \overline{c} 2^2 \overline{x} \text{ which reduces to } \overline{cx} = 2^{-1} \overline{c} \overline{x}$$

$$2^4 [\overline{b + cx}] = 2^4 \overline{b} + 2^4 \overline{cx} \text{ which reduces to } \overline{b + cx} = \overline{b} + \overline{cx}$$

$$2^5 [\overline{x(b + cx)}] = 2^2 \overline{x} [2^4 \overline{(b + cx)}] \text{ which reduces to}$$

$$\overline{x(b + cx)} = 2^1 \overline{x} [\overline{(b + cx)}]$$

$$2^5 \overline{P} = 2^5 \overline{a} + 2^5 [\overline{x(b + cx)}] \text{ which reduces to}$$

$$\overline{P} = \overline{a} + [\overline{x(b + cx)}]$$

These fractional forms all reduce to the final equation

$$\bar{P} = \bar{a} + 2^1 \bar{x} (\bar{b} + 2^{-1} \bar{c} \bar{x})$$

Coding from this expression yields the following routine:

CAD L (\bar{c})

MUL L (\bar{x})

DSR 1

SLR 0

ADD L (\bar{b})

MUL L (\bar{x})

SLR 1

ADD L (\bar{a})

FST L (\bar{P})

A description of this program in general terms is given in Part 1, Chapter 3, Section 5.

CHAPTER 4

IO PROGRAMMING

4.1 INTRODUCTION

This chapter presents information specific to the programming of each IO unit available to the Central Computer System of AN/FSQ-7 Combat Direction Central. Programwise, these IO units are divided into two groups: those selected by a *Select* instruction and the drum fields chosen by the *Select Drum* instruction. The IO units other than drum fields are each discussed separately. The drum fields are discussed primarily as auxiliary memory devices or as time buffers, since the programming considerations for these two types of drum fields differ in several respects. The precautions general to all IO programs are discussed in Chapter 2, Section 3.

4.2 IO UNITS

4.2.1 Card Reader

The card reader (IBM type 713) can be used only in reading operations. It can read a maximum of two words every 15 milliseconds. An *SEL* (01)₈ instruction selects the card reader for use in a subsequent IO operation. It is desirable to give this instruction before the *LDC x* instruction in the sequence of IO class instructions. In this way, more time is left between selection of the card reader and sensing for a not-ready condition with *BSN* (11)₈ *x* just prior to the *RDS n* instruction which initiates reading. A full card is read for each *RDS* 30₈ instruction; if *n* > 30₈, 24 words are read from the first card and as many more words from the next card or cards to make a total of *n* words read. Although not all the words on a card are read, the entire card is passed through the card reader; an *RDS* 0 instruction passes one card through the card reader without reading any words from it. (Refer to Part 1, Ch 3, Sec 4, 4.3.)

4.2.2 Card Punch

The card punch (IBM type 723) can be used only in writing operations. Its maximum punching rate is 100 cards per minute. An *SEL* (02)₈ instruction selects the card punch for use in a subsequent IO operation. It is desirable to give this instruction before the *LDC x* instruction in the sequence of IO class instructions. In this way, more time is left between selection of the card punch and sensing for a not-ready condition with *BSN* (11)₈ *x* just prior to the *WRT n* instruction which initiates writing. If *n* = 30₈, one card image in core memory is punched on one card. If *n* > 30₈, one full

card is punched for each group of 24 words in core memory plus a final card or cards containing the balance of words beyond the last full card image that was punched. If a *WRT* 0 instruction is given, a card is passed through the card punch with no words punched onto it. Two special instructions are available for use with the card punch. The *PER* (73)₈ instruction, if given immediately after the *WRT* instruction which initiates punching on a card, transfers punching of information from columns 17 through 32 to columns 1 through 16. This instruction can be used to cause punching of identification information on a card. The *PER* (74)₈ instruction, if given for each card cycle, causes the duplication, in columns 1 through 16 of the card being punched, of the information in columns 1 through 16 of the card just punched. (Refer to Part 1, Ch 3, Sec 4, 4.4.)

A card punch program which calls for the punching of a number of cards should include a routine which senses for the instant the card punch becomes not ready because its card hopper is empty. The initial detection of the not-ready condition must be followed by another test for that condition after a delay sufficient for various relays to settle in their new positions. (When a relay transfers, some contact bounce occurs; several milliseconds must be allowed for it to settle.) The routine given in table 2-16 illustrates a test for the not-ready condition with the required delay included within the routine. An equivalent routine can be used with the card reader or line printer.

4.2.3 Line Printer

The line printer (IBM type 718) can be used only in writing operations. Its operation is made versatile by the use of a prewired control panel. The line printer is selected for use in a subsequent IO operation by an *SEL* (03)₈ instruction. This instruction is normally given prior to the *LDC x* instruction to allow more time before a not-ready condition is sensed by *BSN* (11)₈ *x*. A *WRT n* instruction initiates the writing operation. Several special instructions are provided for the line printer. These include *Operate* instruction selection codes 51 through 62₈ and *Sense* instruction selection codes 31 and 32₈. Approximately 6.5 to 7 milliseconds elapse between the execution of one of the *Operate* instructions and the appearance of an impulse at the appropriate exit hub. The programmed use of these instructions must take this delay into account. If, for ex-

TABLE 2-16. CARD PUNCH NOT-READY SENSING ROUTINE

LOCATION	INSTRUCTION		COMMENT
	OPERATION	ADDRESS	
00100	<i>BSN</i> (11) ₈	00150	Senses for not-ready condition; if detected, branches to delay and further test.
00101	<i>BSN</i> (14) ₈	00100	Branches, if IO interlock is on, back to previous instruction; during an IO operation, not-ready condition is sensed continuously.
00150	<i>PER</i> (27) ₈		Clears IO interlock; performed only if card punch becomes not ready during IO operation; hangup is thus prevented.
00151	1 <i>XIN</i>	03000	
00152	1 <i>BPX</i> (01) ₈	00152	Delays subsequent operations for 9 milliseconds to allow relays to settle.
00153	1 <i>XIN</i>	10000	
00154	<i>BSN</i> (11) ₈	00156	Senses for not-ready condition again; if present, branches to delay routine; if not present, continues to stop.
00155	<i>HLT</i>	—	
00156	1 <i>BPX</i> (01) ₈	00154	Delay of 73 milliseconds, if not-ready condition exists, before continuing with subsequent program.

ample, a particular *Operate* instruction exit hub is to transfer connections from one set of print magnets to another set, a delay must be provided after the transfer is initiated and before information is applied to the print magnets, or this information will be lost; i.e., the transfer will not be complete. The exact functions of these instructions are determined by the wiring of the line printer control panel. (Refer to Part 1, Ch 3, Sec 4, 4.5.)

4.2.4 IO Register

The IO register, used for reading operations, provides a source of words containing positive zero for clearing all or selected registers in core memory. The IO register is selected as an IO unit by *SEL* (04)₈. No special precautions are required in using the IO register as an IO unit. The clearing operation, starting from the address specified by the *LDC x* instruction, is initiated by the *RDS n* instruction and continues until the memory registers from addresses x to $x + n - 1$ are cleared. (Refer to Part 1, Ch 3, Sec 4, 4.7.)

4.2.5 Manual Input Matrix

The manual input matrix is the time buffer for MDI keyboard messages presented to the Central Computer System. The matrix is selected for reading by the *SEL* (06)₈ instruction. The *LDC x* instruction is required to specify the starting location for the core

memory image of the MDI matrix. The *RDS* 200₈ instruction initiates the reading of 128 words (the entire contents of the MDI matrix). The matrix cannot be read more often than once every 200 milliseconds; an attempt to do so results in loss of information. Transfers from the MDI matrix occur at the rate of one word per 20 microseconds.

4.2.6 Warning Lights Registers

The warning lights registers (part of the Warning Light System) receive their information contents from the Central Computer System during a writing operation. The warning lights registers are selected for an IO operation by an *SEL* (10)₈ instruction. The *LDC x* instruction specifies the location of the first register of the core memory image of the warning lights registers. A *WRT* 10₈ instruction then initiates the transfer of eight words into the warning lights registers in 96 microseconds. (Refer to Part 1, Ch 8.)

4.2.7 Magnetic Tapes

Six magnetic tape units are provided in each computer of AN/FSQ-7 Combat Direction Central. Each tape unit can be used for reading and writing. Selection codes 11 through 16₈ of the *SEL (u)* instruction are used for selection of a specific tape unit for use in an IO operation. Several special instructions are provided for use with the tape element. They are listed, together with

TABLE 2-17. TAPE ELEMENT INSTRUCTIONS

INSTRUCTION CODE	INSTRUCTION	FUNCTION
<i>BSN</i> (10) ₈ <i>x</i>	<i>Sense</i> (tapes not prepared)	Branch to <i>x</i> if not prepared
<i>BSN</i> (11) ₈ <i>x</i>	<i>Sense</i> (IO unit not ready)	Branch to <i>x</i> if not ready
<i>BSN</i> (17) ₈ <i>x</i>	<i>Sense</i> (tape parity)	Branch, if parity error detected in reading from tape, to <i>x</i>
<i>RDS</i> <i>n</i>	<i>Read</i>	Reads <i>n</i> words into memory; sets IO interlock which is cleared at end-of-record gap; <i>RDS</i> 0 skips one record on tape
<i>WRT</i> <i>n</i>	<i>Write</i>	Writes <i>n</i> words from memory; sets IO interlock which is cleared at end-of-record pause; <i>WRT</i> 0 is illegal and can cause hangup
<i>PER</i> (67) ₈ —	<i>Operate</i> (set tapes prepared)	Places selected tape unit in prepared condition; not held up by IO interlock
<i>PER</i> (70) ₈ —	<i>Operate</i> (backspace)	Backs up one record and places tape unit in prepared condition; not held up by IO interlock; sets IO interlock which is cleared when beginning of record is reached
<i>PER</i> (71) ₈ —	<i>Operate</i> (rewind)	Rewinds tape to load point; not held up by IO interlock; sets IO interlock for 60 milli-seconds; before IO interlock is cleared, unit becomes not prepared and remains so until rewind is completed
<i>PER</i> (72) ₈ —	<i>Operate</i> (write end-of-file)	Writes end-of-file mark; not held up by IO interlock; sets IO interlock until completed; leaves unit not prepared

special interpretations of standard IO instructions for the tape element, in table 2-17.

The general precautions and special considerations in programming the tape element are described in Part 1, Chapter 3, Section 4, 4.6.4.

4.2.8 Burst Time Counters

The burst time counters can be read by the Central Computer System for use in establishing the sequence of, and delay in, transmission of output messages. The burst counters are selected for reading by an *SEL* (21)₈ instruction. An *LDC* *x* instruction specifies the location in core memory which will receive the first of the three words read from the burst time counters. An *RDS* 3 instruction initiates the transfer of the three words at 10-microsecond intervals. (Refer to Part 1, Ch 7.)

4.3 DRUM FIELDS

4.3.1 Auxiliary Memory Fields

Reading or writing on auxiliary memory fields is performed by address mode. (The auxiliary drum group, drums C through H, cannot be written during normal operation. An interlock must be operated to erase or write these drums.) The *SDR* (*u*) *r* instruction specifies

the drum field to be used and the address *r* of the first drum register to be read or written. Transfers to or from addressable fields, once started, can occur at a maximum rate of a word every 10 microseconds. Initial delays (for settling of switching circuits and for finding the specified first register) can occupy up to 20 milli-seconds before the first transfer can be accomplished. The rate of transfer to or from addressable drum fields can be modified by using an interleave pattern; an interleave is called for by the index interval of the *RDS* or *WRT* instruction which initiates the transfers. The available interleave patterns are three in number; every 8th, 16th, or 64th register will be used in transfers, depending upon which pattern is selected. (It should be noted that the interleave by 64 is not compatible with the precession pattern used in reading the DD field on the OD side.)

4.3.2 Time Buffer Fields

4.3.2.1 Reading

Reading of time buffer fields by the Central Computer System may be performed by status or status identification. If the status mode is to be used, the *SDR* (*u*) instruction specifies this by giving the field

selection code indicating status operations; the address half of the instruction has no significance in this case. If the same field is to be read by status identification, the index interval of the *SDR* instruction has a different value, indicating reading by status identification, and the address half of the *SDR* instruction specifies the identification bits to be matched by words before they can be accepted. For certain fields (notably the LRI fields), two different identification patterns can be used in reading; specifically, one field selection code specifies reading, using bits R12 through R15 as the identity code, while another selection code for the same field specifies the use of R7 through R15 as the identity code. Thus, the sorting capabilities of reading by status identification are enhanced by this provision. If no sorting is desired, reading by status is used.

4.3.2.2 Writing

Writing by status identification is impossible. In writing by status mode, the *SDR* (*n*) instruction specifies the field to be written and the fact that status mode is to be used. (The only fields written on the CD side by status are the output buffer fields, odd and even.)

4.3.2.3 Number of Words Transferred

In reading by status identification or writing by status, it is usually necessary to determine the number of words transferred. In general, the *RDS n* instruction

which initiates reading calls for the transfer of more words than are expected. The IO operation is terminated by a disconnect signal from the Drum System. Assuming that *t* words have been read and that $t < n$, executing a *CSW* instruction and adding the *RDS n* instruction to it leaves $t + 1$ in the right accumulator.

In writing by status (on the output buffer fields), the *WRT n* instruction usually calls for the transfer of the desired number of words. If, however, there is room on the selected OB field to write only *t* words (where $t < n$), the Drum System generates a disconnect signal terminating the IO operation after *t* words have been written. The Central Computer System must test to discover whether all *n* words have been transferred. With the accumulators cleared, a *CSW* instruction (executed after the IO interlock is cleared) followed by a *BFZ x* will produce a branch to location *x* if all *n* words have been transferred. If only *t* words were transferred, adding the *WRT n* instruction to the accumulators will leave in the right accumulator the term $t + 1$. A new IO program can then be initiated to transfer the remaining words to the other OB field. This new program must include an *LDC x* instruction since, without it, the two words left in the IO register and the Drum write register by the first IO operation would not be transferred to the drum field. (Refer to Part 1, Ch 4, Sec 2.)

CHAPTER 5

INPUT OPERATIONS

5.1 INFORMATION FORMS

5.1.1 Introduction

Input information is supplied to the Central Computer System for processing from the four elements of the Input System. (Refer to Part 1, Ch 5.) The transfer, for the most part, is effected through the Drum System. Radar data is supplied from the long-range radar input (LRI) element and the gap-filler input (GFI) element; information from adjacent AN/FSQ-7 Combat Direction Centrals is processed in the crossteling (XTL) input element; all other input information is inserted via the manual data input (MDI) element. The input elements are temporary storage and processing devices in which the received information is converted into a form that is compatible with the operation of the Central Computer System. This requires that the data be supplied in the form of 33-bit computer words. The words from each element of the Input System are discussed in the following paragraphs.

5.1.2 LRI Message Word Format

5.1.2.1 General

The LRI element receives information in serial form from three sources: search radar data, IFF data, and height-finder replies. Each of the three types of message has additional information added at the element which organizes the information into two 33-bit words for parallel transfer to the two LRI fields on the LOG drum of the Drum System. (See fig. 2-2.)

5.1.2.2 Search Radar

The layout of a search radar message, as it is placed on a drum field, includes two drum words, with the range and azimuth of the target contained in the second word. Bits L1 through L10 designate, in digital form, the range of the target in 1/4-mile increments. The azimuth, in increments of 1/4,096-revolution of the radar scan, is contained in bits L11 through L15 for the five least significant digits, with the balance of the azimuth data in bits R9 through R15. The rest of word two contains clock time (R1 through R6) added in the LRI element and expressed in 1/4-second increments. The balance of word 2 (LS, RS, R7, and R8) is unused. Word one contains three bits designating the run length of the target, five bits of time delay to 1/4-second precision, and five bits of message label information re-

ceived from the radar site. The identity of the site is added at the LRI element (R12 through R15).

5.1.2.3 IFF

The IFF message is similar to the search radar message but with the addition of the 14-bit selective identification feature (SIF) in bits L1 through L13 of word one. Run length is not used in the IFF message.

5.1.2.4 Height Reply

The height-reply message is also contained in two words; however, since there is less information than in a search or IFF message, some bit positions are not used or contain zeros. The positions not used include bits LS and L14 through R1 of word 1 and LS and RS through R8 of word 2. The positions containing zeros include bits L1 through L13 and R2 through R6 of word 1. The message label and the site identity are contained in word 1, but the significant height information is contained in word 2. The new height, specified in units of 100 feet from sea level, is carried by eight bits divided into two sections with the least significant bit standing alone in position L11 and the balance in R9 through R15. Other height-reply information (with its position in word 2) includes the number of the height request being answered (L13 through L15), any special replies (L1 and L2), the formation (L3 through L5), the separation (L6 and L7), and the number of aircraft (L8 through L10). The identity of which of the two height finders at the reporting site is supplying the information is shown as the address (L12).

5.1.3 GFI Message Word Format

One drum field is provided for GFI messages, each supplied in a single 33-bit word. The GFI element receives information from a radar site and converts the information to a binary representation of the range and azimuth of a target. (See fig. 2-3.) The range, in 3/4-mile increments, is designated by bits LS through L7; the azimuth, in 1.4 angular degrees (1/256 revolution), is indicated by bits RS through R7. The identity of the reporting radar site is added in the GFI element (R11 through R15), and the clock time (L10 through L14) is inserted in multiples of 1/4 second. The other six bit positions in the word are not used.

5.1.4 XTL Message Word Format

The XTL element receives information via telephone equipment from adjacent centrals in the form of

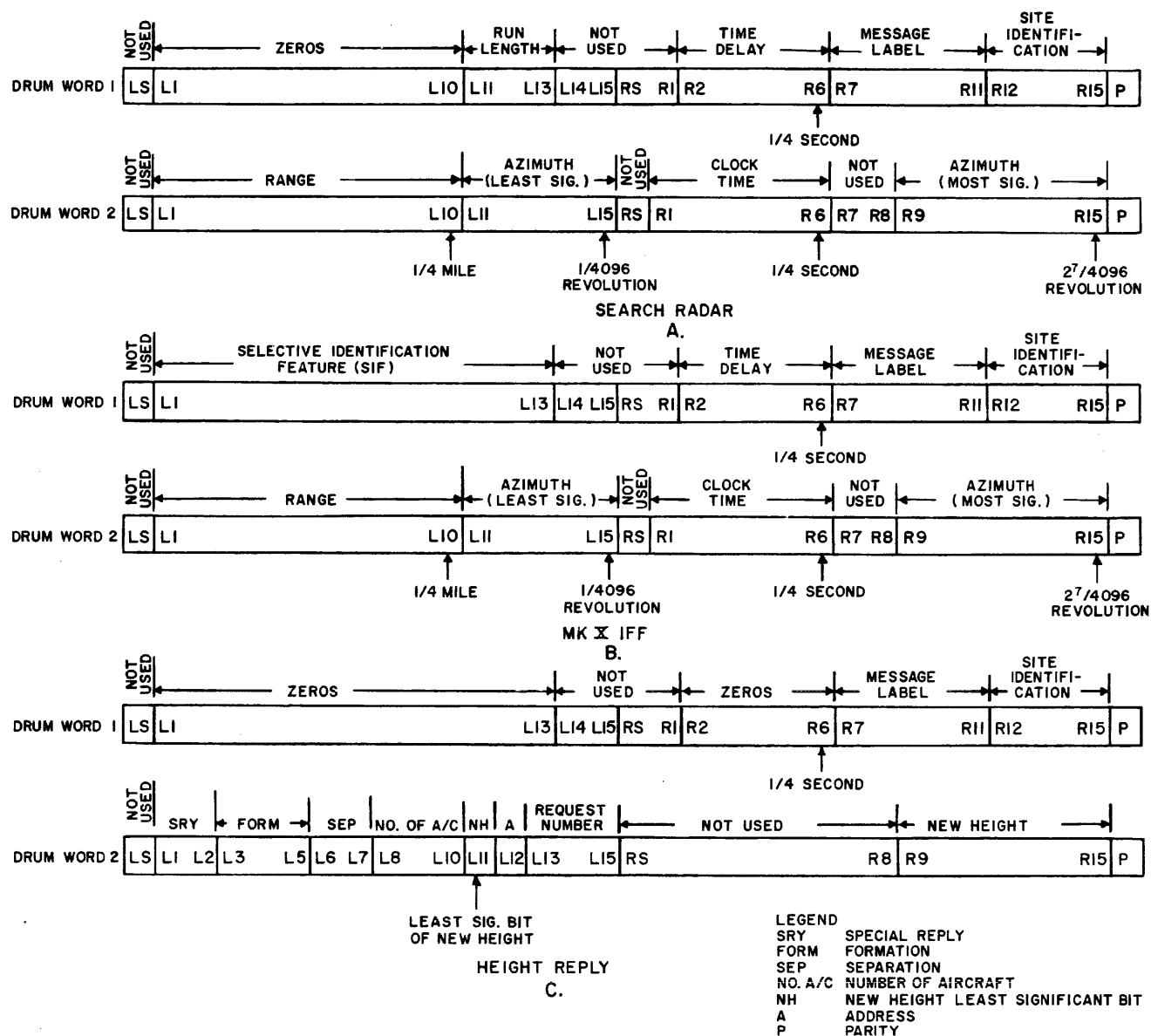


Figure 2-2. LRI Element Message Formats

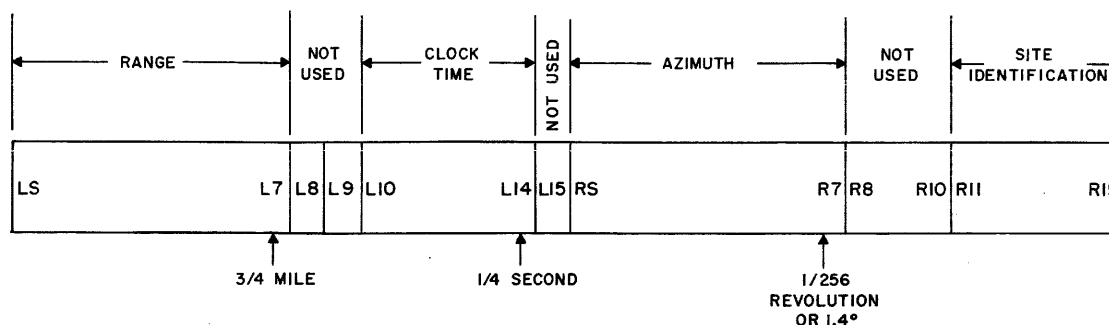


Figure 2-3. GFI Element Message Format

a five-word message. Each word sent in a serial-interleaved manner is 16 bits long and is termed an incoming message word. The information is reassembled into

three drum words in the XTL element, which also adds clock time and site identity bits to the drum words before transferring them in parallel to the XTL field

of the MIXD drum. (See fig. 1-124.) Incoming message word 1 is placed in drum word 1 (LS through L15); incoming message words 2 and 3 form drum word 2; incoming message words 4 and 5 constitute drum word 3. Included in incoming message word 4 is the address of the receiving central (R11 through R15). The identity of the reporting central and the clock time in $\frac{1}{4}$ -second intervals are added to drum word 1 by the XTL element in bits R11 through R15 and R6 through R10, respectively.

5.1.5 MDI Message Word Formats

5.1.5.1 General

The MDI element allows for the manual insertion of data, for manual intervention in the air defense program, or for manual initiation of the transfer of target identification to the Central Computer System. (See fig. 1-130.) Manual insertion is made with the use of punched cards; the other interventions are achieved by the operation of keyboards, light guns, or area discriminators.

