File No. 2922-13
GL24-3597-1

Systems

IBM 2922 Programmable Terminal RPQ 810563, 810564, and 810565 Component Description

IBM®

PREFACE

This publication for systems planners, programmers, and operators, presents the functional description and operating procedures of the IBM 2922 Programmable Terminal. The 2922 consists of a 2922-1 Terminal Control Unit, RPQ 810563, cable-connected to a 2922-2 Terminal Printer, RPQ 810564, and a 2922-3 Terminal Card Reader, RPQ 810565. In addition, the IBM 2152 Printer-Keyboard, RPQ 810581, and the IBM 1442 Card Punch Model 5, RPQ AD1629, can be attached (optional). In this publication:

- Chapter 1 describes the 2922-1:data formats, instructions, time-sharing, arithmetic and logical functions, and gives programming suggestions and terminal console operations.
- Chapter 2 presents operating procedures and associated instructions for the reader, printer, and optional printer-keyboard, and an optional card punch.
- Chapter 3 discusses the binary synchronous communications adapter: operating principles, transmission codes, instruction sets, control characters, and sense information.
- Appendixes contain reference information: conversion tables, operation and condition codes, instruction examples, stop codes, and a glossary.

Second Edition (December 1972)

This is a reprint of GL24-3597-0 incorporating changes released in Technical Newsletter GN24-0500, dated September 22, 1972.

Changes or additions to the specifications contained in this publication are continually being made. When using this publication in connection with the service or operation of IBM equipment, check with your local IBM branch office to ensure that you have the latest edition.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This manual has been prepared by the IBM System Products Division, Product Publications, Dept. K10, P.O. Box 6, Endicott, New York 13760. A form has been provided at the back of this publication for reader's comments. If the form has been removed, comments may be directed to the above address. Comments become the property of IBM.

CONTENTS

INTRODUCTION

The IBM 2922 Programmable Terminal is a communications system designed for teleprocessing applications that require extensive printouts. It can be used for batch data entry and to make inquiries into a computer's data base. The binary synchronous method of transmission is employed to exchange information over telephone lines with either an IBM System/360 or an IBM System/370 (via a 2701 Data Adapter Unit or 2703 Transmission Control); or with another 2922 terminal.

The IBM 2922 Programmable Terminal consists of interconnected functional units operating under the control of a series of instructions, called a program, stored in the IBM 2922-1 Terminal Control unit. Main storage access to one byte is 3.6 microseconds. Figure 1-1 shows a typical (schematic) configuration of units. The 2922-1 Terminal Control Unit is the central figure in the group, and attached to it is the IBM 2922-2 Terminal Printer and the IBM 2922-3 Terminal Card Reader. The IBM 2152 Printer-Keyboard and the IBM 1442 Card Punch Model 5 are available as options.

Data is entered through the terminal card reader, then transmitted to the central computer. After processing the jobs, the central computer can transmit reports and listings back to the remote terminal for rapid printing.

An optional 2152 Printer-Keyboard can provide interaction between the terminal and the central computer for inquiry and update, job status analysis, and program development. An optional 1442 Card Punch can provide punched-card output, serially, at a maximum rate of 160 columns per second.

The programmable terminal uses 36 System/360 compatible machine instructions that work with the microprogram and hardware circuitry to accomplish the assigned taks.

This reference manual provides detailed information on the operation and characteristics of the programmable terminal. The 2922-1 Terminal Control Unit is hereafter referred to as "Control Unit" and the Programmable Terminal is hereafter referred to as "Terminal."