5.1.5.2 Computer Entry Punch

The information supplied to the MDI drum field by a computer entry punch is contained in one-word messages. A 32-bit word contains the information from one card column (LS through L11) and the column count (R7 through R13). The card count is indicated by a code that places a single 1 in any of five bit positions (R2 through R6). The computer entry punch that inserts the word is identified by a 1 in any of the four identity bit positions (L12 through L15). Bits R14 and R15 identify the word as card-originated by the 00 code.

5.1.5.3 Radar Data Word

A radar data word, containing X and Y target coordinates, is transferred to the MDI drum field by a light gun or an area discriminator signal. The data are either correlated or uncorrelated, as indicated by a code in the word (R14 and R15). The console is identified by a code which contains four 1's out of eight bits (R6 through R13). The X co-ordinates are indicated in bits LS through L10; the Y co-ordinates are indicated in bits L11 through R5.

5.1.5.4 Track Display Word

A track display word is identified by the code 01 (R14 and R15). The console which initiated its transfer is identified in the four-out-of-eight console identity code (R6 through R13). The balance of the word is composed of the SD character selection codes for the alphabetical and/or numerical characters in positions C_1 through C_4 of the track display message being referred to the Central Computer System. Characters C_1 and C_2 , alphabetical symbols, are identified in bits LS through

L5 and L6 through L11, respectively. The C_3 character (R2 through R5) is numerical; C_3 , either alphabetical or numerical, is identified in bit positions L12 through R1 of the track display word.

5.1.5.5 MDI Matrix Core Assignments

Situation display information supplied to the Central Computer System via the MDI drum field is identified by an eight-bit code generated by the encoder matrix of the manual data input element. The code allows for ease in checking errors in that four of the eight bits are always 1's and the other four are 0's. Should more or less than four 1's appear as a console identification code, the Central Computer System detects the error. The code is unique for the light gun at each situation display console. The codes obtained from the encoder matrix and the light gun core assignments in the core matrix are given in table 2-18. These light gun cores constitute the first two words in the matrix. Words 3 through 6, inclusive, store the action bits whose assignments are shown in table 2-19. The unit status cores constitute two more rows, as listed in table 2-20. These cores indicate to the active Computer which units of the system are in active status and which units are in stand-by. The remaining cores of the 128 total words are assigned to information bits, as shown in table 2-21.

TABLE 2-18. LIGHT GUN CODES AND CORE ASSIGNMENTS

WORD	BIT POSITION	CONSOLE	ENCODER MATRIX CODE
1	LS	W-81	00001111
	L1	W-90	01100011
	L2	W-00	10000111
	L3	W-20	01010011
	L4	W-40	00100111
	L5	W-60	01110001
	L6	W-86	00101011
	L7	W-88	01101001
	L8	T-10	00110101
	L9	T-12	01101010
	L10	Spare	00111010
	L11	E-10, E-60	01001101
	L12	W-03	00110110
	L13	W-05	01001110
	L14	W-07	10100110

TABLE 2-18. LIGHT GUN CODES AND CORE ASSIGNMENTS (cont'd)

WORD	BIT POSITION	CONSOLE	ENCODER MATRIX CODE
	L15	W-11	11000101
	RS	W-13	00011101
	R1	W-23	01110100
	R2	W-25	00011110
	R3	W-29	11001100
	R4	W-31	10010101
	R5	W-33	11001001
	R6	W-44	10011010
	R7	W-46	11100010
	R8	W-48	10101010
	R9	W-52	11010001
	R10	W-54	10011100
	R11	W-64	11010010
	R12	W-66	10110100
	R13	W-70	11011000
	R14	W-72	10110001
	R15	W-74	11101000
2	LS	S-72	11110000
	L1	S-75	11000110
	L2	S-70	11100001
	L3	S-00	11001010
	L4	S-01	11100100
	L5	S-02	10001110
	L6	S-03	11010100
	L7	S-04	10010110
	L8	S-60	10101100
	L9	S-63	01010110
	L10	S-67	01011100
	L11	S-11	10110010
	L12	I-03	01101100
	L13	I-02	01110010
	L14	I-04	01100101

TABLE 2-18. LIGHT GUN CODES AND CORE ASSIGNMENTS (cont'd)

WORD	BIT POSITION	CONSOLE	ENCODER MATRIX CODE
	L15	W-82	10100011
	RS	S-33	10111000
	R1	S-53	00101110
	R2	S-35	01111000
	R3	S-55	001100111
	R4	S-20	10101001
	R5	S-22	10010011
	R6	S-23	01011001
	R7	S-25	01000111
	R8	S-30	01010101
	R9	S-32	10001011
	R10	S-40	00111001
	R11	S-42	01001011
	R12	S-43	00101101
	R13	S-45	00011011
	R14	S-50	10001101
	R15	S-52	00010111

TABLE 2-19. ACTION BIT CORE ASSIGNMENTS

WORD	BIT POSITION	CONSOLE
Information not available		

TABLE 2-20. UNIT STATUS CORE ASSIGNMENTS

WORD	BIT POSITION	UNIT
Information not available		

TABLE 2-21. INFORMATION CORE ASSIGNMENTS

WORD	BIT POSITION	ASSIGNED TO
Information not available		

5.2 TRANSFER TO CENTRAL COMPUTER SYSTEM

5.2.1 General

The information processed by the Input System (except the information processed by the direct IO buffer entry section of the MDI element) is placed on fields of the Drum System for subsequent block transfer to the Central Computer System. This information is automatically sorted into four groups (each group consists of the information from one element) by the equipment design which causes each element to place its information on a separate drum field. (The LRI element places its information on two drum fields. This information is still distinct from the information from the other three input elements on separate fields.)

Since all like items of information from a given source supplying data to an input element can be processed iteratively by a single program, it is desirable to sort information from each element into blocks, each containing reports from only a single source. This sorting can be done in the course of transferring data from an input drum field to the Central Computer System by reading data on a single field by status-identification. The site identification code, the message label, or both, in each report, is used as the basis of sorting.

5.2.2 Reading by Status Identification

In reading a drum field by status-identification, certain bits within each drum word must compare exactly with code bits specified within the *SDR* instruction which selects the drum field for reading so that the word will be accepted by the Central Computer System. If the identification bits represent the source or type of input message, only the messages from one source or of one type are accepted during a single IO operation. Normally, several IO operations are required to read a given field by status-identification, one IO operation for each identification code combination used. A drum field is selected and the identification code is specified by an *SDR* (u) c instruction. The storage address for the first message in the block being read is specified by an *LDC* x instruction. The reading operation is then initiated by an *RDS* n instruction with n requesting the transfer of more words than are expected. After one complete revolution of the drum field (approximately 20 milliseconds), all the words or messages containing the specified identification code have been transferred to the Central Computer System and the IO operation is terminated by the Drum System. The Central Computer System program then determines the number of words or messages accepted by comparing the contents of the IO word counter (using a *CSW* instruction) with the *RDS* n instruction which initiated the reading operation. The number so determined is later used to control the number of

iterations of the routine which process the messages in the block.

A new reading operation is then started, using the same field but specifying a different identification code. A new *SDR* (u) c instruction, with c containing the next identification code, is required. The remainder of the program is the same for each reading operation. When the field has been read, using all active identification codes, all the information on that field should be in the Central Computer System in blocks suitable for iterative processing. The reading of an entire field takes approximately 20 milliseconds for each identification code used. Thus, if 10 identification codes are used with one field, the complete reading of that field requires about 200 milliseconds.

5.2.3 Sorting

5.2.3.1 LRI Element Information

The LRI element places processed reports on two drum fields, namely, LRI field 1 and LRI field 2, for delivery to the Central Computer System. Each field can be read by status (in which case no sorting is done) or by either of two status-identification combinations. An *SDR* (35)₈ c instruction specifies the reading of LRI field 1 by status-identification, using bits R12 through R15 of the first word of each LRI message for comparison. Bits R12 through R15 of the *SDR* instruction contain the combination with which the corresponding bits of the LRI message must compare before the message will be accepted by the Central Computer System. An *SDR* (37)₈ c instruction is the same for LRI field 2. Since bits R12 through R15 of the first drum word of the LRI message contain the site identification code for the message, use of these instructions allows the formation of blocks of LRI reports in the Central Computer System, each block containing reports from a single P site. (Refer to table 2-22.)

TABLE 2-22. LRI SITE IDENTIFICATION CODES

CHANNEL	SITE IDENTIFICATION CODE				
	R12	R13	R14	R15	OCTONARY
Information not available					

The LRI fields can also be read, using bits R7 through R15 in the first drum word of each message as an identification code. These bit positions contain both the message label and the site identification bits of each message, allowing the sorting of LRI information into separate blocks by message type from each site. Thus, the height-replies from a site (which require different processing than search radar or IFF reports) can be assembled into a block separate from the block of search

radar reports, with a third block for IFF reports. LRI field 1 is selected for reading, using bits R7 through R15, by an *SDR* (50)₈ *c* instruction; LRI field 2 is selected by an *SDR* (51)₈ instruction. Table 2-23 contains the message label codes for LRI.

TABLE 2-23. LRI MESSAGE LABEL CODES

MESSAGE TYPE	MESSAGE LABEL				
	R7	R8	R9	R10	OCTONARY
Information not available					

5.2.3.2 GFI Element Information

GFI element messages can be sorted into blocks, each of which contains messages from only one radar, by use of an *SDR* (33)₈ *c* instruction, with *c* specifying the contents of bits R11 through R15 which allow acceptance of GFI message words. Bits R11 through R15 of a GFI message word contain the site identification codes for that word. Table 2-24 lists the GFI site identification codes.

TABLE 2-24. GFI SITE IDENTIFICATION CODES

CHANNEL	SITE IDENTIFICATION CODE					
	R11	R12	R13	R14	R15	OCTONARY
Information not available						

5.2.3.3 XTL Element Information

XTL element messages can be sorted into blocks, each block containing messages from only one source, by the use of an *SDR* (25)₈ *c* instruction, with *c* specifying the contents of bits R11 through R15 of the first drum word which allow acceptance of XTL message words. Bits R11 through R15 in the first drum word of an XTL message contain the site identification code for that message. Table 2-25 lists the XTL site identification codes.

TABLE 2-25. XTL SITE IDENTIFICATION CODES

CHANNEL	SITE IDENTIFICATION CODES					
	R11	R12	R13	R14	R15	OCTONARY
Information not available						

5.2.3.4 MDI Drum Entry Section Information

Information on the manual input field can be sorted into blocks, each block containing information of only one type, by using an *SDR* (23)₈ *c* instruction, with *c*

specifying the contents of bits R14 and R15 of each MDI word which will be accepted in the reading operation. Bits R14 and R15 of an MDI word contain the message label for that word. The codes for MDI message labels are given in table 2-26.

TABLE 2-26. MDI MESSAGE LABEL CODES

MESSAGE TYPE	MESSAGE LABEL		
	R14	R15	OCTONARY
Computer entry punch	0	0	4, 0
Track display	0	1	5, 1
Radar data:			
Uncorrelated	1	0	6, 2
Correlated	1	1	7, 3

The separate codes for each type of message allow the Central Computer System to obtain card-entered information apart from display data and also to sort display data into three separate blocks. Since different display consoles may take light gun action only on specific types of display data, the programmed operation of finding the one item originated by a specific console can, in most cases, be simplified if the search is initially limited to one of three blocks of reports instead of including the reports of all three in one larger block of items.

5.2.4 Reading by Status

All fields which can be read by status-identification can also be read by status. Although reading by status does not sort the data into separate blocks by source or message type, it requires only one IO operation of approximately 20 milliseconds to read the contents of an entire field. Unless the input field reading routine has a provision for skipping the reading by an identity code corresponding to a channel which is not feeding information to a field, reading a field by status-identification when that field is receiving data from only two out of a possible ten sources would be time-consuming. If the reading routine recognizes the number of sources supplying data to a field, the routine may call for the reading of a field by status rather than by status-identification, provided the number of sources is low enough. In this case, the sorting operation would be performed within the Central Computer System.

5.3 INFORMATION PROCESSING

5.3.1 Radar Data Inputs

5.3.1.1 General

The radar data input function processes the incoming radar data, as obtained from the long-range radar

input element or the gap-filler input element, to facilitate automatic tracking. The functions to be performed include mapping, radar orientation, masking, quality analysis, radar set status control, and co-ordinate conversion and transformation. With the exception of the latter two functions, the operations require man and machine co-ordination.

Mapping requires the elimination of data in areas in which the density of returns is excessive for the automatic tracking function. Radar orientation involves the monitoring of the orientation of the long-range radars by processing the returns from fixed beacons as a standard of comparison. This provides a check on the azimuth and range reports from the radar set. Masking is the elimination of excessive overlapping radar coverage and of data beyond the subsector boundaries. Quality analysis is the determination of the suitability of the radar data for automatic tracking, while the radar set status control determines the status of each radar site on the basis of the limitations of the operating system. The incoming data, in terms of digital information in relationship to a north reference, must be referenced with respect to a common set of co-ordinates for the subsector. This is the co-ordinate conversion and transformation function. The co-ordinate conversion and transformation function is wholly program-controlled; the other functions are also programmed, but with human intervention possible through the use of keyboard intervention switches; situation, digital, and radar data mapping displays; warning lights; and audible alarms.

Program control of input data is best understood by observing the flow of radar information from a site as the various program functions are performed. (See fig. 2-4.)

5.3.1.2 Mapping

Mapping is intended to eliminate ground, sea, and weather clutter, as well as some types of jamming, when the returns from the sources are so dense as to prevent automatic tracking. Mapping is a manual function performed by painting out the unwanted clutter from the face of a plan position indicator (PPI) scope. LRI mapping is performed at the P site and GFI mapping is performed at the central, although only search radar data is mapped. Mapping is completed only after a time delay; i.e., from the time the clutter pattern changes until the mapping fluid can be painted over the clutter display.

The boundaries of the mapped area can be supplied from a P site to the central upon request from the latter. The radar data for one scan are lost during the time the boundaries are supplied. The mapped area boundaries are displayed on the LRI monitor consoles at the central. The range and azimuth co-ordinates defining these boundaries are also stored on the input

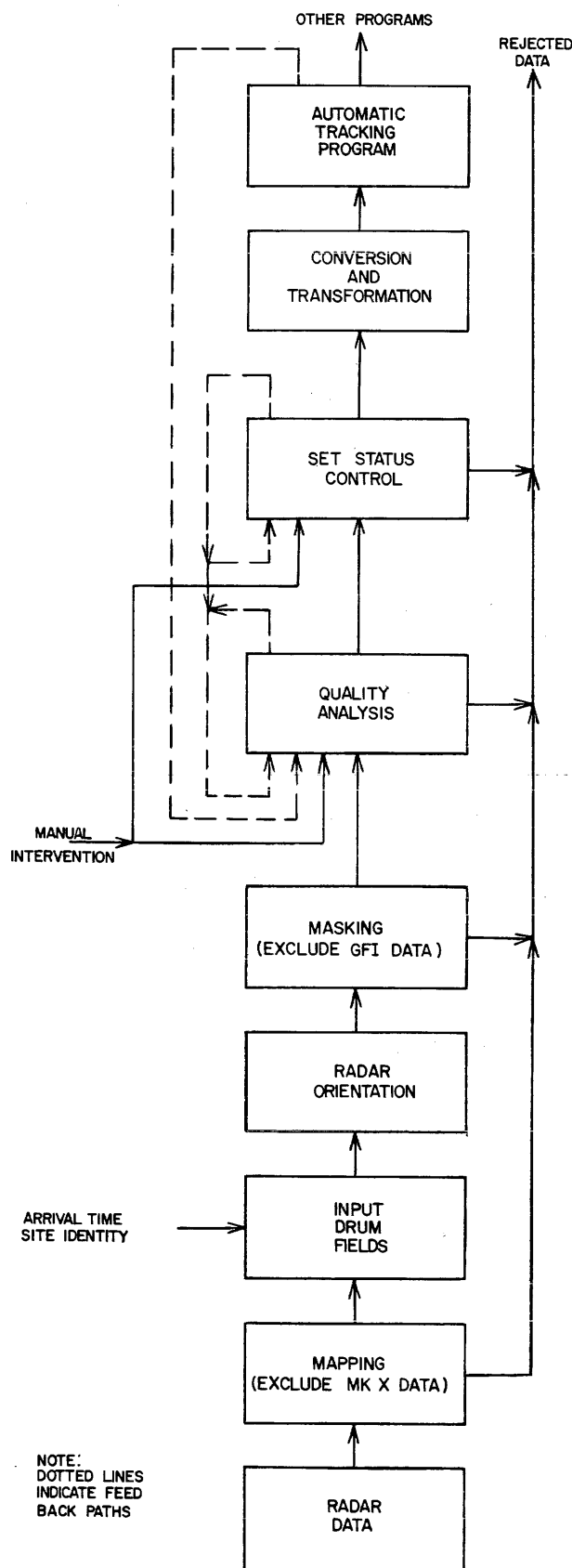


Figure 2-4. Radar Data Flow Through Input Programs

drum fields with a special message label. The co-ordinates are periodically cleared from the drum but are not subject to further processing by the program. The boundary storage applies only to LRI – not to GFI.

5.3.1.3 Radar Orientation

The orientation of LRI radars is monitored through a system of fixed beacons (at least one but not more than three per radar site) whose positions, relative to the sites, are known to an accuracy of 3 minutes of azimuth angle and approximately 100 feet in range. The fixed beacon returns are normally received at each radar scan and are indistinguishable from other LRI returns. For every radar set, the range-azimuth quantum of the corresponding beacon is stored by the program. A BEACON switch on the mapper supervisor console is used to initiate action. When this action is taken and the LRI set to which it is to apply is specified, the program checks all radar returns from that set for proximity to the known beacon location; the check continues for approximately 10 radar scans. During each of the 10 scans, the range and azimuth deviations of the radar return closest to the known location, as well as the count of the number of returns found within the console. This check is performed at least once daily. The orientation function is performed prior to the masking function, as shown in figure 2–4. This avoids the necessity of restricting the beacon location to areas not normally masked.

The range of the radar beacon, received by the central, exceeds its true range, a condition caused by a delay incorporated in the beacon circuits. The beacon returns are mapped out on the initiation area discriminator.

5.3.1.4 Masking

The major part of masking is performed by the program to eliminate transmission of unwanted data and thereby to increase the effective telephone line capacity for live data. The program stores a set of masking parameters that limit the radar coverage to triple coverage over all areas at an altitude of 10,000 feet. The parameters are determined by the coverage patterns of the radars. Masking is changed when a radar set in a sector is inoperative or is supplying poor data. Some or

all of the lost coverage can be restored by selecting the program mask in accordance with one of the following possibilities:

- Normal masking is performed in accordance with the boundaries that have previously been stored.
- Eliminate all masking within the affected subsector to restore coverage in areas where data from one or more radar sets is lost.
- Eliminate all masking outside the subsector. This is used to assist an adjacent central in case that central loses normal coverage.

5.3.1.5 Quality Analysis

Indications of data quality and set status are obtained from both manual and automatic sources, as shown in figure 2–5. These indications are kept current and stored for use as a guide to the personnel and as a program control. Data quality is determined from data counts and from visual evaluations by the mapping supervisors. The number of items of data from each LRI radar is examined in each program subframe both before and after masking. The number of data after masking is combined with the corresponding numbers of the two preceding subframes to form an overall data count. The program provides a data count analysis to determine if the data from each set is normal, excessive, and/or at a rate that differs from the number of items of data set as the predetermined limit. The limit for each radar set is not necessarily the same. Rejection of data takes place if the number of data before or after masking exceeds a fixed limit, with a lower postmasking limit. Rejection is also required if, in three program subframes (approximately one radar scan period), the number of data accepted exceeds a preset threshold. All the above indications are of excessive counts. The quality of data is also determined by operating personnel observing a monitoring console.

5.3.1.6 Radar Set Status Control

The set status control program interprets the manual and automatic indications of data quality in accordance with four possible control actions. One is the on action, which is normal for a set. When in effect, the program determines the set status. Excessive data count

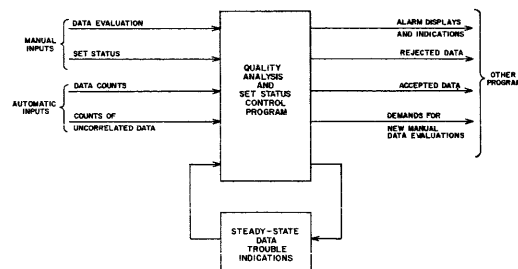


Figure 2–5. Quality Analysis and Set Status Control Flow Diagram

during a subframe calls for rejection. If the most recent insertion of data evaluation for a radar set is poor or if an excessive or changing rate of data exists, the program will provide a programmed off indication.

The second action is the override; this applies stricter limits on the actions of the set status control program. Data are rejected to protect storage capacity for only one subframe at a time where the excessive data count limits are more stringent. Data rejection is complete during the off and unavailable functions in the set status control program. The off function occurs when clutter, or other data trouble, in the radar set is so drastic that automatic tracking is impossible. In cases of system overload, this action may prevent data from other sets from being rejected. The unavailable function is used to indicate that a set is off because of equipment failure or maintenance operations.

5.3.1.7 Co-ordinate Conversion and Transformation

The program converts all range and azimuth data from radar polar to rectangular co-ordinates in a plane parallel to the plane that is tangent to the earth at that radar site. A slant-range correction, based on a nonzero elevation angle, is made for all returns. The data are then transformed to rectangular co-ordinates in a master plane for the subsector. The origin of this master plane is the origin of the subsector co-ordinates. This origin is so chosen as to minimize the distance between it and the LRI site from which it is farthest removed. All normal track and radar data displays, except those on the mapping and input monitor consoles, are relative to this subsector reference point. The stereographic projection is used both for conversion and for transformation.

5.3.2 Correlation

5.3.2.1 Principles of Tracking

Tracking consists of operating on digitalized radar data to detect the presence of new aircraft tracks and to provide velocity and position estimates on all previously established tracks. There are two major processes involved in tracking established tracks. It is necessary to examine all the radar data from all radar sets and to associate each datum with a particular track. This process, termed correlation, is explained in the following text. The second phase of tracking established tracks is the analysis of data correlated with each track, in order to form a current estimate of position and velocity, and to predict the future position for correlation of radar data during the following program frame.

5.3.2.2 Search Areas

In order to correlate radar returns with aircraft tracks, it is first necessary to establish a search area around each track; this defines the area of data accept-

ance for that track. When a radar return lies within the search area of any track, the return is said to be correlated with that track. The search area is subdivided into smaller areas, establishing two areas known as the large and small search areas. The use of two search areas provides additional information concerning the distribution of returns about a track without making it necessary to store the co-ordinates of each return correlated with the track. The small search area normally contains all radar returns belonging to the aircraft. The large search area, approximately three times the size of the small area, is sufficiently large to correlate returns when the track maneuvers or turns.

5.3.2.3 Multiple Radar Coverage

The correlation problem is complicated because several radars may see the same track, and the quantization of these radars is not necessarily the same. For this reason, many sets of large and small areas are used; each set has dimensions corresponding to a particular range of quantization. When returns having varying quantization are received in the small search area of any track, only the best (with regard to quantization) of these returns (the returns associated with the smallest value for the radius of the small search area) are used. These best returns are then averaged, so that only the single resultant position need be stored for use during the smoothing process (estimation of position and velocity). In the large search area, only the return closest to the predicted track position is accepted. However, the total number of returns falling within the large search area of any track is recorded. This helps identify the presence of noise and of clutter returns. In addition, each return is compared with the current best return. If the two returns are separated by more than the radius of the small search area, a bad distribution is indicated. Returns closer than this are probably multiple returns from a single target. Bad distribution indicates two or more unrelated returns; i.e., two tracks, a track and clutter, etc.

5.3.2.4 Track Sorting

It is not necessary to check each return for correlation with each track. The number of correlation attempts is reduced by pre-sorting the tracks on a geographical basis. Thus, a return need be checked against only those tracks in the same general geographical area.

The problem is to sort tracks in such a way that all the necessary comparisons of radar returns and tracks can be made as quickly as possible, particularly in conditions of maximum allowable load to the system. The basic logic of sorting and comparison programs assumes that the following logical steps might be required in a correlation program. Only those functions of the correlation program affecting or affected by the sorting scheme are listed.

First, all tracks are to be sorted at least once per frame, according to predicted geographical location. This is done not only with tentative and established tracks, but also with radar returns from the previous scan, for which automatic initiation is attempted.

An overlap check routine cycles through the sorted tracks and marks those whose search areas overlap. Tentative or pickup-phase tracks with overlapping search areas are treated as multiple indications of the same track. All except the first of such a group of overlapping tentatives are marked for elimination from comparison with incoming radar data. Overlapping established tracks are assumed to be crossing and may be assigned to track monitors.

Comparison of radar data with track data is the most time-consuming of the processes involved. Each radar return is compared with the tracks that could possibly be correlated with the return. The return is said to correlate with a track if it lies within the search area centered around that track. Comparison is not made with tentative tracks eliminated in the overlap check. For any one radar return, the search for a correlating track stops as soon as one is found, unless that track is in overlap condition, in which case the search continues through all the remaining tracks that might correlate with the return.

There are several methods of sorting and comparing. The first method is by the use of strips, which involves dividing the subsector into strips running either east to west or north to south. Each radar return is checked against tracks in its own strip and in strips on either side. By this method, it takes relatively little time to find which strip a return is in and to set up indexing ranges for the corresponding tracks.

The second method is by the use of half-strips. Each radar return is checked against tracks in the necessary half-strips in its own half of the subsector. Then, if a correlated track has not yet been found, and if the return is within r miles of the center dividing line, it is checked against tracks in the corresponding half-strips on the other side of the dividing line. For returns more than r miles from the dividing line, it is not necessary to check the strips on the other side of the line. The time required for setting up indexing ranges is only slightly longer for this method than for the strip method. It is compensated for by the time saved in not always having to compare with tracks in the whole strip. Other sorting methods can be devised which require more sorting time but reduce correlation time more than the methods just described.

5.3.2.5 Crossing Track Situations

In some cases, as a result of overlapping search areas, a single radar datum may correlate with more than one track. Thus, the return is associated with the

closest track. If one of the tracks is an initial pickup track, the datum will not be associated with it.

5.3.2.6 Smoothing and Prediction

As a result of positional errors inherent in radar data which has been correlated with a track, the data must be smoothed to obtain a reasonable estimate of the actual aircraft position. This smoothing corresponds to drawing the smoothest curve through all past and present data. The track position must then be predicted for the next correlation time so that the automatic tracking is continued in a closed cycle.

Smoothing and prediction is done in two stages. Some of the radar data correlated with a track can be immediately recognized (by its proximity to the predicated position of the track) as data which may be processed by straightline tracking procedures. Such data is processed three times per frame, following each completion of the correlation function. This takes advantage of the effective increase in the sampling rate provided by the multiple radar coverage, improving the response time of the automatic tracking function and permitting the early detection of deviations from a straight-line flight path.

The second stage of smoothing and prediction is performed once per frame. If none of the data described in the preceding paragraph has been received during the frame, all other data correlating with the track must be examined to determine if any may be used for smoothing and prediction or whether the track must be dead-reckoned along its previously computed velocity. By performing this phase of smoothing and prediction only once per frame, the tracking decision may be deferred, when necessary, until more complete information is available.

5.3.3 XTL Information Processing

A crosstold message is accepted after the address and parity have been checked. An extrapolation is performed to bring the information to a current basis. If required, the information is then placed in an automatic tracking condition.

5.3.4 MDI Information Processing

5.3.4.1 Automatic Initiation

Initiation is the activation of a new track after it has been detected. In the initiation process, it is necessary to treat all uncorrelated radar data as tracks, subject to automatic or manual track detection procedures. As far as possible, detection is performed automatically by the program; manual detection is performed when necessary.

The usefulness of automatic track detection and initiation lies in the ability to maintain watch without tiring and the capacity to relieve operators of that part of the work load represented by completely routine decisions.

The selection of areas in which automatic initiation is performed is by means of an area discriminator. The area discriminator is a situation display console on which uncorrelated radar data is displayed. Areas in which automatic initiation is not to take place are mapped out by the use of a semiopaque ink, which prevents the photoelectric tube mounted above the console from responding to reports in the mapped areas.

The area discriminator selects radar data in areas of automatic initiation for examination as potential tracks. The program observes data on subsequent scans in the region of the initial return until it can determine whether this data resulted from an aircraft or from an extraneous return. Data thus processed is called a tentative track. After it has been determined that the track represents an aircraft, it is called an established track.

5.3.4.2 Manual Initiation

Manual initiation is used in areas where automatic initiation is prohibited by use of the area discriminator. Heavy traffic areas and critical areas are restricted; only manual initiation is used in them.

5.3.4.3 Card Processing

The program provides for a check of the legality of MDI card data; then it translates the data into forms compatible with its operation. The data is then used when necessary.

5.3.4.4 Light Gun and Keyboard Messages

A light gun mounted on a situation display console is used to transfer radar data or a track display message. Depressing the trigger of the light gun, which is aimed at the target display in question, causes the transfer of the track number in a track display message or the transfer of X and Y co-ordinates in a radar data message. The console identify code is generated in the MDI element and transferred at the same time. Where a keyboard message precedes initiating action via the light gun, the program provides for the inspection of the core matrix image which is placed in core memory. The action specified by the keyboard message will be taken on the light gun message.

CHAPTER 6

DISPLAY OPERATIONS

SECTION 1

SITUATION DISPLAYS

1.1 INTRODUCTION

This Chapter presents descriptions of the significance of the messages handled by the Display System and of the routing of these messages. This Section describes situation display message contents and routing; Section 2 describes digital display messages. (Refer to Part 1, Ch 6, for a discussion of the Display System.)

1.2 MESSAGE TYPES

1.2.1 Major Classifications

The messages classed as situation display messages are of two major types, radar data messages and track display messages. Radar data messages are each contained in a single drum word; track display messages each occupy eight drum words.

1.2.2 Radar Data Messages

Radar data messages are sometimes called point messages since each such message designates the point locating a target. The symbol displayed at the point identifies the radar source (search radar or IFF) and whether the target report is correlated or not; i.e., whether a track data message can be developed from the radar report or not. The age of the message is distinguished by the brightness of its display: if bright, the message is a present report; if dim, the message is a past report. A series of successive radar data messages on a single target is displayed as shown in figure 1-145. The direction in which the target is moving is indicated by the position of the present (bright) report on the target. Radar data messages are useful primarily in manual initiation of tracks and in analyzing trouble track situations. They are used most in conjunction with the radar inputs program operations and with the smoothing and trouble detection program operations. The drum layout of a radar data message is given in figure 1-146.

1.2.3 Track Display Messages

1.2.3.1 General

Track display messages are capable of presenting a great deal of information on the targets they describe. Two subtypes of track display messages are distinguished,

tabular messages and vector messages. Each subtype occupies eight drum words. A tabular message, however, can cause display of one vector and up to 13 characters (although 18 characters are contained within the message), whereas a vector message can cause display of four vectors and four characters. The designations of the character positions in a tabular message are given in figure 1-148. The equivalent information for vector messages is given in figure 1-152.

1.2.3.2 Tabular Messages

A tabular message may be either a tabular track message describing a target in track or a tabular information message describing anything which can be represented by up to 13 characters. (See fig. 1-149.) The drum layout of a tabular message is given in figure 1-150.

1.2.3.3 Vector Messages

A vector message can cause display of up to four independently positioned vectors and four characters associated with the fourth vector. Vector messages are used for a variety of information displays including flight histories, geographical boundaries, raid symbols, and attention devices. The drum layout of a vector message is given in figure 1-154.

1.3 INFORMATION CONTENTS

1.3.1 Radar Data Messages

A single radar data message (a single point message) describes the location of a target at some instant of time. Two successive reports on a single target are separated in time by one program frame (approximately 15 seconds). Therefore, a series of eight reports on a single target (the maximum number of radar data messages possible for a single target) displays the path of the target over the past two minutes. If the radar data symbol indicates that the reports are correlated, a tabular track message has been established for that target.

1.3.2 Vector Messages

A vector message representing the flight history of a track can be made up. Each vector within the mes-

sage represents the average path of the track over a two-minute period; i.e., summarizes the information presented by eight radar data messages. The entire vector track history message can display the target path over the past eight minutes. Every two minutes (eight program frames) the vector message must be revised, forming a new vector from eight radar data messages, placing this vector either first or last (whichever is selected to represent the latest vector), moving three of the old vectors down one position in the drum layout, and discarding the oldest vector.

In a track history message for a hostile, the G characters have the following meaning:

- a. G₁ — Subsector having tracking responsibility
- b. G₂ — Number of hostile tracks: 1, 2, or blank for more than 2 tracks
- c. G₃ and G₄ — Number of hostiles destroyed; blank 0 in these character positions indicating that no hostiles have been destroyed; blank U indicating that the number of hostiles destroyed is unknown

The rest of the symbology to be used with the G characters of vector messages, as well as the different attention symbols for which the vectors may be utilized, are not available at this time.

1.3.3 Tabular Track Messages

1.3.3.1 Point Feature

The point feature is unique in that it conveys no information to a console operator. It is used solely in conjunction with light guns or area discriminators and appears on a display console only when a light gun (or area discriminator) at that console is operated.

1.3.3.2 Vector Feature

The vector feature of a tabular track display message indicates the speed and heading of the target described by that message. The vector origin coincides with the position of the point feature. The length of the vector is proportional to the average velocity of the target; a vector 1/4-inch long represents a speed of 200 knots. The maximum length of a vector is approximately 2 inches.

1.3.3.3 E Feature

The E feature contains classification information on the track described by a tabular track display message. Three track classes are distinguished: tentative tracks, interceptor tracks, and all others. Each track class is indicated by a different symbol or one of a group of symbols different for each class. Table 2-27 lists the possible symbols which can appear as the E feature for each class and the significance of each symbol.

TABLE 2-27. E FEATURE SYMBOLOGY

TRACK CLASS	SYMBOL	SIGNIFICANCE
Tentative	T	Track is tentative
Interceptor	N	Not on command tracking
	C	On command tracking
All other tracks	—	Flight size, one target
	=	Flight size, several targets
	≡	Flight size, many targets
	U	Unknown flight size

Further information on each class of track is given in the other features of a message.

1.3.3.4 C Feature

The C feature occupies up to four character positions designated C₁ through C₄. The C feature contains the track number which uniquely identifies that track within its class. The C feature symbology is given in table 2-28.

TABLE 2-28. C FEATURE SYMBOLOGY

TRACK CLASS	SYMBOLS				SIGNIFICANCE
	C ₁	C ₂	C ₃	C ₄	
Tentative	0	0	1		Tentative track number; C ₁ is blank
	through				
	4	0	0		
Interceptor		0	1		C ₁ and C ₂ contain two-letter squadron designation; C ₃ and C ₄ contain two-digit interceptor number
	through				
		2	5		
All other tracks	0	0	1		C ₁ is subsector designation letter (identifies originating subsector);
	through				
	6	9	9		C ₂ through C ₄ contains three-digit track number

1.3.3.5 D Feature

The D feature, consisting of character positions D₁ and D₂, provides further information on a track. Character position D₁ contains an identification symbol. Table 2-29 gives the symbols which may appear in the D₁ position and their significance.

TABLE 2-29. D₁ CHARACTER POSITION SYMBOLOGY

SYMBOL	SIGNIFICANCE	SYMBOL	SIGNIFICANCE
Blank	Tentative	H	Hostile
P	Pending	U	Unknown
F	Friendly	K	Faker
S	Special	I	Interceptor
R	Round robin		

TABLE 2-30. D₂ CHARACTER POSITION SYMBOLOGY

TRACK CLASS	SYMBOL	SIGNIFICANCE	TRACK CLASS	SYMBOL	SIGNIFICANCE
Tentative	G	Initiation score good	All tracks, status	L	Lost (no data on track)
	F	Initiation score fair		D	Drop cycle (about to be dropped from system)
	P	Initiation score poor		E	Extrapolated (not based on radar data)
Established	G	Tracking merit good	Crosstold in		Tracking merit or tracking status digit from originating subsector
	F	Tracking merit fair			
	P	Tracking merit poor; less than 1 minute			
	1	Tracking merit poor; between 1 and 2 minutes	Manual tracks, source	5, 4*	Manual center
	2	Tracking merit poor; between 2 and 3 minutes		6, 4*	Group Observer Corps (GOC) filter center
	3	Tracking merit poor; between 3 and 4 minutes		4, 3*	Antiaircraft Operations Center (AAOC)
Interceptor	S	Scrambled (ordered aloft)		3, 4*	Airborne Early Warning (AEW)
	A	Airborne		3, 2*	Picket ship

* Symbols identified by character selection co-ordinates; see figure 1-147.

TABLE 2-31. B₄ AND B₅ CHARACTER POSITION SYMBOLOGY

TRACK CLASS	SYMBOLS B ₄ B ₅	SIGNIFICANCE
Any	Blank	Tentative, drop cycle, or manual input
Established or lost	U	Unassigned for monitoring
	T 1	Assigned for monitoring to track supervisor 1 or 2
	or T 2	
	S 1	Assigned for monitoring to special track monitor 1 or 2
	or S 2	
	1 1 through 2 6	Assigned for monitoring to track monitors 1-1 through 2-6, respectively
	0 1	Assigned for monitoring to overlap technician 1, 2, or 3
	0 2	
	0 3	
	I	Interceptor, monitored by assigned intercept director
Scrambled or airborne interceptors	I	Assigned for initiation by assigned intercept director
	1 1 through 1 5	Assigned for initiation by track initiators 1 through 5, respectively
	I S	Assigned to initiation supervisor for initiation
Correlated crosstold in		B ₄ shows tracking merit in receiving subsector; B ₅ contains originating subsector identification
Noncorrelated crosstold in		B ₄ is blank; B ₅ contains designation of originating subsector

Character position D₂ indicates the tracking merit for the target described by the message. This tracking merit indication informs console operators of the degrees of validity of the information contained in the message. In some cases, the D₂ character position may indicate tracking status or source rather than tracking merit if the target is not radar originated. The symbology used in the D₂ character position is given in table 2–30.

1.3.3.6 B Feature

The B feature is used primarily by the air surveillance section, whereas the A feature is used primarily by the weapons direction section. The B feature is divided into four subgroups consisting of B₄ and B₅, B₁, B₂, and B₃, respectively. Character positions B₄ and B₅ contain air surveillance assignments or crosstelling track merit information. The symbology for these positions is given in table 2–31.

Character position B₁ identifies the type of radar data correlating with the track. If search radar data is used in maintaining the track, B₁ contains an N; if IFF data is used, the character position contains an X; all other tracks have the B₁ position blank.

Character position B indicates the age of the track measured from a starting point which differs for each class of track. The age is indicated in minutes by one of the digits from 0 through 9. The point of measurement for each track class is given in table 2–32.

TABLE 2–32. TRACK AGE MEASUREMENT

TRACK CLASS	POINT OF MEASUREMENT
Tentative	From time of detection
Established, lost, or extrapolated	From time of establishment
Airborne	From time of becoming airborne
Correlated crosstold in	From time of entrance into overlap zone
Non-correlated crosstold in	From time of initial report
Manual data input	From last report
Scramble and drop cycle	None (B ₂ is blank)

Character position B₃, the crosstell status digit, indicates for any track the reason for its being crosstold. For established, lost, extrapolated, scrambled, airborne, or manual data input tracks, B₃ indicates outgoing crosstell status. For crosstold-in tracks, whether correlated or not, B₃ shows incoming crosstell status; i.e., the purpose for which the track was crosstold. The symbols which may appear in B₃ and their significance include the following:

- I – crosstold for interception
- T – tracking
- A – antiaircraft
- W – warning
- R – recrosstold for warning
- H – handover of tracking
- Blank – track is not crosstold

1.3.3.7 A Feature

The A feature indicates, to weapons direction personnel, the status of various tracks for weapons direction purposes. The character positions A₁ through A₆ are called combat status digits. Their contents are different for different track classes. For tentative tracks, no A feature appears; all A character positions are blank. The contents of the A feature for other types of tracks are discussed separately.

For interceptor tracks, character positions A₁ through A₄ show the type of mission to which the interceptor is assigned. Character positions A₅ and A₆ identify the weapons director and intercept director to which the interceptor is assigned with a digit in each position. The symbology used in positions A₁ through A₄ for interceptor tracks is described in table 2–33. Characters A₁ and A₂ show weapons assignment status, A₃ shows crosstell status, A₄ shows antiaircraft status (AA), and A₅ and A₆ show weapons director (WD) to which the target tracks are assigned and their altitudes. For nontarget tracks, characters A₅ and A₆ show air movement data (AMD) status and the time required to terminate the flight plan. A target track to be designated by the characters of the A feature is any hostile, unknown, faker, or pending track, or a track of any

TABLE 2–33. INTERCEPTOR MISSION SYMBOLOGY

MISSION	CHARACTER POSITION	SYMBOL	SIGNIFICANCE
Interception	A ₁ through A ₄		Track number of target being intercepted
Noninterception	A ₁	C	Combat air patrol (CAP)
		R	Return to base (RTB)
		D	Deployment
		Blank	Combat air patrol (CAP)
	A ₂ through A ₄	3 letters	Base designation for RTB or deployment

other noninterceptor identity against which a weapons direction has committed one or more interceptors.

The symbology for the A feature tracks is as follows:

- a. In character A_1 and A_2 (weapons assignment status)
 1. FF — Assigned for flight following
 2. UU — Unassigned
 3. AD — Assigned, action deferred
 4. AU — Assigned, no weapons assigned
 5. 00 to 99 — Assigned for interception; number represents the number of interceptor aircraft committed. If this number is 100 or more, the character MM will appear instead of numbers.
 6. Blank — Not assigned for weapons direction (not a target track)
- b. In character A_3 (crosstell status)
 1. I — Interception purposes
 2. T — Tracking purposes
 3. A — AA purposes
 4. W — Warning only
 5. R — Recrosstold (for warning)
 6. H — Tracking responsibility has been handed over
 7. Blank — Not crosstold
- c. In character A_4 (AA status)
 1. Blank — Not assigned to AA
 2. 1, 2, J — No AA action; directed to AAD 1 or 2, or 1 and 2
 3. P — Pilots being passed to AA
 4. H — Hold fire (HU only; simulated if faker)
 5. E — Engaged (HU only; simulated if faker)
 6. S — Splashed (H only; simulated if faker)
- d. In character A_5 (WD assignment or AMD status)
 1. 1 to 4 — Target tracks, WD to whom assigned
 2. Blank — No AMD (unassigned)
 3. C — For nontarget tracks, on correlated flight plan
 4. 4, 6* — For nontarget tracks, off correlated flight plan
 5. 7, 7* — Pending in a raid
 6. 6, 6* — Suspicious friendly
 7. F — For nontarget tracks, free-space friendly

**These numbers are the character selection coordinates for the symbols whose significance is given above. (See fig. 1-147.)*

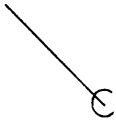
- e. In character A_6 (altitude or time to termination of flight plan)

1. L — For target tracks, low (0 to 9,999 feet)
2. M — For target tracks, medium (10,000 to 29,999 feet)
3. H — For target tracks, high (30,000 feet or higher)
4. U — For target tracks, altitude unknown
5. Blank — For nontarget tracks, no request
6. V — For nontarget tracks, requested, negative reply.

1.3.3.8 Examples

A typical message to be used by the air surveillance section is shown in figure 2-6, part A. This message indicates that an interceptor (I in the D_1 character position) 21 of the JK squadron (JK21 in character position C_1 to C_4) is on track. This track is correlating with normal search radar data (N in the B_1 character position) and between 6 to 7 minutes have elapsed since the track became airborne (6 in the B_2 character position). This interceptor is on command tracking (C in the E character position), its tracking quality is good (G in the D_2 character position), and it is under direction of the air surveillance section track initiator 4 (I4 in character positions B_4 and B_5). The interceptor is heading at an azimuth of approximately 315 degrees at the speed represented by the length of the vector.

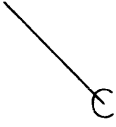
The same message that was viewed by the air surveillance section may be viewed by the weapons direction section with the A feature substituting for the B feature. (See fig. 2-6, part B.) This message indicates that the



```

      N 6
    C I G I 4
      J K 2 1
  
```

AIR SURVEILLANCE
A.



```

      V 2 6 3
    C I G 3 5
      J K 2 1
  
```

WEAPONS DIRECTION
B.

Figure 2-6. Tabular Track Data Message Displays

TABLE 2-34. RADAR DATA CATEGORIES

CATEGORY NUMBER	DISPLAY ABBREVIATION	DISPLAY
Basic Radar Data Categories		
0	RD _{pu}	Present RD uncorrelated.
1	RD _{pc}	Present RD correlated.
2	RX _{pu}	Present IFF uncorrelated.
3	RX _{pc}	Present IFF correlated.
4	RD _{hu}	History RD uncorrelated.
5	RD _{hc}	History RD correlated.
6	RX _{hu}	History IFF uncorrelated.
7	RX _{hc}	History IFF correlated.
Mixed Radar Data Categories		
1	RP _u	All present uncorrelated (RD and IFF).
2	RH _u	All history uncorrelated (RD and IFF).
3	RP _c	All present correlated (RD and IFF).
4	RH _c	All history correlated (RD and IFF).
5	RD _p	All present RD (correlated and uncorrelated).
6	RD _h	All history RD (correlated and uncorrelated).
7	RX _p	All present IFF (correlated and uncorrelated).
8	RX _h	All history IFF (correlated and uncorrelated).
9	RP	All present.
10	RH	All history.

interceptor is being directed against track V263 (V263 in character positions A₁ to A₄). This interceptor is on command tracking under control of weapons director team 3 (3 in the A₁ character position), intercept director 5 (5 in the A₆ character position). All other information is the same for both displays of the message.

The tabular track messages discussed thus far were of special nature, since they displayed the E character and vector to the left or the right of the other characters. The tabular track message may also be displayed with the E character and vector above or below the other characters. This arrangement simplifies recognition of the general type of information the message contains. Thus, if the E character and the vector are to the left or to the right of the other characters, the message represents a track display. If the E character and vector are above or below the other characters, the message represents a flight plan. The tabular message

representing a flight plan consists only of the A, B, and E feature characters and the vector.

1.3.4 Tabular Information Messages

Information not available.

1.4 MESSAGE ROUTING

1.4.1 General

Situation display messages are routed from the situation display generator element to situation display consoles in the operating stations of AN/FSQ-7 Combat Direction Central according to the tactical functions of the stations. To permit this functional routing, situation display messages are sorted into tactical classes according to the type of information contained in the messages. The appropriate displays are then presented at particular consoles by the selection of specific classes of messages. The classes of situation display messages fall

into two major divisions, called categories and DAB's (display assignment bits), as explained below.

1.4.2 Categories

Each radar data message belongs to one of eight categories; each tabular and vector message belongs to one of 31 categories. Two control bits (L14 and L15) in each radar data message, together with the display bright or display dim signal accompanying the message from the radar data drum, provide the category encoding information (number) for the message. The category encoding information (number) for each tabular

and vector message is contained in bits LS through L4 of word 2 of the message.

The display of specific categories at a situation display console is controlled by fifteen category switches and by wiring at each console. The categories are controlled individually or in groups. The mixing of individual categories to form the groups called mixed categories occurs at the situation display generator element.

Table 2-34 lists the categories of radar data messages. Table 2-35 lists the categories of tabular and vector messages.

TABLE 2-35. TABULAR AND VECTOR MESSAGE CATEGORIES

CATEGORY NUMBER	DISPLAY ABBREVIATION	DISPLAY
Special Categories		
0		Null.
1		Special test pattern.
2	A	All (connected to all consoles).
Geography Categories		
3	GS ₁	Subsector boundaries, coastline, cities, landmarks — x1 expansion.
4	GS _a	Vector messages added to GS ₁ for x2 expansion.
5	GS _b	Vector messages added to GS ₁ and GS ₂ for x4 expansion.
6	GS ₄	Special areas — x2 and x4 expansions.
7	GRS	Radar sites, status (long-range input, gap-filler, and height-finder).
8	GR	Radio sites, status (interceptor control).
9	GB ₀	Operational interceptor bases, weather, status.
10	GB _s	Supplementary recovery bases, weather.
11	GA ₁	Antiaircraft batteries — x8 expansion.
12	GA ₂	Antiaircraft defense — x1 expansion.
13	GA _a	Vector messages added to GA ₂ for x2 and x4 expansions.
14	GI ₁	Air defense identification zone, open and/or defense areas — x1, x2, and x4 expansions.
15	GI ₂	Multiple corridor identification system corridors and reporting points — x1, x2, x4, and x8 expansions.
31	GG	Georef grid; real time.
Track Categories		
16	HN	Hostile, unknown, faker, pending, tracks not in raids (includes crosstold in).
17	HR	Hostile, unknown, faker, pending, tracks in raids (includes crosstold in).
18	RA	Raids of hostiles, unknowns, fakers, pendings, tracks not in raids (includes crosstold in).

TABLE 2-35. TABULAR AND VECTOR MESSAGE CATEGORIES (cont'd)

CATEGORY NUMBER	DISPLAY ABBREVIATION	DISPLAY
19	INN	Interceptors (tracks) not in groups on return to base, combat air patrol, deploy (includes crosstold in).
20	INM	Interceptors (tracks) not in groups on interception mission (includes crosstold in).
21	IGN	Interceptors (tracks) in groups on return to base, combat air patrol, deploy (includes crosstold in).
22	IGM	Interceptors (tracks) in groups on interception mission (includes crosstold in).
23	G	Groups of interceptors (includes crosstold in).
24	S	Special tracks — keystone, etc. (includes crosstold in).
25	F	Friendly round robin tracks (includes crosstold in).
26	T	Tentative tracks.
29	AMD _u	Air movement data, uncorrelated.
30	AMD _c	Air movement data, correlated.
Mixed Categories		
	GS ₂	Subsector boundaries, x2 expansion (GS ₁ plus GS _a).
	GS ₃	Subsector boundaries, x4 expansion (GS ₂ plus GS _b).
	GA ₃	Point defense coverage zones, x2 and x4 expansions (GA ₂ plus GA _a).
	GB	All airbases (GB ₀ plus GB _s).
	RHN	Raids plus hostile, unknown, faker, and pending tracks not in raids (RA plus HN).
	HA	All hostile, unknown, faker, and pending tracks (HN plus HR).
	IG	Interceptors (tracks) in groups (IGN plus IGM).
	IN	Interceptors (tracks) on return to base, combat air patrol, deploy (INN plus IGN).
	IM	Interceptors (tracks) on interception mission (INM plus IGM).
	GIN	Groups plus interceptors (tracks not in groups) (G plus INN plus INM).
	IA	All interceptors tracks (IN plus IM).
	RG	Raids plus groups (RA plus G).
	SF	Special plus friendly and round robin tracks (S plus F).
	TA	All tracks (HA plus IA plus SF).

1.4.3 Display Assignment Bits

A total of 90 bits, called display assignment bits (DAB's), are reserved in words 2, 3, and 4 of each tabular and vector message in order to classify the message into 90 tactical classifications of displays. The displays are known as DAB displays, and differ from categories in that, while a message may belong to only

one category, it may belong to any number of DAB displays. Individual or mixed DAB displays are controlled at a console by the category switches or by forced display wiring. Mixing of the DAB displays is accomplished in the situation display generator element. A DAB display may be forced at a situation display console by a bypass connection around the category switches. Table 2-36 lists the DAB displays.

TABLE 2-36. DISPLAY ASSIGNMENT BITS

DAB NUMBER	BIT POSITION	DISPLAY ABBREVIATION	DISPLAY
1	Word 2, L6	XiA	All automatically crosstold-in messages.
2	L7	XiM	All manual input tracks.
3	L8	XiH	All crosstold-in hostile, unknown, faker, and pending tracks.
4	L9	XiI	All crosstold-in interceptor tracks.
5	L10	X _o	All crosstold-out messages.
6	L11	CC	All forward-told messages.
7	L12	XXT	All crosstell trouble tracks.
8	L13	TT	All tracking trouble tracks.
9	L14	Ta	All tracks above 15,000 feet.
10	L15	Tb	All tracks below 15,000 feet.
11 through 18	RS through R7		Spare.
19	R8	SD	Weapons assignment display, tactical action display, new unassigned hostile, unknown, faker, and pending tracks with alert displays, new crosstold-in interception or warning tracks.
20	R9	ATO	Tracks out of raid correlation-box limits, tracks in group being reassigned.
21	R10	SWD	Weapons assignment display, tactical action display, assigned tracks.
22	R11	AAD ₁	Antiaircraft assigned tracks, alert displays.
23	R12	AAD ₂	Alert displays on weapons site — x8 expansion.
24	R13	WD ₁	Weapons assignment display, tactical action display, alert displays for assigned tracks (no assigned tracks).
25	R14	WD ₂	
26	R15	WD ₃	
27	Word 3, LS	WD ₄	
28	L1	IND ₁ 1	Assigned tracks, alert displays, intercept points.
29	L2	IND ₁ 2	
30	L3	IND ₁ 3	
31	L4	IND ₁ 4	
32	L5	IND ₁ 5	
33	L6	IND ₂ 1	
34	L7	IND ₂ 2	
35	L8	IND ₂ 3	
36	L9	IND ₂ 4	

TABLE 2-36. DISPLAY ASSIGNMENT BITS (cont'd)

DAB NUMBER	BIT POSITION	DISPLAY ABBREVIATION	DISPLAY
37	L10	IND ₂ 5	Assigned tracks, alert displays, intercept points.
38	L11	IND ₃ 1	
39	L12	IND ₃ 2	
40	L13	IND ₃ 3	
41	L14	IND ₃ 4	
42	L15	IND ₃ 5	
43	RS	IND ₄ 1	
44	R1	IND ₄ 2	
45	R2	IND ₄ 3	
46	R3	IND ₄ 4	
47	R4	IND ₄ 5	
48	R5	ASO	Track history, etc.
49	R6	TO	Track history, alert displays.
50	R7	IS	Alert displays.
51	R8	TI ₁	Assigned tracks, alert displays.
52	R9	TI ₂	
53	R10	TI ₃	
54	R11	TI ₄	
55	Word 3, R12	TI ₅	Assigned tracks, alert displays.
56	R13	TS ₁	Track history, assigned tracks, alert displays.
57	R14	TS ₂	
58	R15	TMS ₁	Track history, assigned tracks, alert displays.
59	Word 4, LS	TMS ₂	
60	L1	TM ₁ 1 and 2	Track history, assigned tracks, alert displays.
61	L2	TM ₁ 3 and 4	
62	L3	TM ₁ 5 and 6	
63	L4	TM ₂ 1 and 2	
64	L5	TM ₂ 3 and 4	
65	L6	TM ₂ 5 and 6	
66	L7	OT ₁	Track history, crosstold-in hostile, unknown, faker tracks, alert displays.
67	L8	OT ₂	
68	L9	OT ₃	
69	L10	HS	Requested height finder coverage, assigned tracks, alert displays.

TABLE 2-36. DISPLAY ASSIGNMENT BITS (cont'd)

DAB NUMBER	BIT POSITION	DISPLAY ABBREVIATION	DISPLAY
70	L11	IDO	Alert displays, pending tracks, track history, flight plan routes.
71	L12	IDT ₁	
72	L13	IDT ₂	
73	L14	MS	Alert displays, beacon display.
74	L15	SIMS	Assigned simulated tracks, alert displays.
75	RS	SIM	
76	R1	M-R	
77	R2	C ₁	Camera Console DAB 1
78	R3	C ₂	Camera Console DAB 2
79	R4	TWD ₁	Assigned tracks (no alert displays).
80	R5	TWD ₂	
81	Word 4, R6	TWD ₃	Assigned tracks (no alert displays).
82	R7	TWD ₄	
83	R8	IDX	Identification officer's special expanded display.
84	R9	SSC-D/O	Tactical action display.
85	R10	FOI	Tactical action display.
86	R11	Spare	
87	R12	Spare	
88	R13	Spare	
89	R14	Spare	
90	R15	Spare	

1.4.4 Special Test Lines

Three special test lines provide another means of

routing situation display messages. Table 2-37 lists these lines.

TABLE 2-37. SPECIAL TEST LINES

TEST LINE NUMBER	DISPLAY ABBREVIATION	DISPLAY
1	T ₁	All categories (no RD) mixed except null and test pattern.
2	T ₂	All DAB's mixed.
3	T ₃	All mixed categories and DAB's, mixed.

SECTION 2

DIGITAL DISPLAYS

2.1 GENERAL

Digital display information is presented on the screens of 5-inch digital display cathode-ray tubes (DD CRT's) located in situation display consoles, auxiliary consoles, and at the Command Post desk. Digital displays are used primarily to furnish additional information supplementing messages displayed on the situation display cathode-ray tubes (SD CRT's) and to summarize existing situations such as weather data, assignment data, and other data of a comparatively stable nature. Unlike the situation display information which is re-displayed every 2.54 seconds, digital display information is stored in the DD CRT for any desired period and must be erased before new information can be written.

The distribution of digital display information differs from the distribution of situation display information. The Central Computer System writes the digital display data in slots of lengths varying from 8 to 32 registers on the digital display (DD) field of the MIXD drum. Each slot is assigned to a specific digital display indicator section. (In a few exceptional cases, one slot is assigned to two indicator sections.) In contrast, situation display messages are written by the Central Computer System in slots of equal length (8 registers for TD and 1 register for RD messages) on the respective drums, and contain their own routing information

in the form of DAB and category bits within the messages.

2.2 CHARACTER-FORMING MATRIX

A digital display message consists of letters, numerals, and symbols, referred to as characters. There are 63 characters and one blank space, which is used when no character is to be displayed. Each character is selectable by an x and a y character selection co-ordinate. Figure 1-161 gives the addresses of characters on the character matrix in octonary form.

2.3 MESSAGE FORMAT

A digital display message, as viewed on the DD CRT screen, consists of up to 32 rows of five characters each, grouped into two columns of 16 rows. (See fig. 1-162.) Because of the requirements of six selection bits per character, a total of 30 bits is required for each row of characters. (See fig. 1-160.) The remaining two bits, LS and RS, are used as control bits to determine the position of the row with respect to the preceding rows. Table 2-38 illustrates the function of the control bits.

2.4 MESSAGE DISTRIBUTION

Each digital display message is assigned to a specific digital display indicator section by means of a slot number. Table 2-39 shows the assignment, as well as the length, of the respective slot.

TABLE 2-38. CONTROL BIT FUNCTIONS, DIGITAL DISPLAY MESSAGES

BIT POSITION AND CONTENTS		MEANING
LS	RS	
0	1	Start of new slot (message). Display word on upper left row of next indicator section.
0	0	Display word immediately below preceding row.
1	1	Skip a row between preceding row and the row in which word is to be displayed.
1	0	Display word on upper row of opposite side of the same indicator section.

TABLE 2-39. DD SLOT ASSIGNMENTS

SLOT NUMBER	STATION NAME	CONSOLE NUMBER	SLOT LENGTH (NUMBER OF REGISTERS)
0	Not Usable		0
1	Utility control	E20 E70	32
2	Simplex monitor	E22 E72	16
3	Duplex switch	S77, E77 S78, E78	16
4	ATO	W82	16
5	ATT	W83	16
6	SD	W81	32
7	SDT	W80	32
8	SWD	W90	32
9	SWDT	W91	32
10	WD ₁	W20	24
11	WDT ₁	W21	24
12	WD ₂	W40	24
13	WDT ₂	W41	24
14	WD ₃	W60	24
15	WDT ₃	W61	24
16	WD ₄	W00	24
17	WDT ₄	W01	24
18	IND1-1 INT1-1	W23 W22	20
19	IND1-2 INT1-2	W25 W24	20
20	IND-3 INT-3	W29 W28	20
21	IND1-4 INT1-4	W31 W30	20
22	IND1-5 INT1-5	W33 W32	20
23	IND2-1 INT2-1	W44 W43	20
24	IND2-2 INT2-2	W46 W45	20
25	IND2-3 INT2-3	W48 W47	20

TABLE 2-39. DD SLOT ASSIGNMENTS (cont'd)

SLOT NUMBER	STATION NAME	CONSOLE NUMBER	SLOT LENGTH (NUMBER OF REGISTERS)
26	IND2-4	W52	20
	INT2-4	W51	
27	IND2-5	W54	20
	INT2-5	W53	
28	IND3-1	W64	20
	INT3-1	W63	
29	IND3-2	W66	20
	INT3-2	W65	
30	IND3-3	W70	20
	INT3-3	W69	
31	IND3-4	W72	20
	INT3-4	W71	
32	IND3-5	W74	20
	INT3-5	W73	
33	IND4-1	W03	20
	INT4-1	W02	
34	IND4-2	W05	20
	INT4-2	W04	
35	IND4-3	W07	20
	INT4-3	W06	
36	IND4-4	W11	20
	INT4-4	W10	
37	IND4-5	W13	20
	INT4-5	W12	
38	AAD ₁	W86	28
39	AADA ₁	W85	28
40	AAD ₂	W88	28
41	AADA ₂	W87	28
42	ASO	S72	28
43	AST	S73	28
44	TO	S75	28
45	TOT	S74	28
46	IS	S70	12
47	TI ₁	S00	8
48	TI ₂	S01	8
49	TI ₃	S02	8
50	TI ₄	S03	8

TABLE 2-39. DD SLOT ASSIGNMENTS (cont'd)

SLOT NUMBER	STATION NAME	CONSOLE NUMBER	SLOT LENGTH (NUMBER OF REGISTERS)
51	TI ₅	S04	8
52	TS ₁	S55	16
53	TM(S) ₁	S53	12
54	TS ₂	S35	16
55	TM(S) ₂	S33	12
56	TM1-1	S52	12
57	TM1-2	S50	12
58	TM1-3	S45	12
59	TM1-4	S43	12
60	TM1-5	S42	12
61	TM1-6	S40	12
62	TM2-1	S32	12
63	TM2-2	S30	12
64	TM2-3	S25	12
65	TM2-4	S23	12
66	TM2-5	S22	12
67	TM2-6	S20	12
68	OT ₁	S60	16
69	OT ₂	S63	16
70	OT ₃	S67	16
71	HS	S11	32
72	HT ₁	S12	8
73	HT ₂	S13	8
74	HT ₃	S14	8
75	Spare slot		8
76	Spare slot		8
77	IDO	I03	28
78	IDT ₁	I02	28
79	IDT ₂	I04	28
80	Spare for IDT ₃		16
81	MSL	R00	24
82	MSG	R02	24
83	MIS	M21	16

TABLE 2-39. DD SLOT ASSIGNMENTS (cont'd)

SLOT NUMBER	STATION NAME	CONSOLE NUMBER	SLOT LENGTH (NUMBER OF REGISTERS)
84	SSC	C54	24
85	SSC-D/O	C55	24
86	D/O	C56	24
87	FOI	C57	24
88	FOM	C58	24
89	CIO	C50	24
90	METO	C51	16
91	C&EO	C52	16
92	AC&W	C53	24
93	AAO	C59	28
94	Spare Staff	C60	16
95	CAA	C40	16
96	Spare liaison 2- observer	C42, C44, C45	16
97	SIMS	T12	28
98	M&R	T10	28
99	Spare slot		16
100	Spare slot		12
101	Spare slot		12
102	Spare slot		12
103	Spare slot		23
104	Spare slot		32
105	Test pattern		32
106	Not usable		1
Total of: 107 slots		124 DD indicator sections 9 spare slots	2,048

Slot number 1 is used by computer A for the DD indicator section at the utility control console (E20). Computer B utilizes the same slot number for its utility control console (E70). Console E20 is permanently connected to computer A; console E70 is permanently connected to computer B.

Slot number 2 is used by computer A for the DD indicator section at the simplex monitor console (E22), sometimes called the maintenance display console. The

same slot number is used by computer B for its simplex monitor console. Each simplex monitor console is permanently connected to its computer.

Slot number 3 is used by computer A for the DD indicator sections on the A halves of the duplex switching consoles (S77 and E77); the same slot number is used by computer B for the B halves of the duplex switching consoles (S78 and E78). Each duplex switching console comprises two consoles; one switching con-

sole includes consoles E77 and E78 and the other includes consoles S77 and S78.

Slot number 105, the test display, is selectable at any DD indicator section.

Slot number 106 and 0 exist for technical reasons only; neither one is ever displayed. Slot 106 contains one drum register; slot 0 contains no register.

2.5 PROVISIONS FOR EXPANSION OF DISPLAY CAPACITY

At present, 105 slots are allocated for display, of which 9 are spares. The total possible number of allocations for display is 124 slots. If future requirements

cause more slots to be used or cause an increase in the length of some slots, the total number of registers may exceed 2,047. The equipment is so designed that if 2,047 registers are not sufficient to display all digital display messages, the slots may be divided into two groups consisting of up to 2,047 registers each. The Central Computer System makes up messages for the first group (DD 1) and places them on the DD field of the MIXD drum for display on the DD indicator sections of group 1. It then places messages on the drum for the second group for display on the remaining indicator sections. Each group of displays remains on the DD CRT screens until replaced, without affecting the other group.

CHAPTER 7

OUTPUT OPERATIONS

7.1 INFORMATION FORMS

7.1.1 General

Output information as prepared by the Central Computer System program is formed into blocks referred to as bursts. A burst is composed of several words. Each word is composed of 32 bits (logically divided into the left and right halves) and a parity bit. The right half-word contains data for transmission, and the left half-word contains address information (a section address, register address, and burst number). The left half of each word is composed of the same type of address information. However, the construction of a right half-word depends on the type of burst of which the word is a part.

Altitude, time-to-go, target range, command heading, attack heading, or target bearing. Each type-of-message code and the meanings of the specific data contained in bits R8 through R15, as associated with each type of message, are shown in table 2-40.

7.1.3 Ground-to-Ground Information

A ground-to-ground (G/G) right half-word may be a part of one of three types G/G messages: forwardtelling, crosstelling, or height request. The data used to form either a forwardtelling or a crosstelling message can be varied as required by the program. Therefore, the type of forwardtelling or crosstelling data used to form each respective right half-word of a G/G message is not fixed. A typical forwardtelling or crosstelling

TABLE 2-40. G/A MESSAGE TYPES AND VALUES

TYPE	CODE (R4-R7)	UNITS	INCRE- MENTS	VALUES MAXIMUM	CODED
Altitude	0010	Feet	250	63,000	252
Time-to-go	0111	Minutes	1/4	31	124
		Seconds (less than 2 minutes)	1	127	127
Target range	1000	Miles	1/2	127	254
Command heading	1011	Degrees	Approx.1.4	360	256
Attack heading	1100	Degrees	Approx.1.4	360	256
Target bearing right	1110	Degrees	1	127	127
Target bearing left	1111	Degrees	1	127	127

7.1.2 Ground-to-Air Information

Each ground-to-air (G/A) right half-word contains three pieces of information; aircraft call letters, type of message, and specific data. (See fig. 1-187.) The aircraft call letters are the address of the interceptor to which the specific data is directed; they occupy bit positions RS through R3. The four bits allow a maximum of 16 interceptor addresses. The type-of-message code, which occupies bit positions R4 through R7 of each G/A right half-word, identifies the data represented by bits R8 through R15. The type-of-message code can in-

message might refer to a specific target and contain the following information:

- X and Y co-ordinates of the target position, 26 bits.
- Height of target, 10 bits
- Velocity of target, X and Y co-ordinates, 14 bits
- Identity of target, target number, flight size, and receiving central address, 30 bits combined.

Each right half-word of the typical forwardtelling or crosstelling message is formed of some part of the above

listed data. The one fixed requirement is that the receiving central address must occupy bit positions R11 through R14 of the fourth right half-word; if the message is to be received by all centrals on a telephone circuit, R15 (all receive bit) of that half-word must be a 1.

The information used to form each right half-word of a G/G height request message pertaining to a definite target is fixed as follows:

- Right half-word 1; RS and R1 are always zeros; R2 through R8 represent the old height of the target; R9 through R11 are the request number; R12 through R15 are the radar site address.
- Right half-word 2; RS and R1 are always zeros; R2 through R11 represent the Y co-ordinate of the target position; R12 through R15 are always zeros.
- Right half-word 3; RS and R1 are always zeros; R2 through R11 represent the X co-ordinate of the target position; R12 through R15 are always zeroes.
- Right half-word 4; RS and R1 are always zeros; R2 through R7 are spare bits, and, as spares, are zeros; R8 through R11 represent the type of request; R12 through R15 are always zeroes.
- Right half-word 5; presently not employed; contains all zeros. (See fig. 1-189.)

The old height inserted into right half-word 1 is formed in 500-foot increments. The Y and X co-ordinates are formed in 1/2-mile increments and occupy 10 bit positions (R2 through R11, right half-words 2 and 3, respectively). Of these 10 bits, nine are used to represent the numerical value of the co-ordinate, and the remaining bit represents the sign of the co-ordinate. Of the four bits (R12 through R15 of the first right half-word) allotted for the radar site address, three form the actual site address and the remaining bit is used to signify which of the two height-finder radars at the site is to receive the request. The types of requests available and the codes used to represent them are listed in table 2-41.

7.1.4 Teletype Information

A teletype (TTY) right half-word is composed of three five-bit teletype characters plus the RS bit. (See

TABLE 2-41. TYPE OF HEIGHT-FINDER REQUEST

TYPE	CODE (WORD 4, R8-R11)
Height information only, requested	0000
Last request, no reply required	0001
Special request, heading north	1000
Special request, heading northeast	1001
Special request, heading east	1010
Special request, heading southeast	1011
Special request, heading south	1100
Special request, heading southwest	1101
Special request, heading west	1110
Special request, heading northwest	1111
Spare	0010
Spare	0111

fig. 2-7.) The teletype codes used to form the major portion of a TTY right half-word vary according to the intelligence that is to be transmitted. (Refer to table 1-79.) One restriction on the forming of a TTY right half-word is that the RS bit must always be a 1 to enable proper parity checking of the word in the Output System.

7.2 MESSAGE PREPARATION

7.2.1 General

Output messages are prepared in the Central Computer System when required by current air defense situations. The Central Computer System program causes data (required by a remote unit to aid it in accomplishing its assigned function) to be formed into an output message.

Output messages vary in size. A G/A message intended for an airborne interceptor is composed of one right half-word. A G/G message intended for AN/FSQ-8 Combat Control Central, adjacent AN/FSQ-7 Combat Direction Centrals, or remote P sites is composed of five

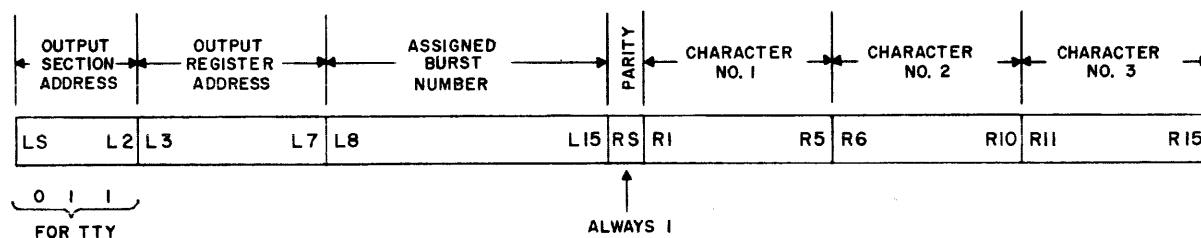


Figure 2-7. Teletype Message, Word Format

right half-words. A TTY message intended for anti-aircraft operational centrals, air bases, higher headquarters, or non-automatized direction centrals is composed of several right half-words. Each right half-word in a given TTY message is transmitted during a separate TTY burst period.

The type of data used to compose a right half-word is outlined in the preceding text. When the right half-word or words of an output message are completely formed, the left half-words are then prepared. A left half-word is composed of a 3-bit section address (L5 through L7), a 5-bit register address (L8 through L12), and an 8-bit burst number (L13 through L20). (Refer to Part 1, 7.1.3.2.) These three pieces of data are used to enable the Output System to transmit the right half-word to the proper destination at the required time. The section and register addresses combined determine the group of remote units to which the right half-word is transmitted. The burst number specifies the transmission interval.

7.2.2 Section and Register Addresses

During the formation of an output message, the remote unit intended to receive the message (e.g., data link transmitter, adjacent central, or anti-aircraft central) is known. The Central Computer System program determines the message destination from the type of data used to form the message or from additional information made available to the Central Computer System. Such additional information might be made available by a console operator through the use of the manual data input element of the Input System. The section and register addresses of a word are employed to insure that output data is transmitted by the Output System to the proper group of remote units.

A section address is a general address because it specifies only a group of destinations. For example, if an output message is intended for an airborne interceptor, the G/A section address is assigned to the message word. The G/A, G/G, and TTY storage section addresses are shown in table 1-76, together with five unassigned section addresses. During active operation, the five unassigned section addresses must never be used.

The register address completes the addressing of an output message word. The register address is more critical than the section address; it selects the channel over which the word is transmitted by the selected storage section. The selection of the channel determines the group of remote units that receive the output data. In addition to selecting the channel, the register address is used to place the right half-word in its proper position within the burst. This causes the bits of the right half-word to be received by the remote unit in a certain position within the block of received data.

A register address is allotted five bit positions. Thus, there can be a maximum of 32 possible register addresses. Since a maximum of only 26 register addresses are used, the last six combinations are unassignable and must never be used during active operation. The register addresses are listed in table 1-77.

The register address assigned to a G/A message word must place the word in the burst so that it is received by a data link transmitter at the proper time during receipt of data from the central. This allows the G/A message word data to modulate the subcarrier to which the intended airborne receiver is tuned. For example, if the intended airborne receiver is tuned to the first subcarrier, the first register address is assigned to the G/A message word.

All of the 26 register addresses are usable for addressing G/A message words. However, the 26th register address is reserved for data link transmitter test data.

A G/G message is composed of five words and might be transmitted by the G/G storage section over one of three channels. The register addresses assigned to the five words within a G/G message place the message in the proper G/G output storage array slot, so that the message is sent to the proper group of remote units (adjacent centrals, AN/FSQ-8 Combat Control Central, or P sites). The five words of a G/G message are assigned the sequential register addresses of the proper slot.

Only 15 of the usable 26 register addresses can be employed in addressing G/G words. Therefore, during active operation, the 16th through 26th register address must never be used to address G/G words.

Teletype data may be transmitted over one of 25 channels. Consequently, in assigning a register address to a TTY word, the address of the TTY storage register connected to the correct TTY channel must be assigned to the word. The 26th register address must never be assigned to a TTY word during active operation.

7.2.3 Burst Number Assignment

7.2.3.1 General

The assignment of a burst number to an output message determines the time of transmission of the message. Normally, the burst number assigned to a message is one higher than the number assigned to the last burst of the same type; however, this procedure is not always possible. The expected burst number can be assigned only if there is sufficient time to place the burst with that number onto the OB field and deliver it to the Output System at the appropriate time. In determining the proper burst number to assign, the program must find the answers to the following two questions:

- a. Are there bursts awaiting transmission via the

same storage section presently stored on the OB drum fields?

- b. Is there sufficient time within the present burst period to write the new burst onto the OB fields before it must be read by the Output System?

7.2.3.2 Number of Stored Bursts

In order to discover whether or not there are bursts awaiting transmission via the same storage section presently stored on the OB drum fields, a number of items of information must be made available to the program. These items include:

- B_L — the last assigned burst number
- N_L — the number of the burst period in progress when B_L was assigned
- N_c — the number of the burst period currently in progress.

The first two items can be retained in memory between iterations of the burst number assignment routine. The third item is obtained by reading the burst counter.

With these three items, B_c , the burst number to be assigned to the burst currently being prepared, can properly be determined.

Since burst numbers are assigned modulo 256 (the maximum count of a burst counter plus 1), there are six possible arrangement of N_L , N_c , B_L , and M , the modulus. (See fig. 2-8.) The first three cases are those in which bursts remain to be transmitted. The last three cases are those in which no bursts are still on the OB fields waiting for transmission. For any of the first three cases, B_c can be assigned the value $B_L + 1$. For any of the last three cases, B_c must be assigned at least the value $N_c + 1$.

The first case is relatively simple; bursts numbered from $N_c + 1$ to B_L remain on the OB fields and the burst being prepared can be assigned the burst number $B_L + 1$. This first case is recognized by the fact that $B_L > N_c$ and $N_c > N_L$. Cases II and III, shown in figure 2-8, part A, are essentially identical with case I, with the exception that one or more of the terms has gone beyond the modulus, thus altering their relative

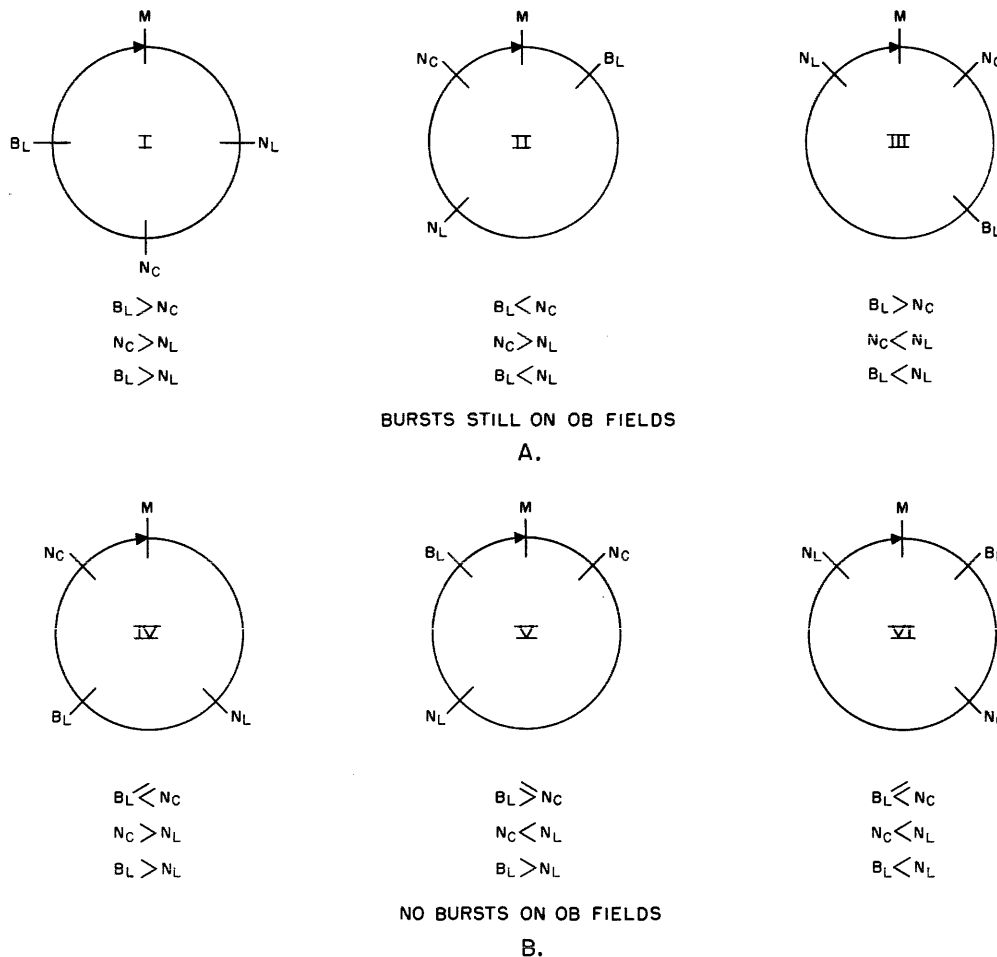


Figure 2-8. Burst Storage Determination

magnitudes without, however, modifying their relative positions.

In cases IV, V, and VI, shown in figure 2-8, part B, the present burst count, N_c , has passed the last assigned burst number, B_L , indicating that there are no bursts of the type being prepared still on the OB fields. In order to have the burst being prepared transmitted before the burst counter goes through most of a cycle, B_c must be given a value greater than N_c and at least $N_c + 1$. Further, to insure an interval of at least one burst period for placing the burst on the OB fields before it must be read, B_c should be given a value of $N_c + 2$.

The sequence of comparisons given in Figure 2-9 can be used to determine whether or not bursts remain on the OB fields waiting for transmission. For convenience in following the comparisons, each result is identified by a case number given in figure 2-8. For most cases, three comparisons are required; in cases I and VI, only two comparisons are required. As a result of these comparisons, B_c is assigned either the value $B_L + 1$ or $N_c + 1$. Further, the burst number assignment routine has an indication of the presence or absence of bursts on the OB fields which is necessary in connection with writing time, the time taken in placing a burst on the OB fields.

7.2.3.3 Writing Time

If no bursts of the type being prepared are stored on the OB fields, it is necessary to determine whether sufficient time is available to write the burst being prepared onto the OB fields before it is to be read by the Output System.

When the burst under consideration is a G/A burst, the program answers the second question by comparing the time required to write a burst onto the OB drum fields (a constant value) with the time remaining in the current G/A burst period. The time remaining in the current G/A burst period is found by subtracting the time that has elapsed since the start of the current G/A burst period from the fixed duration of a G/A burst period (0.25 second). The G/A elapsed time is found by the program through the reading of the output burst counter and the elapsed time counter. (Refer to Part 1, 7.2.1.4.) If a positive answer is obtained from the subtraction operation, it signifies that there is ample time remaining in the current G/A burst period to write the burst onto the OB drum fields.

When a G/G or TTY burst is under consideration for the assignment of a burst number, the time remaining in the current G/G or TTY burst period cannot be determined. (Refer to Part 1, 7.2.1.4.) Thus, the program is unable to answer the second question accurately. Consequently, to avoid any error, it assumes that there is not sufficient time remaining in the current G/G or TTY burst period to complete the OB writing.

Table 2-42 summarizes the considerations involved in the assignment of burst numbers to bursts prepared for transmission.

Once the burst number has been assigned to a burst, it may be necessary to modify the information within the burst, particularly if there is time-to-go information within the burst. The modifications necessary are described in Part 1, Chapter 7, 7.2.1.4 and 7.3.1.

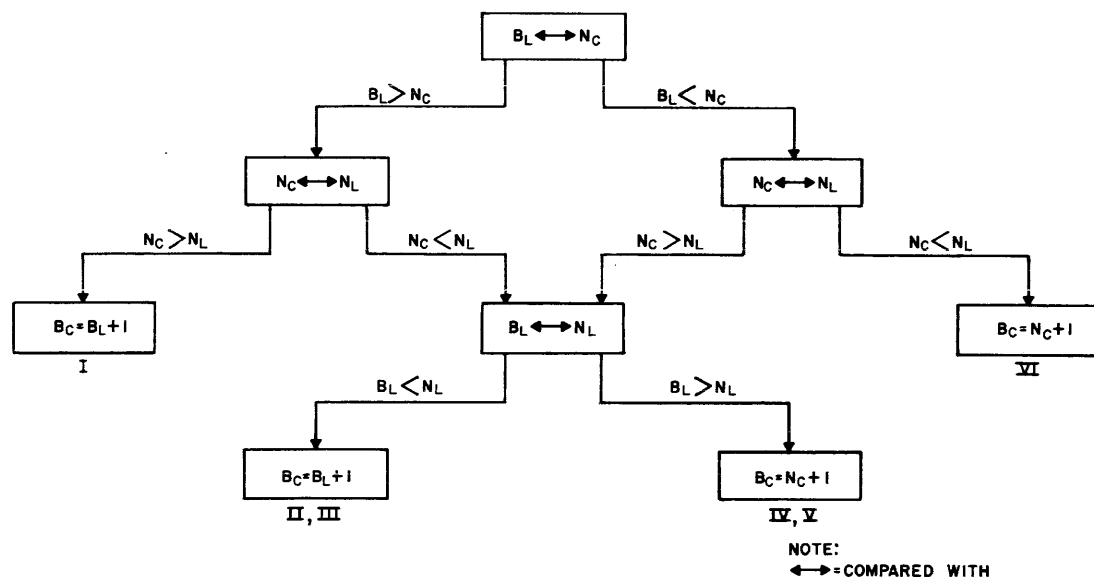


Figure 2-9. Burst Number Determination

TABLE 2-42. BURST NUMBER ASSIGNMENT

TYPE OF BURST	BURSTS STORED		WRITING TIME	
	YES	NO	SUFFICIENT	INSUFFICIENT
G/A	B_L+1	N_c+1	B_L+1	N_c+2
G/G	B_L+1	N_c+1	B_L+1	N_c+2
TTY	B_L+1	N_c+1	B_L+1	N_c+2

7.3 MESSAGE DELIVERY

7.3.1 General

Information intended for the Output System is written on the three output buffer (OB) fields of the LOG drum. These individual OB fields cannot be selected by the Central Computer System for writing. They are chosen consecutively by a counting circuit in the CD element. (A 120-microsecond delay is provided automatically between the switching of fields to allow the the selection circuits to settle.) Since a given output storage section cannot process information at the maximum rate of transfer (one word every 10.0 microseconds), the Central Computer System modifies the status mode of writing by placing words in alternate drum registers (odd or even). As the selection of odd or even drum registers is the only choice available, the Central Computer System sees the three OB fields as two logical fields: OB odd, selected by $SDR (30)_8$; and OB even, selected by $SDR (31)_8$. Words containing the same section address are therefore written in alternate drum registers.

On the OD side of the drum, the drum registers in the three OB fields are read consecutively by status identification. Since words containing the same section address and burst number appear in alternate drum registers, each section can receive words with a minimum interval of 20.0 microseconds between each word. However, words in adjacent registers can be transferred to different output sections.

7.3.2 Input-Output Transfers to the Output Buffer Fields

When transferring information to the OB fields, the Central Computer System specifies the writing of n words in the odd or even field. However, it is possible that n drum registers (odd or even, whichever have been selected) are not available for writing. In this case, after all of the selected drum registers have been examined, a disconnect pulse which clears the IO interlock and ends the writing operation is generated by circuits in the CD element of the Drum System. Assuming t

words were transferred, the IO word counter will contain the number $-n+t+1$ at the end of the IO operation. The Central Computer System must then determine whether the writing operation was ended by the Drum System and, if so, how many words remain to be transferred. When this has been determined, the original IO instructions are modified to complete the writing of $n - t$ words. Table 2-43 contains a representative program for insuring that all n words are written on the OB fields. The first three instructions are the original IO instructions which specify writing on the OB fields. After execution of the $WRT\ n$ instruction, a routine is performed which may last 80 milliseconds. This is sufficient time to allow n words to be written or, if n drum registers are not available, to allow the generation in the CD element of the disconnect pulse. No IO instruction may be given during this period since the first instruction in the following routine is a *Sense* instruction which determines the status of the IO interlock.

After transferring the contents of the IO word counter to the right accumulator, a *BRM* instruction examines the contents of RS. If RS is 0, all the specified words were transferred and the Central Computer System continues with the next program. If RS is 1, the writing operation was ended by the Drum System and $n - t$ words remain to be transferred. In this case, the program branches to a routine which determines how many words remain to be transferred. The original IO instructions are then modified to write the remaining words and another IO operation is initiated.

A *BRM* instruction (location 05.04.00) is given to determine whether the right accumulator, which, at this time, contains t , contains -0 . If so, then $t = 0$ and no words were transferred. The program then branches to an alarm routine. Assuming no branch ($t > 0$), the straight line routine modifies the original IO instructions and then branches back to location 01.01.00 to initiate the transfer of the words which were not written originally.

If the OB odd field was chosen the first time, the *SDR* instruction is modified to choose the OB even field. However, if all n words were not written in the OB odd field and if the Central Computer System writes the remaining words in the OB even field, the possibility exists that words containing the same burst number and section address may appear in adjacent drum registers. In this case, defective transmission may occur.

Another possibility is that after the second write operation is ended, the transfer of all of the original n words is still incomplete. Provisions may be made for this possibility by turning on a condition light during the first modification of the IO instructions. The condition light may be sensed during the second modification, branching the program to another alarm routine.

TABLE 2-43. OB FIELD WRITING, SAMPLE PROGRAM

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
01.01.00	<i>LDC</i>	x	
01.02.00	<i>SDR</i> (30) ₈ or <i>SDR</i> (31) ₈	—	Odd field. Even field.
01.03.00	<i>WRT</i>	n	
02.01.00	Start of any routine lasting approximately 80 milliseconds.		
02.01.00	<i>BSN</i> (14) ₈	06.01.00	Branches if IO interlock is on.
03.02.00	<i>CSW</i>	—	Transfers contents of IO word counter to right accumulator.
03.03.00	<i>BRM</i>	05.01.00	Branches if contents of right accumulator are negative.
04.01.00	Next program		
05.01.00	<i>ADD</i>	01.03.00	Adds <i>WRT</i> n instruction to accumulators leaving $t+1$ in right accumulator.
05.02.00	<i>SUB</i>	05.22.00	Leaves t in right accumulator.
05.04.00	<i>BRM</i>	07.01.00	
05.05.00	<i>RST</i>	05.24.00	Temporarily stores t .
05.06.00	<i>SUB</i>	05.22.00	Leaves $t-1$ in right accumulator.
05.07.00	<i>ADD</i>	01.01.00	Adds <i>LDC</i> x leaving $x+t-1$ in right accumulator.
05.08.00	<i>RST</i>	01.01.00	Replaces right half-word of <i>LDC</i> x instruction with $x+t-1$.
05.09.00	<i>CAD</i>	01.03.00	
05.10.00	<i>SUB</i>	05.24.00	Leaves $n-t$ in right accumulator.
05.11.00	<i>RST</i>	01.03.00	Replaces right half-word of <i>WRT</i> n instruction with $n-t$.
05.12.00	<i>CAD</i>	01.02.00	Whichever <i>SDR</i> instruction has been given is placed in accumulators.
05.13.00	<i>SUB</i>	05.23.00	If instruction in accumulators is <i>SDR</i> (30) ₈ , result in left accumulator is -0 ; if instruction in accumulators is <i>SDR</i> (31) ₈ , result in left accumulator is 1.
05.14.00	<i>BLM</i>	05.18.00	
05.15.00	<i>CAD</i>	05.23.00	
05.16.00	<i>LST</i>	01.02.00	Stores <i>SDR</i> (30) ₈ .
05.17.000	<i>BPX</i>	01.01.00	
05.18.00	<i>CAD</i>	01.02.00	
05.19.00	<i>ADD</i>	05.22.00	Changes <i>SDR</i> (30) ₈ to <i>SDR</i> (31) ₈ .

TABLE 2-43. OB FIELD WRITING, SAMPLE PROGRAM (cont'd)

LOCATION	INSTRUCTION		COMMENTS
	OPERATION	ADDRESS	
05.20.00	<i>LST</i>	01.02.00	Stores <i>SDR</i> (31) ₈ .
05.21.000	<i>BPX</i>	01.01.00	
05.22.00	0.00001	0.00001	
05.23.00	<i>SDR</i> (30) ₈	—	
05.24.00	—	—	Temporary storage for <i>t</i> .
06.01.00	Alarm routine		IO interlock is still on; possible malfunction.
07.01.00	Alarm routine		No words were transferred during writing operation.

CHAPTER 8

WARNING LIGHT OPERATION

8.1 GENERAL

The Warning Light System is utilized to notify personnel at various consoles of special or unusual programming operation or equipment malfunctions. This is accomplished under program controls by illuminating neon warning lights on the consoles or activating audible alarms.

The Warning Light System consists of three elements: the control element, the storage element, and the interconnection and indicator element. The control and storage elements are duplex; the interconnection and indicator element is simplex. The control element synchronizes Warning Light System operation with Central Computer System operation. The storage element consists of 256 flip-flops arranged in eight 32-bit flip-flop registers (called warning light registers) which control the operation of the warning devices. The interconnection and indicator element consists of a patchboard interconnection unit and the various indicators (neons and audible alarms) located at the consoles. Each indicator at each console is connected to a specific flip-flop in the warning light registers through the interconnection unit. The status of each indicator is determined by the status of the flip-flop to which it is assigned. Thus, if a flip-flop is set, the associated neon lights remain illuminated until the flip-flop is cleared. If the flip-flop also controls an audible alarm, the alarm is turned on by the flip-flop but may be turned off manually by depressing a momentary-contact pushbutton at the console. The alarm remains off until the controlling flip-flop is cleared and reset.

In order to change the condition of one or more

warning light, the Central Computer System must institute an IO operation, during which the contents of the eight warning light registers are replaced by the contents of eight core memory registers. The core memory registers contain the image of the contents of the warning light registers. If it is desired to change the condition of one or more warning lights, the bits in the core memory image are changed accordingly. Periodically, the program calls for the transfer of the eight words in core memory to the eight warning light registers. The words transferred to the warning light registers contain not only the 1 bits corresponding to the lights that are to be changed, but also the 1 bits corresponding to the lights that were on and which will remain unchanged. (A more complete discussion of the operation of the Warning Light System during an IO operation is given in Part 1, Ch 8.)

8.2 WARNING LIGHTS PROGRAMMING

Once every subframe, the contents of the warning light registers are replaced by the contents of the core memory image. In order to accomplish this, the Central Computer System must institute an IO operation. (Refer to table 2-44.)

Since the warning light registers are chosen sequentially during an IO operation, the *WRT* instruction may specify any number of words to be written. Thus, if only the first three warning light registers are to be written, the *WRT* instruction will specify the writing of only three words. Specifying the writing of more than eight words will result in the rewriting of $n-8$ warning light registers with undesirable results.

TABLE 2-44. IO PROGRAM FOR WARNING LIGHT SYSTEM

PROGRAM STEP	CODE		REMARKS
	OPERATION	ADDRESS	
1	<i>LDC</i>	x	Loads IO address counter with memory address x , which is the address of the first word in the core memory image
2	<i>SEL</i> (10) ₈	—	Specifies that the Warning Light System is to be involved in an IO process
3	<i>WRT</i>	n	Specifies the number of words to be written into the warning light registers and initiates the first breakout cycle

8.3 INFORMATION FLOW

Each bit position in all of the eight warning light registers controls a specific number of neons and audible alarms. (Refer to table 2-45.)

In addition to the assignments listed, each flip-flop controls another neon located on the duplex maintenance

console. These neons indicate the contents of the warning light registers.

Each neon and audible alarm at each console is assigned to a specific register and bit position in the warning light registers. Table 2-46 lists the assignments, identifies the console location, and, where a neon is specified, the neon number.

TABLE 2-45. LOADING OF EACH BIT POSITION

BIT POSITION	NEONS	AUDIBLE ALARMS	BIT POSITION	NEONS	AUDIBLE ALARMS
S	3	0	8	3	0
1	3	0	9	3	0
2	1	1	10	1	1
3	1	1	11	1	1
4	3	0	12	1	0
5	3	0	13	1	0
6	1	1	14	1	1
7	1	1	15	1	1

TABLE 2-46. WARNING LIGHT AND AUDIBLE ALARM BIT ASSIGNMENTS

BIT NO.		WORD 1	WORD 2	WORD 3	WORD 4	WORD 5	WORD 6	WORD 7	WORD 8
LS	N	S10-0	W81-0	S73-0	S73-0	S73-0	S72-0	E22-0	E20-0
	N			R00-0	R00-0	R02-0			
L1	N	S10-1	W81-1	S73-1	S73-1	S73-1	S72-1	E22-1	E20-1
	N			R00-1	R00-1	R02-1			
L2	A	S00	W13	W48	W00	W83	S30	P22	E20
	N	S00	W13	W48	W00	W83	S30	P22	E20
L3	A	S01	W12	W47	W01	W82	S32	P22	E11
	N	S01	W12	W47	W01	W82	S32	P22	E11
L4	N	S10-2	W81-2	S73-2	S73-2	S73-2	S72-2	E22-3	E20-2
	N			R00-2	R00-2	R02-3			
L5	N	S10-3	W81-3	S73-2	S73-3	S73-3	S72-3	E22-3	E20-3
	N			R00-3	R00-3	R02-3			
L6	A	S02	W23	W53	W20	W84	Spare	P23	Spare
	N	S02	W23	W53	W20	W84		P23	
L7	A	S03	W22	W51	W21	W85	Spare	P20	Spare
	N	S03	W22	W51	W21	W85		P20	
L8	N	S10-4	W81-4	S73-4	S73-4	S73-4	S72-4	E22-4	E20-4
	N			R00-4	R00-4	R02-4			
L9		Spare	Spare	Spare	Spare	Spare	Spare	Spare	Spare
L10	A	S04	W25	W54	W40	W89	S43	P21	I02
	N	S04	W25	W54	W40	W89	S43	P21	I02

TABLE 2-46. WARNING LIGHT AND AUDIBLE ALARM BIT ASSIGNMENTS (cont'd)

BIT NO.		WORD 1	WORD 2	WORD 3	WORD 4	WORD 5	WORD 6	WORD 7	WORD 8
L10	A	S60	W24	W53	W41	W88	S45	Spare	I04
	N	S60	W24	W53	W41	W88	S45		I04
L12	N	RM10-1	RM10-2	RM10-3	RM10-4	RM20-1	RM20-2	RM20-3	RM20-4
L13	N	RM11-1	RM11-2	RM11-3	RM11-4	RM21-1	RM21-2	RM21-3	RM21-4
L14	A	S63	W29	W64	W60	Spare	S50	P24	RM10
	N	S63	W29	W64	W60		S50	P24	R02-0
L15	A	Spare	W28	W63	W61	Spare	S52	P25	RM20
	N		W28	W63	W61		S52	P25	R02-5
RS	N	P24-0	W81-5	S73-5 R00-5	S73-5 R00-5	S73-5 R02-5	S72-5 P24-5	E22-5	E20-5
R1	N	P24-1	W81-6	S73-6 R00-6	S73-6 R00-6	S73-6 R02-6	S72-6 P24-6	E22-6	E20-6
	N								
R2	A	S70	Spare	W31	W66	W81	S35	Spare	RM11
	N	S70		W31	W66	W81	S35		R02-1
R3	A	D50	Spare	W30	W65	W80	S55	P30	RM21
	N	D50		W30	W65	W80	S55	P30	R02-6
R4	N	P24-2	W81-7	S73-7 R00-7	S73-7 R00-7	S73-7 R02-7	S72-7 P24-7	E22-7	E20-7
	N								
R5	N	P24-3	W81-8	S73-8 R00-8	S73-8 R00-8	S73-8 R02-8	S72-8 P24-8	E22-8	E20-8
	N								
R6	A	S11	W05	W33	W70	S72	S33	P33	RM12
	N	S11	W05	W33	W70	S72	S33	P33	R02-2
R7	A	S12	W04	W32	W69	S73	S53	T12	RM22
	N	S12	W04	W32	W69	S73	S53	T12	R02-7
R8	N	P24-4	W81-9	S73-9 R00-9	S73-9 R00-9	S73-9 R02-9	S72-9 P24-9	E22-9	E20-9
	N								
R9	N	RM12-1	RM12-2	RM12-3	RM12-4	RM22-1	RM22-2	RM22-3	RM22-4
R10	A	S13	W07	W44	W72	S75	Spare	T13	RM13
	N	S13	W07	W44	W72	S75		T13	R02-3
R11	A	S14	W06	W43	W71	S74	Spare	T21	RM23
	N	S14	W06	W43	W71	S74		T21	R02-8
R12	N	RM13-1	RM13-2	RM13-3	RM13-4	RM23-1	RM23-2	RM23-3	RM23-4
R13	N	RM14-1	RM14-2	RM14-3	RM14-4	RM24-1	RM24-2	RM24-3	RM24-4
R14	A	C55	W11	W46	Spare	R00	S23	T23	RM14
	N	C55	W11	W46		R00	S23	T23	R02-4
R15	A	C56	W10	W45		R02	S25	T10	RM24
	N	C56	W10	W45		R02	S25	T10	R02-9

CHAPTER 9

DUPLEX OPERATIONS

9.1 INTRODUCTION

While duplex operations were discussed generally in Chapter 1, this Chapter presents a summary of the operations connected directly with duplexing. These operations include:

- a. Intercommunication between the active and the standby computer
- b. Alarm indications from other computer
- c. Unit status evaluation using unit status cores in MDI matrix
- d. Input data selection control panel (keyboard) message interpretation and digital display preparation for duplex switching console

9.2 INTERCOMMUNICATION

Intercommunication generally involves the transfer of air defense data from the active computer to the standby computer for use by the standby computer in its safe data storage function. The active computer writes a block of information onto its own IC field using the *SDR* (26)₈ instruction to select the IC (own) field (with the address half of the *SDR* instruction specifying the starting field address). Upon completion of the writing operation, the active computer notifies the standby computer that information is available on its IC (other) field and indicates the nature of the information by using the appropriate *Operate* instruction: *PER* (10)₈, *PER* (11)₈, *PER* (12)₈, *PER* (13)₈. These four operations each set a particular sense unit in the other computer. Each sense unit may be assigned a particular meaning; i.e., may signify the availability of a particular type of information.

When the standby computer checks the sense units for intercommunication and finds one on, the standby computer proceeds to read the IC field on which the active computer wrote its information. The standby computer selects this field with an *SDR* (16)₈ *r* instruction. The *Sense* instruction codes, by which the standby computer recognizes the availability of information and the type of information, include *BSN* (43)₈ *x*, *BSN* (44)₈ *x*, *BSN* (45)₈ *x*, and *BSN* (46)₈ *x*, one sense code for each sense unit. When one of these *Sense* instructions finds its sense unit on, the sense unit is turned off and a branch of program control is executed. The branch may be to a reading routine. Upon completion of the reading of the IC (other) field, the standby computer notifies the

active computer that it can accept more information by using the selection code with an *Operate* instruction which turns on one of the intercommunication sense units in the active computer; these operate selection codes are 10₈ through 13₈. Again, once an intercommunication sense unit is turned on in the active computer, it can be turned off only by the appropriate *Sense* instruction being executed in the active computer.

It is also possible for the standby computer to supply data to the active computer via the IC (own) field of the standby Drum System. At present, transfers in this direction seem unlikely to be useful.

9.3 OTHER COMPUTER ALARMS

Each computer has two sense units which are set by alarms generated in the other computer. One sense unit, checked by a *BSN* (41)₈ *x*, is turned on by the occurrence of an overflow or a memory parity error in the other computer. If the sense unit is on when checked, a branch of program control is executed and the sense unit is turned on. It is important to realize that this alarm sense unit is set whenever an overflow or memory parity error occurs; neither programming of instructions which can cause overflow nor the various alarm control switches in the other computer have any effect.

The other sense unit set by alarms generated in the other computer is set by the occurrence of a tape or drum parity error in the other computer. This sense unit is checked by a *BSN* (42)₈ *x* instruction, and, if on, is turned off by it. The alarm control switches in the other Computer have no effect upon the setting of this sense unit.

These two sense units are useful for evaluation of the immediate reliability of the other computer. For example, the standby computer might, upon detecting one of these alarms, present a summary of its reliability status at the duplex switching console, enabling the operators at that console to decide upon the desirability of switching over more rapidly than might otherwise be possible.

9.4 UNIT STATUS EVALUATION

Simplex units within AN/FSQ-7 Combat Direction Central indicate their status, active or standby, to the computers by controlling unit status cores in the MDI matrix. Each simplex unit is assigned one core in both matrices. If the unit is in active status, its cores are set

to the 1 state. If the unit is in standby status, its cores are set to the 0 state. Thus, each computer can discover which simplex units are working with it by interpreting the states of their unit status cores; the active computer works with all simplex units whose status cores contain 1's, whereas the standby computer ignores all simplex unit whose cores contain 1's. (Refer to table 2-5-A.)

The active computer can make use of unit status core interpretation, particularly in controlling the radar input program. This program, which reads all radar data fields by status-identification, can be used to select the identification codes corresponding to the active channels, skipping the reading operations with those codes corresponding to standby channels. In this way, the operating time of this program is held to the time necessary for its operation without wasting 20 milliseconds on each reading operation which would attempt to obtain words supplied by standby channels that are not on the active radar data fields. Similarly, the active computer can use the unit status cores to determine

which display consoles within a group are active and, therefore, which of them are to receive forced displays.

9.5 SWITCHING CONSOLE OPERATIONS

The duplex switching console is serviced by special display makeup and input data selection control panel (keyboard) message interpretation operations. The two digital display tubes in the duplex switching console, one tube receiving display information from each computer, are used to display summary information on the reliability of each computer and on the activity of the active computer. This information is presented to facilitate the job of deciding on a switchover operation in the event of active computer unreliability. In effect, these displays provide program-generated information on reliability to supplement the condition and alarm indications provided on the console.

The input data selection control panels (keyboards) on the duplex switching console allow the operators at the console to request other information from each computer. (A complete description of the duplex switching console is given in Part 1, Ch. 9, Sec. 3.)

CHAPTER 10

MAINTENANCE OPERATIONS

SECTION 1

MAINTENANCE PROGRAMS

1.1 INTRODUCTION

Operations of all AN/FSQ-7 Combat Direction Central systems are affected and/or controlled by programs executed by the Central Computer System. Since so much of the central equipment is directly affected by programs, they form a valuable maintenance tool. Three types of programs are used for maintenance: reliability programs, diagnostic programs, and utility (sometimes called auxiliary) programs.

1.2 RELIABILITY PROGRAMS

1.2.1 Definition

Reliability programs are used to check the operation of specific portions of AN/FSQ-7 Combat Direction Central. Performance of these checks enables errors caused by circuit failure to be detected rapidly. Included in the error detection performed by reliability programs is the discovery of failures that may occur only under particular operating conditions, such as failures that appear at specific repetition rates and for certain combinations of bits. In order to discover these and other types of errors, reliability programs check logical operation, paths of information flow, timing, ability of equipment to function in all states, execution of instructions, etc. Because of their ability to perform these varied checks, equipment reliability programs can be used for both preventive and corrective maintenance.

1.2.2 Types

Reliability programs either check the logical functioning of a given circuit (or group of circuits) for all possible conditions, or exercise the individual components of the circuit being checked. Whichever technique is employed, it is assumed that associated circuits not directly checked by the program are in satisfactory operating condition.

Reliability programs fall into two categories, first order and second order. First order reliability programs are employed to check the operation of an entire system or element. Second order programs check the operation of individual parts of systems or elements such as registers, counters, and so on. First order programs usually include a number of second order programs.

1.2.3 Interpretation

If no errors are detected by a reliability program, it can usually be assumed that the equipment being checked at the time the program is run is in proper operating condition. However, it is highly desirable to extend the use of the program to determine which circuits present the highest probability of future failure. While no numerical values can as yet be assigned to failure probabilities for any given circuit or group of circuits, areas of suspicion can be detected by adding a technique known as marginal checking to reliability programs. Certain voltages in the equipment are varied either manually or by the program during program operation time. The magnitudes of these voltages can be varied within the operating limits of the circuits being checked or until these circuits fail.

Successful operation of a circuit during variations of a marginal checking voltage within operating limits indicates that the circuit checked will probably continue to operate reliably for some indeterminate period of time. However, checks of this kind do not provide a guarantee of prolonged operation, since components that do not exhibit deterioration can fail suddenly. Reliability programs which are run in conjunction with marginal checking techniques are usually shorter than those designed to run without marginal checking. However, most reliability programs are capable of being run both with and without marginal checking.

1.3 DIAGNOSTIC PROGRAMS

1.3.1 Definition

Diagnostic programs are corrective maintenance programs which localize malfunctions to as small an area of AN/FSQ-7 Combat Direction Central as possible. Diagnostic programs isolate errors to a specific pluggable unit or a relatively small number of pluggable units. In general, diagnostic programs are designed to isolate known failures, unlike reliability programs, which are designed to discover these failures. However, there is no clearly defined distinction between the two maintenance program types. Reliability programs can provide indications of the nature and location of a failure and

may actually be used as diagnostic programs. By the same token, diagnostic programs may indicate that a given portion of the equipment is operating reliably. The characteristics of reliability and diagnostic programs fall somewhere between the two extremes of overall check without failure localization and diagnosis with failure isolation.

Diagnostic programs are not always completely automatic. They frequently require the use of an operator to perform such functions as button pushing or oscilloscope checking. Whether completely automatic or semi-automatic, most diagnostic programs employ one of the following five techniques to isolate troubles.

1.3.2 Diagnostic Techniques

1.3.2.1 Increasing Area Check

The smallest possible number of circuits that can be checked by a program is tested. Then, successive checks are run in which a progressively greater number of circuits are added. The sequence of circuit checks is made so that each circuit that has passed its test is used in the checking of other circuits. This checking process is continued until all circuits that can be checked by a program have been tested. A failure in any given circuit, if it is a steady-state failure, is detected by the portion of the program testing that circuit.

1.3.2.2 Decreasing Area Check

A large area of the equipment is tested by the program. Thereafter, successive program steps check successively smaller and smaller portions of the equipment until a failure is no longer detectable. If checking of the large area reveals no error, a second large area is tested.

1.3.2.3 Overlapping Area Check

A number of program routines are run, each capable of testing a given area of equipment. These routines overlap with portions of a number of routines checking the same part of a given area of equipment. Thus, a detected failure is detected by all common areas of a number of program routines. Observation of the portions of program routines that detect a common failure reveals the area which is the source of malfunction.

1.3.2.4 Large Area Localization

In cases in which failures cannot be isolated by diagnostic programs previously discussed, special diagnostic programs are written to enable detection of the general equipment area in which a malfunction occurs. Thereafter, other techniques such as system tests, oscilloscope probing, etc. are used to isolate the trouble via a process of elimination.

1.3.2.5 Individual Circuit Check

Diagnostic program routines are written for relatively small areas of the equipment, with each area

having its own program. If a trouble is indicated during normal operation or by a reliability program, only the affected area is tested by the program; it is assumed that all other areas are free of trouble.

1.3.3 Marginal Checking with Diagnostic Programs

Marginal checking techniques can be used with most diagnostic programs. However, those diagnostic programs designed to work exclusively with marginal checking voltage variations are usually less complex than other diagnostic programs. Failures that are observable only during specific marginal checking voltage variations used by a diagnostic program are localized to the group of circuits affected by the voltage variations used.

1.4 UTILITY PROGRAMS

Utility programs, sometimes called auxiliary programs, are not confined to maintenance usage. Utility programs perform nonchecking functions such as loading programs, assembly programs, tracing programs, and simulation programs. Loading programs take programmed information from punched cards, tapes, or drums and transfer them into the Central Computer System core memory element. Assembly programs are used to prepare other programs for transfer to punch cards. Tracing programs provide printed records of registers to aid in the following of program operations. Simulation programs are used to pretest programs on other computers known to be operating without error. Simulation programs thus provide a means of detecting program errors (errors which are not due to equipment failure but are mistakes in programmed routines).

Utility programs can also be used to generate a particular pattern which is used to calibrate portions of the equipment and to exercise a particular group of circuits so that normal waveforms can be observed, or design data obtained.

1.5 MAINTENANCE PROGRAM IDENTIFICATION CODES

All programs (whether maintenance or operational) are prepared for insertion on punched cards. The information punched in columns 1 through 16 of each card identifies the program and the place of the card within the program. The exact coding used in this identification field and the various abbreviated entries used in this field for maintenance programs are described in table 2-47.

A sample of a code for a program is 01CNT-COMPO4B003. (The hyphen indicates a blank space.) This coding indicates that this is the third card in the second revision of the fourth program written for the Central Computer System.

TABLE 2-47. MAINTENANCE PROGRAM IDENTIFICATION CODES

COLUMN NUMBER	ASSIGNMENT	ABBREVIATED ENTRY	COLUMN NUMBER	ASSIGNMENT	ABBREVIATED ENTRY
1-2	Site number:			Crosstell input element	XTL
	XD-1	X1		Tape system	TAPES
	XD-2	X2		Drum System	DRUMS
	DC-1	O1		Manual data input element	MDI
	DC-2	O2		Digital display elements	DD DISP
3-10	Descriptive name; Central Computer System	CNT COMP		Situation display element	SD DISP
	Arithmetic element	ARITH		Symbolic assembly	SYM ASSY
	Memory buffer register	MEM BFR		Memory print	MEM PRINT
	A registers	A REG		Binary load	BIN LOAD
	Core memory element	MEMORY		Correct checksum	CHECKSUM
	Card machines	READER		Print tapes	TAPE PRT
		PRINTER	11-12	Program number	0 thru 99
		PUNCH	13	Revision status:	
	Output System	OUTPUTS		No revision	(blank)
	Gap-filler input element	GFI		1st revision	A
	Long-range radar input element	LRI		2nd revision	B
			14-16	Binary program card number	

SECTION 2

MARGINAL CHECKING

2.1 INTRODUCTION

2.1.1. General

Marginal checking is used in conjunction with programmed maintenance operations to improve the reliability of the equipment within AN/FSQ-7 Combat Direction Central. The physical facilities for marginal checking in this equipment have already been described. (Refer to Part 1, Ch. 10, Sec. 4.) This Section reviews the basic philosophy of marginal checking and describes the techniques necessary to its use.

2.1.2 Basic Reliability Testing

Preventive maintenance may be performed on the equipment within AN/FSQ-7 Combat Direction Central either with or without the use of marginal checking wherein supply voltages for the circuits under test are varied while they are tested. A programmed test which does not make use of marginal checking voltages is called a nonmarginal checking reliability program. In this instance, each logical system is tested, unit by unit, to detect the occurrence of errors when normal operating voltages are applied to circuits undergoing tests. If no errors occur during the execution of the program, it is assumed that errors occurring in further tests are a result of new conditions imposed by new tests. A marginal checking reliability program is defined in terms of input and output signal characteristics. If an input or output signal voltage amplitude is below the level of reliability, errors in information transfer or storage will result. Since the circuits have been designed with safety factors allowing specified limits for voltage variations, aging of components, and component deterioration, long-term reliability is assured.

A convenient method for determining circuit reliability is to vary a supply voltage to a tube element such as the plate, screen grid, control grid, or cathode. If, under these conditions, reliable information transfer is maintained (i.e., no errors occur), then the circuit is not in danger of failing immediately or before the next preventive tests are run. Marginal checking operations are based on the principle of varying supply voltages to the various tube elements. If a given portion of the system was reliable when tested by a nonmarginal checking reliability program, it is assumed that only those circuits which have had their supply voltages varied can possibly be at fault. By narrowing down the re-

liability program and progressively eliminating unaffected units, the faulty units are localized and isolated.

2.1.3 Definition of Terms

Many terms used in connection with marginal checking sound much alike but describe different units or functions. Therefore, in order to avoid confusion, the terms used in the subsequent text are defined:

- a. Service voltage is a normal operating potential applied from the regular power supply to the circuit elements during normal operating conditions.
- b. Marginal checking service voltage is the specific service voltage selected as the input voltage to the marginal checking system (+250, +150, +90, -150, and -300).
- c. Voltage group is a group of circuits within the entire system which is checked by one marginal checking service voltage.
- d. Marginal checking group is a large group of circuits having a common location and function.
- e. Margin is the operating limit of a circuit. A margin will gradually and constantly change as a result of variations due to aging of components. The voltage variation which causes the circuit to malfunction is called circuit margin.
- f. Circuit group is a group of circuits within a voltage group of a marginal checking group having approximately the same margin for a specific marginal checking service voltage.
- g. Line is a smaller group of circuits within a circuit group having the same function and the same approximate margin for a specific marginal checking service voltage.
- h. Excursion is the magnitude of the voltage variation used for marginal checking. This variation may be in either a positive or negative direction.
- i. Marginal checking test voltage is the sum obtained by adding or subtracting an excursion (voltage variation) to or from the service voltage used for marginal checking.

2.1.4 Checking for Imminent Failure

An imminent failure is indicated by a decreasing margin. A decreasing margin is an indication of component aging and deterioration. A margin is defined as the

excursion limit necessary to cause malfunction or complete failure of a circuit. In order to detect imminent failures in the system, margins are measured and examined periodically to determine the stage of component deterioration. The probability of failure of a circuit within a given time versus the margin of that circuit is graphically illustrated by the hypothetical curve shown in figure 2-10. As the circuit margin decreases from X_3 to X_1 , the probability of failure within a given time increases from Y_3 to Y_1 . Conversely, as the voltage required to cause failure increases from X_1 to X_3 , the probability of failure decreases. Note the exponential changes in the value of Y for equal increments of X . It is expected that the curve will have a knee, as shown by the diagram, such that, if the circuit margin has a value of X_2 volts the probability of failure is extremely low (Y_2), and that an increase of margin beyond the point X_2 (not X_3) will not result in an appreciable decrease in the probability of failure (Y_2 to Y_3).

A practical testing method, therefore, is to establish fixed excursions of such a magnitude that if no failures occur (i.e., circuit margin is not reached), the machine can be expected to operate without error until the next periodic test. This method of testing requires only the application of prescribed excursions to the marginal checking lines and, if no failures occur, circuit margins are assumed to be greater than the prescribed excursions. Testing with fixed excursions offers the following advantages over the routine measurement of margins:

- It is easier to apply fixed excursions to test for malfunctions than to apply ever-increasing excursions until malfunctions or failures occur.
- Collecting and amassing large volumes of data for analysis and comparison with previous data would not be needed. Only malfunctions need be investigated when using fixed excursions.

Once the circuit design has been established and the excursions are specified, it becomes a simple operation for the program to select the proper preset excursion. The machine may be exercised either by test programs, checking the results of each operation for error, or by special test equipment used to generate input signals and to measure or test output signals concurrently with the application of marginal checking voltages. Use of test programs for error detection has the following advantages over the use of special test equipment:

- The program can easily be changed when machine changes occur. In addition, the program can be constantly improved to meet new demands.
- No extra equipment is needed when programs

are run, since the computing facilities are already available.

- The program uses the machine in the manner in which the machine is normally used; test equipment can only simulate normal use.
- A checked-out program, usually contained on cards, is not subjected to deterioration as is test equipment.

2.2 CALCULATOR-CONTROLLED MARGINAL CHECKING

2.2.1 General

Calculator-controlled marginal checking is used for marginal checking in conjunction with reliability programs being executed by the Central Computer System. Each marginal checking operational sequence is initiated and controlled according to the requirements of the reliability program. When the program requires the application of an excursion to a group of circuits while their operation is tested by the program, a marginal checking control word is placed in the live register of test memory (address 20017₈). The contents of the live register drive decoding matrices which direct the marginal checking operation. The bit assignments within a marginal checking control word are given in table 2-48.

A programmed check of program-controlled marginal checking can be run before running any reliability programs which incorporate marginal checking. The programmed check may consist simply of observing the voltmeter on the marginal checking control panel of the maintenance console. A program could be run which would cause the voltmeter pointer to move slowly and smoothly from one extreme value to the other if the

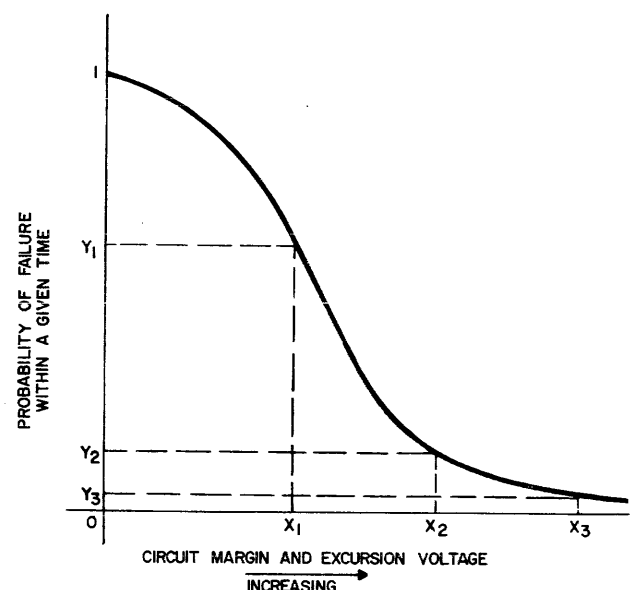


Figure 2-10. Probability of Failure Curve

TABLE 2-48. MARGINAL CHECKING CONTROL WORD BIT ASSIGNMENTS

BIT	SELECTION	CODE	ACTION
LS	Change or start excursion	1	Start excursion
		0	Change excursion
L1-L2	Restart after excursion applied	00	Load from drums
		01	Continue from 00000 ₈
		10	Continue from 20000 ₈
		11	Load from card reader
L3-L4	Restart after excursion removed	Same as L1-L2	
L5-L6	Time duration of excursion	00	Infinite
		01	3 seconds
		10	7 seconds
		11	30 seconds
L7	Polarity of excursion	0	Positive
		1	Negative
L8	Spare	Used by programmers for control decoding (safe limit). Does not affect MC system.	
L9-L12	Excursion magnitude	0000	0 volts
		0001	5
		0010	10
		0011	15
		0100	20
		0101	25
		0110	30 volts
		0111	35
		1000	40
		1001	50
		1010	60
		1011	70
		1100	75
		1101	80
		1110	90
		1111	100
L13-L15	Voltage group selection	001	+250
		010	+150
		011	+ 90

TABLE 2-48. MARGINAL CHECKING CONTROL WORD BIT ASSIGNMENTS (cont'd)

BIT	SELECTION	CODE	ACTION
RS-R3	Marginal checking group	100	—150
		101	—300
		0000	Spare
		0001	1 Memory
		0010	2 Arithmetic
		0011	3 Program and control
		0100	4 IO Control
		0101	5 Drums
		0110	6 Displays
		0111	7 LRI, GFI, XTL common
		1000	8 Outputs
R4-R9	Circuit group	1001	Simplex
		100000	A
		010000	B
		001000	C
		000100	D
		000010	E
R10-R15	Lines	000001	F
		100000	1
		010000	2
		001000	3
		000100	4
		000010	5
R10-R15	Simplex (RS-R3 = 1001)	000001	6
		100000	G
		010000	H
		001000	J
		000100	K
		000010	L
		000001	M

marginal checking controls are operating properly. Any malfunction of the control circuits will cause the volt-meter pointer to move erratically.

No provision has been made for programmed checking of the Power Supply System. Power supply troubles

are located through the use of trouble condition and trouble location indicators at the individual units or on the mimic panel. (Refer to Part 1, Ch. 10, Sec. 4.) Further, there are air and temperature alarms on the duplex maintenance console to indicate that ambient

temperature and air pressure in various parts of the equipment are within limits.

2.2.2 Testing Sequence

2.2.2.1 Basic Reliability Programs

A reliability test of Central Computer System operations is the first step performed in programmed preventive maintenance. When the operation of the Central Computer System has been found to be reliable, the system is then used in checking other systems. When the computer has been found to be reliable, the computer is then checked for imminent failures by means of marginal checking routines.

2.2.2.2 Reliability Programs with Marginal Checking

The primary marginal checking reliability routine is an executive program. The executive marginal checking program provides a flexible method of calculator control for use with marginal checking reliability and diagnostic test routines. It is a control program which selects the lines to be tested and specifies the excursion voltage to be used in checking them. It starts, stops, or recycles the program after either changing the lines or the excursion voltage, or both. The executive program specifies which part of the reliability program to run by indicating the address from which the program starts. Should the test routine fail to run successfully, the executive program provides a printed record of the marginal checking data. Since a faulty operation causes a print-out of information rather than a program halt, the operator has a printed record of all circuit failures when the checking routines are completed.

Diagnostic programs (corrective maintenance sequence) are then used to locate the faulty circuits. Upon completion of a corrective maintenance sequence, it is generally necessary to repeat the reliability checks to detect any new error or errors which may have been masked by previously detected errors. When all the circuits causing errors have been located and repaired, the executive program is applied to insure that all the circuit margins are up and to obtain data for establishing prescribed circuit margins. The excursion on a given marginal checking control word is increased until the test routine fails or until the safe limit on that circuit has been reached. The data thus obtained is then used to speed marginal checking operations. The program is started at the prescribed excursion for that circuit instead of starting at 0 volts and working up to the excursion which causes circuit failure. When the prescribed margins on all circuits have been determined, marginal checking operations are run on a go no-go basis, thus speeding all marginal checking operations.

During the operation of the marginal checking executive program, it is assumed that all the marginal

checking control circuits are operating reliably. If any doubt exists, the marginal checking circuits are tested before applying the marginal checking procedures.

2.2.2.3 Advantages of Programmed Marginal Checking

AN/FSQ-7 Combat Direction Central is able to direct information to the marginal checking system specifying the selection of the marginal checking lines and the magnitude and polarity of the excursion to be applied. The control program can then initiate the marginal checking excursion. The test program is restarted after the excursion reaches a steady state and terminates either at the conclusion of the program or after some predetermined time interval. (See table 2-11.) The control program is restarted when the excursion has been removed. The results of the test can then be printed out and a new test started.

This completely automatic operation has the advantage of speed and accuracy. Both the test program and the marginal checking control data are stored on punched cards or on magnetic tape. To carry out the complete test procedure, the operator need only load the cards or tape and, at the end of the test, examine the results on the printer. The high-speed computer equipment need not idle while the maintenance engineer manually operates the controls. The equipment cost of this automatic control of marginal checking is justified, since it eliminates human errors arising in the performance of routine tasks and maintenance time is held to a minimum.

2.2.3 Programmed Marginal Checking

2.2.3.1 Excursion Application

Once the control word for a marginal checking routine is placed in the live register by the marginal checking executive program, the excursion is initiated by the execution of the *PER* (21)₈ instruction. This instruction delivers the control word to the marginal checking controls, removes any previously applied excursion and (after sufficient delay to allow transients generated by the application of the excursion) generates a *control clear* command just prior to the program restart called for by the marginal checking control word. The delay between the execution of the *PER* (21)₈ instruction and the generation of the *control clear* command varies between 300 and 500 milliseconds. This interval could be used for further calculation but, since the delay time is variable, the *PER* (21)₈ instruction is usually followed by a *Program Stop* (*HLT*) instruction, leaving the Central Computer System idle until the program restart course.

The *control clear* command generated just before restart clears all computing and control flip-flops in the Central Computer System, allowing the restart program to begin from a known condition. Since the index regis-

ters are also cleared by this command, if they are to be used after the restart, their contents must be stored temporarily in core memory and then reset by the restart program itself.

2.2.3.2 Excursion Removal

Two separate *Operate* instructions are available for the removal of excursions. *PER* (22)₈ removes excursions applied to duplex equipment groups; *PER* (23)₈ removes excursions applied to simplex circuit groups. Each instruction generates a *control clear* command just prior to the restart after excursion removal called for in the marginal checking word. Since the delay between execution of either of these instructions and the generation of the *control clear* command is the same as for *PER* (21)₈, the same cautions apply in continuing calculation after execution of either instruction; calculation may be continued for 300 milliseconds after any of these instructions and then terminated by an *HLT* instruction.

It is conceivable that some marginal check programs will require margins on duplex equipment for some phases and on simplex equipment for other phases of the program. The application of these margins is accomplished by proper selection coding of the marginal checking word in conjunction with the *PER* (21)₈ instruction. However, removal requires that a *PER* (22)₈ instruction be used for duplex and a *PER* (23)₈ instruction for simplex. This necessitates the placing of both of these instructions in the same program. When programming this type of routine, *PER* (22)₈ and (23)₈ should never be given in successive order, since this causes two consecutive restarts, resulting in unpredictable operation. A method of discriminating between duplex and simplex margins should be included in this type of program, after which the applicable *PER* instruction to remove that margin may be given.

2.2.3.3 Excursion Detection

One of the functions of the *PER* (21)₈ instruction is to turn on the appropriate marginal checking sense unit, indicating that either duplex or simplex excursions are applied. The *PER* (22)₈ and (23)₈ instructions turn off the duplex and simplex sense units, indicating that excursions have been removed.

The *BSN* (20)₈ *x* instruction provides a means of interrogating the duplex marginal checking sense unit to determine if duplex excursions are on or off. The branch will be executed only if the sense unit is on and will not affect the setting of the sense unit. This sense unit and its associated *Sense* instruction is useful only when applying margins in calculator-controlled mode. Manually applied excursions will not affect the marginal checking sense unit which is normally in the cleared or off state. Hence, the *BSN* (20)₈ *x* instruction will not branch when in a manual mode.

The *BSN* (27)₈ *x* instruction operates in the same manner for simplex equipment and has its own marginal checking sense unit. Again, the branch will be executed only when the sense unit is on and will not be effective in the manual mode. The *BSN* (20)₈ and (27)₈ instructions provide a means of discriminating between margins applied to duplex and simplex equipment.

2.2.3.4 Use of LS Bit

The presence of a 1 in the left sign of a marginal checking word indicates that it is a start excursion word. When the *PER* (21)₈ instruction is executed, any previous excursion and voltage, equipment group, circuit group, and line selection will be removed. The new selection, as coded in the marginal checking word, will be made. In this type of word, all information pertinent to the margins to be applied must be specified.

Having made the desired selections by means of a start word, it is possible to alter the excursion magnitude, polarity, and the mode of restart after the excursion is applied by using a change word. The change word does not have provisions for selecting equipment groups, voltage groups, circuit groups, or lines, nor does it cause former selections to be removed. Therefore, any polarity or magnitude changes made affect the previously selected equipment and can only be made after a start word has been used to specify the desired selections.

The duration timer is restarted from zero by a change word but selection of duration is not affected. The same timer setting that was specified by the preceding start word will remain selected and will start retiming the new excursion from time 0. If the time duration specified by the start word expires, causing an automatic *PER* (22)₈ instruction, the next change word or words will be useless. This is also true following a programmed *PER* (22)₈ or (23)₈ instruction; a change word can be used only if excursions are still applied to the selected equipment.

When using a change word, the excursion magnitude, polarity, and restart (L1 and L2) must be specified with all other bits given as 0. This is true even when some information affected by a change word is to remain the same. Since a change word requires approximately half the time required for a start word, it may prove advantageous when time is a major problem.

2.2.3.5 Restarts

Bits L1 and L2 designate restart after excursion applied and bits L3 and L4 designate restart after excursion removed. These restarts are necessary to maintain program control. They are initiated after margins have been applied or removed and the *control clear* has been generated. The restart after excursion applied is initiated by the *PER* (21)₈ instruction. Restart after removal is initiated by the *PER* (22)₈ or (23)₈ instruc-

tion or by the duration timer running down to zero. In either case, there are four possible modes of restart.

The first of these is the load from drums mode in which the first 30₈ words from AM drum A field 1 are read into core memory locations 0 through 27₈. The program will then continue from location 00000₈, the first instruction loaded from the drum field. The continue from 00000₈ mode causes the program to start from core memory location 00000₈ after the excursion has been applied or removed. Continue from 20000₈ causes the program counter to be set to 20000₈, the first location in test memory, from which the Central Computer System will obtain its first instruction after the restart.

The load from card reader mode is equivalent to depressing the LOAD FROM CARD READER push-button on the duplex maintenance console. A break-in operation is initiated and one card of 30₈ binary words is loaded into core memory locations 0 through 27₈. The program then continues from memory location 00000₈. Since these four modes are available for both restarts, neither of which is dependent on the other, a great deal of versatility is obtainable through careful programming.

2.2.3.6 Time Duration

Bits L5 and L6 of the marginal checking word are devoted to selecting a desired time for an excursion to remain applied to selected equipment. There are four preset times available; infinite, 3 seconds, 7 seconds, and 30 seconds. The duration timers initiate an automatic PER (22)₈ or (23)₈ instruction in the event that these instructions have not been reached by the end of the desired time. These timers provide a means of retaining program control by initiating an automatic restart after excursions are removed. The PER (21)₈ instruction resets all timers to zero prior to the selection of a timer by the marginal checking word.

2.2.3.7 Polarity

Marginal checking excursions may be applied with positive or negative polarity. Bit L7 specifies the polarity of excursion to be applied as selected by the marginal checking control word. If this bit is 0, a positive excursion is called for; if the bit is a 1, a negative excursion is called for.

For certain selections of equipment, voltage, circuit group, and line, only a positive or negative excursion can be applied. The safe limit information in marginal checking breakdown charts indicates whether excursion of a particular polarity is permitted on a given line; if a positive excursion is not permitted, for example, the positive safe limit is indicated as 0 volts. Only if both safe limits have values other than zero can the line tolerate excursions of either polarity.

2.2.3.8 Safe Limit

Due to component considerations, certain selections may not have their supply voltages subjected to excursions beyond definite limits. There are five of these limits, 0, 25, 50, 75, and 100 volts, referred to as safe limits. Although the duplex marginal checking word has provisions for coding in bit L8 for one of two of these limits (100 or 25 volts), the code will not directly affect the marginal checking system. The safe limit bit is for decoding purposes only and is used to indicate at what magnitude the excursion should be removed. This is primarily a time-saving feature, since the system is interlocked to prevent the safe limit from being exceeded regardless of the coding or mode of operation.

Assume, for example, that excursions are to be applied from zero magnitude to failure to a line whose safe limit is 25 volts, and either that no safe limit is coded in the marginal checking word or that the control program has no provision for recognizing safe limits. If no failure is incurred between zero and a 25-volt excursion, the control program will continue to apply an excursion to that line. However, the magnitude of the excursion applied will not exceed 25 volts because of the safe limit feature of the marginal checking system. The 25-volt excursion will be applied 8 more times until the 100-volt word is reached. Since the time required to apply an excursion is in the hundreds of milliseconds, a large amount of time is wasted in this type of operation. This time can be saved by providing the control program with a means of recognizing the safe limit and removing excursions upon reaching that limit. The safe limits may be obtained from the marginal checking breakdown charts.

2.2.3.9 Excursion Magnitude

There are 16 possible excursion magnitudes, ranging from 0 to 100 volts, which are coded in bits L9 through L12 of the marginal checking word. These bits are decoded by the marginal checking system and the proper magnitude of excursion is applied to the selected line. If, for some reason, the coded magnitude exceeds the safe limit for the selected line, only the safe limit will be applied. Control programs may be designed to increase the magnitude on consecutive passes. In this manner, margins may be applied from a prescribed point to failure or until the safe limit is reached.

2.3 MARGINAL CHECKING BREAKDOWN CHARTS

These charts indicate the pluggable units and type of circuits affected by each selection as well as the safe limits associated with those selections. They are used extensively when forming the specifications for marginal checking programs and for isolating the cause of failures incurred during the running of these programs.

2.4 SPECIAL CONSIDERATIONS

It is possible to apply duplex and simplex excursions simultaneously, but this is not advisable. Before such a program is undertaken, careful consideration should be given to the circuits and equipment involved, since loss of program control could easily result from this type of operation. When applying margins to duplex and simplex equipment, the simplex (RS through R4 = 1001) excursion must be applied first. The duplex marginal checking word (RS through R4 = 0001 to

1000) may then be placed in the live register and a second *PER* (21)₈ instruction executed. Any change excursion word used while both excursions are on will affect only the duplex excursion. The order of removal is arbitrary, but the second *PER* instruction to remove excursions should not be given until the first has been completed. This applies also to the two *PER* (21)₈ instructions used to apply the simplex and duplex excursions.

OUTPUT SYSTEM																			
OUTPUT ALARMS								TTY OUTPUT PARITY COUNTERS								G/G OUTPUT PARITY COUNTERS			
<div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%;">NON SCH COMP</div> <div style="width: 50%;">PARITY NG</div> <div style="width: 50%;">ILL ADR</div> <div style="width: 50%;">ILL SEC</div> </div> <div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%;">G/A</div> <div style="width: 50%;">G/G</div> <div style="width: 50%;">TTY</div> <div style="width: 50%;">SPARE</div> </div> <div style="display: flex; flex-wrap: wrap;"> <div style="width: 50%;">SPARE</div> <div style="width: 50%;">SPARE</div> <div style="width: 50%;">SPARE</div> <div style="width: 50%;">SPARE</div> </div>								<div style="display: grid; grid-template-columns: repeat(4, 1fr); gap: 5px;"> <div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div> <div>7</div><div>8</div><div>9</div><div>10</div><div>11</div><div>12</div> <div>13</div><div>14</div><div>15</div><div>16</div><div>17</div><div>18</div> <div>19</div><div>20</div><div>21</div><div>22</div><div>23</div><div>24</div> <div>25</div><div>SPARE</div><div>SPARE</div><div>SPARE</div><div>SPARE</div><div>SPARE</div> </div>								<div style="display: flex; flex-wrap: wrap;"> <div>1</div><div>2</div><div>3</div><div>4</div> <div>5</div><div>SPARE</div><div>SPARE</div><div>SPARE</div> <div>SPARE</div><div>SPARE</div><div>SPARE</div><div>SPARE</div> <div>SPARE</div><div>SPARE</div><div>SPARE</div><div>SPARE</div> </div>			
G/A OUTPUT PARITY COUNTERS																			
<div style="display: flex; flex-wrap: wrap;"> <div>1</div><div>2</div><div>SPARE</div><div>SPARE</div> <div>SPARE</div><div>SPARE</div><div>SPARE</div><div>SPARE</div> </div>																			

G/A LOOPED TO LRI	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> 0-12 13-25 </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> </div>	<div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <div>SCALE OF 4 COUNTER</div> <div>SPARE</div> <div>PAUSE</div> </div> <div style="display: flex; justify-content: space-around;"> <div>1</div><div>2</div><div>3</div><div>4</div> </div>
SELECT G/G	
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> 12345 </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> </div> </div> <div style="width: 45%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> 1 & 2 3 & 4 </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> </div> </div> </div>	

TTY LOOPED TO TTY RCVR	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> 1-10 11-20 21-30 </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> <div style="width: 40px; height: 40px; border-radius: 50%;"></div> </div>	
<div style="text-align: center;"> </div>	

FRAME LOOP TE			
SINGLE CYCLE OUTPUTS	MASTER STOP	TEST TRANSFER	TEST SHIFT
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>
TEST TRANSFER DELAY	RESTART TO DRUMS SYNC	FF	SPARE
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>

G/A OUTPUT DATA			
1	2	SPARE	SPARE
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>
SPARE			
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>
SPARE			
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>
SPARE			
<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>	<div style="width: 40px; height: 40px; border-radius: 50%;"></div>

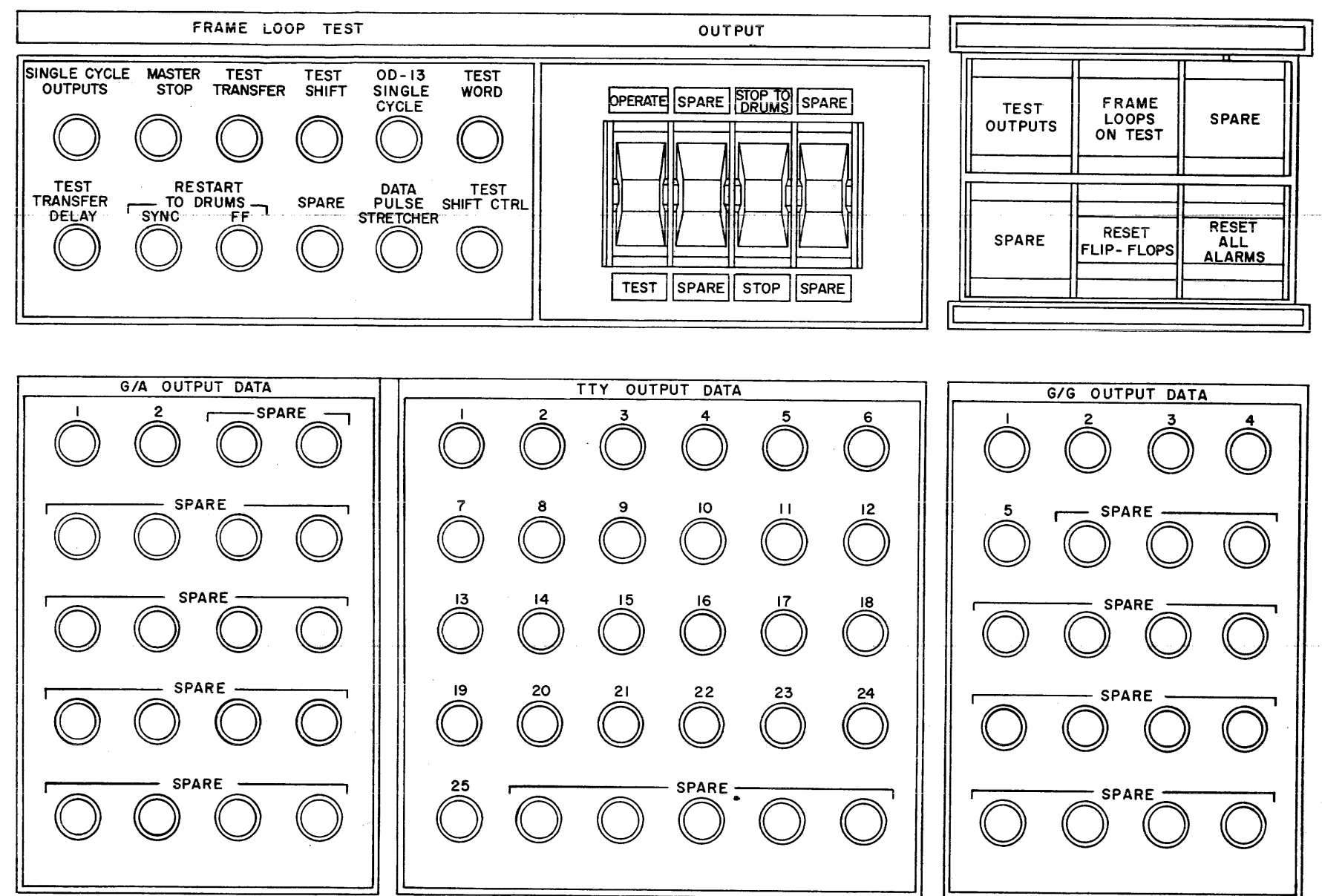
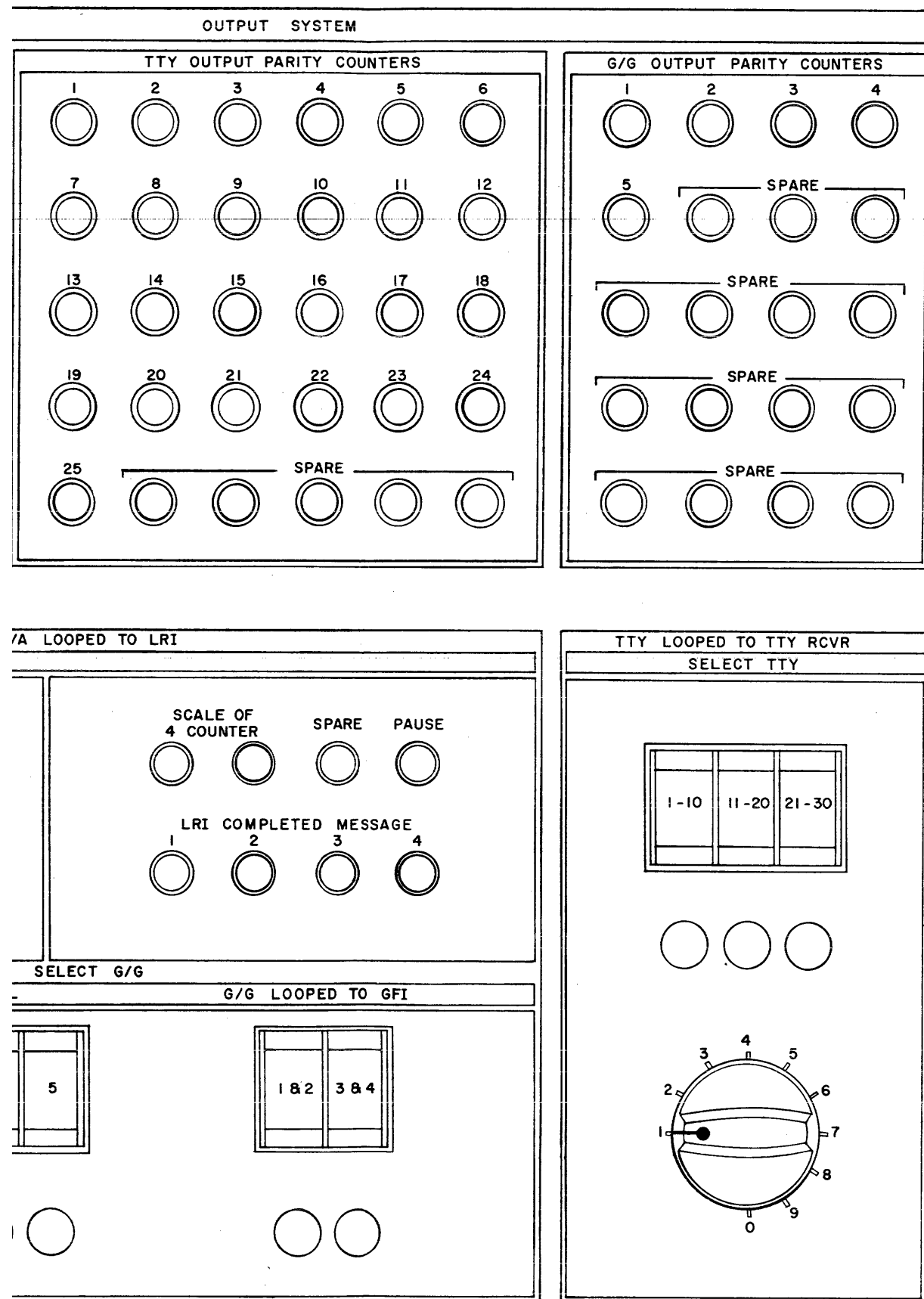


Figure 1-220. Output Test Controls and Indicators

- The following is a key to the symbology used in table 2-7.
- A – The execution of this instruction will be delayed if the IO interlock is on.
 - B – The time required to execute this instruction is 1 machine cycle plus $n+1$ times 0.5 ± 0.5 microseconds, for shifts greater than 0 places.
 - C – The time required to execute this instruction is 17 ± 0.5 microseconds.
 - D – The time required to execute this instruction is 51.5 ± 0.5 microseconds.
 - E – The time required to execute this instruction is 1 machine cycle plus $n-4$ times 0.5 ± 0.5 microseconds for every shift greater than 6 places.
 - U – Remains unchanged.
 - CL – Is cleared.
 - CH – Is changed; for the precise nature of the change refer to the Remarks column.
 - RAR – Right A register.
 - LAR – Left A register.
 - LAC – Left accumulator register.
 - RAC – Right accumulator register.
 - LBR – Left B register.
 - RBR – Right B register
 - LX – Left half-word at memory register x .
 - RX – Right half-word at memory register x .
 - PC – Program counter.
 - C – Original contents of register.
 - (s) – Index interval contains number to be subtracted from index register.
 - (u) – Unit specified by index interval bits.
 - (i) – Interleaving mode specified by index interval.
 - (o) – Instruction during which overflow can occur.
 - * – Indexable instructions.

TABLE 2-7. INSTRUCTION CODE SUMMARY

INSTRUCTION	ABBREVIATION	BINARY AND OCTONARY CODE	TIME	LAR	RAR	LAC	FINAL CONTENTS OF		RBR	LX	RX	REMARKS
							RAC	LBR				
1. Program Stop	HLT	000 000 0 000	12 usecs (A)	U	U	U	U	U	U	U	U	PC increased by 1 before stopping; delayed if IO interlock is on.
2. Extract	*ETR x	000 000 1 004	12 usecs	C(LX)	C(RX)	CH	CH	U	U	U	U	Products of logical multiplication between corresponding bits of memory register \times and accumulator registers appear in the accumulator registers.
3. Operate	PER (1) –	000 001-01-	12 usecs	U	U	U	U	U	U	U	U	Bits L10-L15 specify the operate unit.

TABLE 2-7. INSTRUCTION CODE SUMMARY (cont'd)

INSTRUCTION	ABBREVIATION	BINARY AND OCTONARY CODE	TIME	LAR	RAR	LAC	FINAL CONTENTS OF		RBR	LX	RX	REMARKS
							RAC	LBR				
4. Clear and Subtract Word Counter	CSW —	000 010 0 020	6 usecs	U	U	U	CH	U	U	U	U	The contents of the IO word counter are placed in the RAC.
5. Shift Left and Round	SLR (o) n	000 010 1 024	(B)	CL	CL	CH	CH	CL	CL	U	U	A shift left is performed followed by a round. The shifting is modulo 64. Possible overflow alarm.
6. Load B Registers	*LDB x	000 011 0 030	12 usecs	U	U	U	U	C(LX)	C(RX)	U	U	
7. Clear and Add	*CAD x	001 000 0 100	12 usecs	CL	CL	C(LX)	C(RX)	U	U	U	U	
8. Add	*ADD (o) x	001 000 1 104	12 usecs	CL	CL	C(LX)+ C(LAC)	C(LX)+ C(RAC)	U	U	U	U	Possible overflow alarm.
9. Twin and Add	*TAD (o) x	001 001 0 110	12 usecs	CL	CL	C(LX)+ C(LAC)	C(LX)± C(RAC)	U	U	U	U	Possible overflow alarm.
10. Add B Registers to Accumulator Registers	*ADB (o)	001 001 1 114	12 usecs	CL	CL	C(LB)+ C(LAC)	C(RB)+ C(RAC)	U	U	U	U	Possible overflow alarm. No address part.
11. Clear and Subtract	*CSU x	001 011 0 130	12 usecs	CL	CL	—C(LX)	—C(RX)	U	U	U	U	
12. Subtract	*SUB (o) x	001 011 1 134	12 usecs	CL	CL	C(LAC)— C(LX)	C(RAC)— C(RX)	U	U	U	U	Possible overflow alarm.
13. Twin and Subtract	*TSU (o) x	001 100 0 140	12 usecs	CL	CL	C(LAC)— C(LX)	C(RAC)— C(LX)	U	U	U	U	Possible overflow alarm.
14. Clear and Add Magnitude	*CAM x	001 110 0 160	12 usecs	CL	CL	C(LX)	C(RX)	U	U	U	U	
15. Difference Magnitudes	*DIM x	001 110 1 164	12 usecs	CL	CL	C(LAC) — C(LX)	C(RAC) — C(RX)	C(LAC)	C(RAC)	U	U	
16. Multiply	*MUL x	010 101 0 250	(C)	C(LX)	C(RX)	C(LAC)• C(LX) (Digits S-15)	C(RAC)• C(RX) (Digits S-15)	C(LAC)• C(LX) (Digits 16-30)	C(RAC)• C(RX) (Digits 16-30)	U	U	Signs of accumulator registers are signs of products. LBR 15 and RBR 15 are identical with LAC S and RAC S.
17. Twin and Multiply	*TMU x	010 101 1 254	(C)	C(LX)	C(LX)	C(LAC)• C(LX) (Digits S-15)	C(RAC)• C(LX) (Digits S-15)	C(LAC)• C(LX) (Digits 16-30)	C(RAC)• C(LX) (Digits 16-30)	U	U	Signs of accumulator registers are signs of products. LBR 15 and RBR 15 are identical with LAC S and RAC S.
18. Divide	*DVD x	010 110 0 260	(D)	± C(LX)	± C(RX)	CH	CH	C(LAC)+ C(LX)	C(RAC)+ C(RX)	U	U	Signs of accumulator registers are signs of products. 16-bit quotients are in B registers. Remainders are in accumulator registers.

TABLE 2-7. INSTRUCTION CODE SUMMARY (cont'd)

INSTRUCTION	ABBREVIATION	BINARY AND OCTONARY CODE	TIME	LAR	RAR	LAC	FINAL CONTENTS OF		RBR	LX	RX	REMARKS
							RAC	LBR				
19. <i>Twin and Divide</i>	*TDV <i>x</i>	010 110 1 264	(D)	$\pm C(LX) $	$\pm C(LX) $	CH	CH	C(LAC)+ C(LX)	C(RAC)+ C(LX)	U	U	Signs of accumulator registers are signs of products. 16-bit quotients are in B registers. Remainders are in accumulator registers.
20. <i>Store</i>	*FST <i>x</i>	011 010 1 324	12 usecs	U	U	U	U	U	U	C(LAC)	C(RAC)	
21. <i>Left Store</i>	*LST <i>x</i>	011 011 0 330	18 usecs	C(LX)	C(RX)	U	U	U	U	C(LAC)	U	
22. <i>Right Store</i>	*RST <i>x</i>	011 011 1 334	18 usecs	C(LX)	C(RX)	U	U	U	U	U	C(RAC)	
23. <i>Store Address</i>	*STA <i>x</i>	011 100 0 340	18 usecs	C(LX)	U	U	U	U	U	U	C(RAR)	
24. <i>Right Add One</i>	*AOR (o) <i>x</i>	011 100 1 344	18 usecs	C(LX)	C(RX)	U	C(RX)+2 ⁻¹⁵	U	U	U	C(RX)+2 ⁻¹⁵	Possibility of overflow alarm.
25. <i>Exchange</i>	*ECH <i>x</i>	011 101 0 350	18 usecs	C(LX)	C(RX)	C(LX)	C(RX)	U	U	C(LAC)	C(RAC)	
26. <i>Deposit</i>	*DEP <i>x</i>	011 110 0 360	18 usecs	CH	CH	CH	CH	U	U	CH	CH	Contents of memory contain bits deposited in accordance with the mask in B register. Accumulator registers and memory registers unaltered.
27. <i>Shift Left</i>	DSL <i>n</i>	100 000 0 400	(E)	U	U	CH	CH	CH	CH	U	U	LAC and LBR and RAC and RBR form 31-bit shifting accumulator; register signs not shifted. Vacated position filled by accumulator register signs.
28. <i>Shift Right</i>	DSR <i>n</i>	100 000 1 404	(E)	U	U	CH	CH	CH	CH	U	U	Accumulator registers shifted into B register. Bits shifted beyond LBR 15 and RBR 15 are lost. Vacated positions in accumulator registers filled by sign bits.
29. <i>Shift Accumulators Left</i>	ASL <i>n</i>	100 010 0 420	(E)	U	U	CH	CH	U	U	U	U	Each accumulator register is a separate shift register. Sign bit not shifted. Vacated positions filled by accumulator register signs.
30. <i>Shift Accumulators Right</i>	ASR <i>n</i>	100 010 1 424	(E)	U	U	CH	CH	U	U	U	U	Each accumulator register is a separate shift register. Bits shifted beyond LAC 15 and RAC 15 are lost. Vacated positions filled by signs.

TABLE 2-7. INSTRUCTION CODE SUMMARY (cont'd)

INSTRUCTION	ABREVIATION	BINARY AND OCTONARY CODE	TIME	LAR	RAR	LAC	FINAL CONTENTS OF		RBR	LX	RX	REMARKS
							RAC	LBR				
31. <i>Left Element Shift Right</i>	<i>LSR n</i>	100 100 0 440	(E)	U	U	CH	U	CH	U	U	U	LAC and LBR form a 31-bit shift register.
32. <i>Right Eelement Shift Right</i>	<i>RSR n</i>	100 100 1 444	(E)	U	U	U	CH	U	CH	U	U	RAC and RBR form a 31-bit shift register.
33. <i>Cycle Left</i>	<i>DCL n</i>	100 110 0 460	(E)	U	U	CH	CH	CH	CH	U	U	LAC and LBR together and RAC and RBR together form two 32-bit shift rings.
34. <i>Cycle Accumulators Left</i>	<i>FCL n</i>	100 111 0 470	(E)	U	U	CH	CH	U	U	U	U	RAC and LAC form a 32-bit ring.
35. <i>Branch and Index</i>	<i>BPX (s) x</i>	101 001-51-	6 usecs	CL	C(PC) if branch	U	U	U	U	U	U	If no index register or index register No. 3 is specified, this is an unconditional branch. If sign of index register is positive, subtract s from it and branch to x; program counter was stepped by 1 before put into RAR. If no branch, RAR is cleared.
36. <i>Sense</i>	<i>BSN (u) x</i>	101 010-52-	12 usecs	CL	C(PC) if branch	U	U	U	U	U	U	Bits L10-L15 specify sense unit. If conditions are met, branch to x and put contents of program counter stepped by1 in RAR. Otherwise, RAR is cleared
37. <i>Branch on Zero</i>	<i>BFZ x</i>	101 100 0 540	12 usecs	CL	C(PC) if branch +0 if no branch	U	U	U	U	U	U	Branch if LAC and RAC are zero; this means either positive or negative zero.
38. <i>Branch on Minus</i>	<i>BFM x</i>	101 100 1 544	6 usecs	CL	C(PC) if branch +0 if no branch	U	U	U	U	U	U	Branch if both accumulator registers are minus.
39. <i>Branch on Left Minus</i>	<i>BLM x</i>	101 101 0 550	6 usecs	CL	C(PC) if branch +0 if no branch	U	U	U	U	U	U	Branch if LAC is minus.
40. <i>Branch on Right Minus</i>	<i>BRM x</i>	101 101 1 554	6 usecs	CL	C(PC) if branch +0 if no branch	U	U	U	U	U	U	Branch if RAC is minus.

TABLE 2-7. INSTRUCTION CODE SUMMARY (cont'd)

INSTRUCTION	ABREVIATION	BINARY AND OCTONARY CODE	TIME	LAR	RAR	LAC	FINAL CONTENTS OF		RBR	LX	RX	REMARKS
							RAC	LBR				
41. <i>Load Input-Output Address Counter</i>	*LDC <i>n</i>	110 000 000 600	6 usecs (A)	U	U	U	U	U	U	U	U	The address <i>n</i> is placed in IO address counter. Delayed if IO interlock is on.
42. <i>Select Drum</i>	*SDR (u) <i>r</i>	110 001-61-	12 usecs (A)	U	U	U	U	U	U	U	U	Bits L10-L15 and R1 specify drum field. Address <i>r</i> may specify the drum register or identity. Delayed if IO interlock is on.
43. <i>Select</i>	*SEL (u) —	110 010-62-	12 usecs (A)	U	U	U	U	U	U	U	U	Bits L10-L15 specify input-output unit. This instruction does not select drums. Delayed if IO interlock is on.
44. <i>Read</i>	*RDS (i) <i>n</i>	110 111 0 670	6 usecs (A)	U	U	U	U	U	U	U	U	Initiate reading. Interleaving possible. Delayed if IO interlock on.
45. <i>Write</i>	*WRT (i) <i>n</i>	110 111 1 674	6 usecs (A)	U	U	U	U	U	U	U	U	Initiate writing. Interleaving possible. Delayed if IO interlock on.
46. <i>Reset Index Register</i>	XIN <i>n</i>	111 101 1 754	6 usecs	U	U	U	U	U	U	U	U	Contents <i>n</i> of address register put in specified index register (1, 2, 4, or 5).
47. <i>Reset Index Register from Right Accumulator</i>	XAC —	111 110 1 764	6 usecs	U	U	U	U	U	U	U	U	Contents of RAC put into specified index register.
48. <i>Add Index Register</i>	ADX <i>n</i>	111 111 0 770	6 usecs	U	$n + C(\text{index register})$	U	U	U	U	U	U	Address <i>n</i> added to contents of specified index register and sum placed in RAR. Any overflow is ignored.

GROUND TO AIR MSG

L5- L2 L3- L7 L8- L15
 SECTION REGISTER BURST NUMBER
 ADDRESS ADDRESS 8 BITS MAX -
 3 BITS 5 BITS USES ONLY 6

R5- R3 R4 - R7 R8-DATA R15
 INTERCEPTOR MESSAGE
 ADDRESS LABEL

TELETYPE RIGHT HALF WORD.

P L5- L2 L3- L7 L8- L15
 SECTION REGISTER BURST
 ADDRESS ADDRESS NUMBER

R5- R1 - R5) (R6- R10) (R11- R15)
 1 FIRST 2ND 3RD
 TELETYPE TELETYPE TELETYPE
 ALWAYS CODE CODE CODE

ODD PARITY

GROUND TO AIR MSG - III

L5- L2 L3- L7 L8- L15
 (1) Register Burst # 8 BITS
 Section (subcarrier) USES ONLY 6
 address R8- R15

SCALE INDICATORS (BIT 15 FOR ALT BIT 8 FOR TTG)
 R5- R3 R4- R7
 INT ADDRESS TTG = 7; RANGE = 8 SPEED = 4 IF SCALED R8- R14
 1- 15 BEARING = 14; ALT = 2 PHASE = 1 R9- R15
 COMMAND HEADING = 11 IF NOT SCALED R8- R15
 ATTACK " = 13

HEIGHT FINDER MESSAGE

① P L5- L2 L3- L7 L8- L15 R5-1 R2- R8 R9-11 R12- R15
 Section address register burst old height request site
 address address number number TO
 ② R5-1 R2- R11 R12- R15
 blank Y coordinate blank
 ③ R5-1 R2- R11 R12- R15
 blank X coordinate blank
 ④ R5-1 R2- R7 R8- R11 R12- R15
 blank blank request blank
 ⑤ R5-1 R2- R7 R8- R11 R12- R15
 blank blank blank blank

Time	Instruction	LAR	RAR	LAC	RAC	LBR	RBR	LX	RX
64	DSL	U	U	CH	CH	CH	CH	U	U
64	DSR	U	U	CH	CH	CH	CH	U	U
64	ASL	U	U	CH	CH	U	U	U	U
64	ASR	U	U	CH	CH	U	U	U	U
64	LSR	U	U	CH	U	CH	U	U	U
64	RSR	U	U	U	CH	U	CH	U	U
64	ICL	U	U	CH	CH	CH	CH	U	U
64	ICL	U	U	CH	CH	U	U	U	U
6	DPX	CL	C(PC)	U	U	U	U	U	U
12	BSN	CL	C(PC)	U	U	U	U	U	U
12	BFZ	CL	C(PC)	U	U	U	U	U	U
6	BFM	CL	C(PC)	U	U	U	U	U	U
6	BLM	CL	C(PC)	U	U	U	U	U	U
6	BRM	CL	C(PC)	U	U	U	U	U	U
6	LDC	U	U	U	U	U	U	U	U
12	SDR	U	U	U	U	U	U	U	U
12	SEL	U	U	U	U	U	U	U	U
6	RDS	U	U	U	U	U	U	U	U
6	WRT	U	U	U	U	U	U	U	U
6	KIN	U	U	U	U	U	U	U	U
6	KAC	U	U	U	U	U	U	U	U
6	ADX	U	C(ADR REG) C(INDEX REG.)	U	U	U	U	U	U

April 17, 1957

Time	Instruction	LAR	RAR	LAC	RAC	LBR	RBR	LX	RX
12	HLT	U	U	U	U	U	U	U	U
12	ETR	C(LX)	C(RX)	C(LX)-C(LAC)	C(RX)-C(RAC)	U	U	U	U
12	PER	U	U	U	U	U	U	U	U
6	CSW	U	U	U	CH	U	U	U	U
64	SLR	CL	CL	CH	CH	CL	CL	U	U
12	LDB	U	U	U	U	C(LX)	C(RX)	U	U
12	GAD	CL	CL	C(LX)	C(RX)	U	U	U	U
12	ADD	CL	CL	C(LX) / C(LAC)	C(RX) / C(RAC)	U	U	U	U
12	1AD	CL	CL	C(LX) / C(LAC)	C(LX) / C(RAC)	U	U	U	U
12	ADB	CL	CL	C(LB) / C(LAC)	C(RB) / C(RAC)	U	U	U	U
12	CSU	CL	CL	-C(LX)	-C(RX)	U	U	U	U
12	SUB	CL	CL	C(LAC) - C(LX)	C(RAC) - C(RX)	U	U	U	U
12	TSU	CL	CL	C(LAC) - C(LX)	C(RAC) - C(LX)	U	U	U	U
12	CAM	CL	CL	C(LX)	C(RX)	U	U	U	U
12	DIM	CL	CL	C(LAC) C(LX)	C(RAC) - C(RX)	C(LAC)	C(RAC)	U	U
17	MUL	C(LX)	C(RX)	C(LAC) - C(LX)	C(RAC) - C(RX)	C(LAC)	C(RAC)	U	U
17	TMU	C(LX)	C(LX)	C(LAC) - C(LX)	C(RAC) - C(LX)	C(LX)	C(LX)	U	U
51	DVD	C(LX)	C(RX)	CH	CH	C(LAC)	C(RAC)	U	U
51	TDV	C(LX)	C(LX)	CH	CH	+C(LX)	+C(RX)	U	U
12	FST	U	U	U	U	+C(LX)	+C(LX)	C(LAC)	C(RAC)
18	LST	C(LX)	C(RX)	U	U	U	U	C(LAC)	U
18	RST	C(LX)	C(RX)	U	U	U	U	U	C(RAC)
18	STA	C(LX)	U	U	U	U	U	U	C(RAR)
18	AOR	C(LX)	C(RX)	U	C(RX) / 1	U	U	U	C(RX) / 1
18	ECH	C(LX)	C(RX)	C(LX)	C(RX)	U	U	C(LAC)	C(RAC)
18	DEP	CH	CH	CH	CH	U	U	CH	CH

April 17, 1957