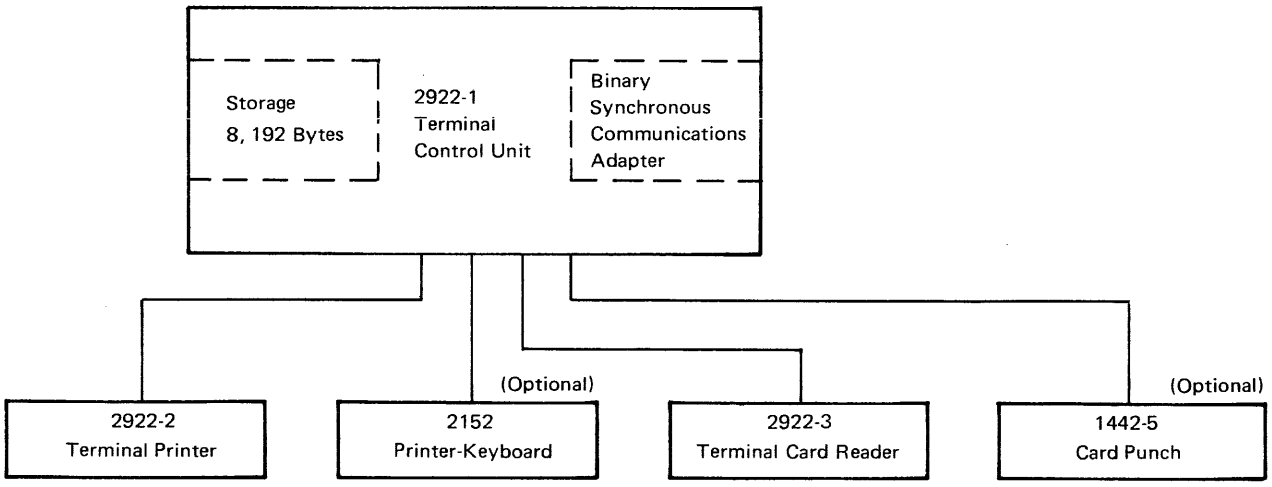


Figure 1-1. System Configuration

TERMINAL CONTROL UNIT

MAIN STORAGE

Main storage consists of 8,192 positions of magnetic core storage. Each position has an address and contains an eight-bit unit of information referred to as a byte. Coded combinations of bits within a byte can represent alphabetic, numeric, binary, or logical data.

Main storage is used to hold all the data that is to be operated upon, or processed, at a given time. It also holds the instructions, or program, which control the operation of the terminal. For each individual job performed by the terminal, certain portions of main storage are assigned to store instructions, and certain portions to store data to be processed.

GENERAL REGISTERS

Eight general purpose registers, small auxiliary storage units, are provided for temporary storage of small amounts of data. Each register is the equivalent of two bytes (one halfword) and is loaded or unloaded under the control of the stored program. Information may flow from register to register, main storage to register, or from register to main storage.

The registers are numbered 8-15 and are selected by the four-bit R or B field of an instruction.

PARITY CHECKING

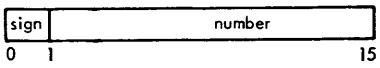
To ensure the accurate transfer of data, an extra (parity) bit is generated for each four bits transferred to or from main storage or register storage. The parity bit is added during transfer, if it is needed, to maintain an odd number of bits. The bit count is monitored continuously, and any missing or extra bits which result in an even number of bits cause a control unit parity error.

DATA FORMATS

The basic unit of addressable data is an eight-bit byte. Each address contains eight bits of data and can be considered a byte boundary.

The byte is divided into two sections of four bits each. A check (parity) bit is provided for each four bits.

Halfword Binary Number

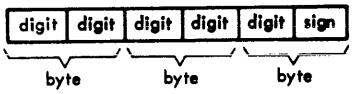


A halfword binary number has a fixed length of two bytes (16 bits). The leftmost bit is reserved for sign (+ or -) control.

Decimal Number

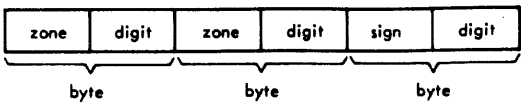
A decimal number may be in either of two forms: packed decimal or zoned decimal.

Packed Decimal: This format allows two numeric digits to be stored within one eight-bit byte.



An eight-bit byte may contain two numeric digits, except in the case of the rightmost byte, which has a sign to the right. Variable field length in this format allows for fields up to 16 bytes in length.

Zoned Decimal: This format contains one digit in the rightmost four positions of each byte.

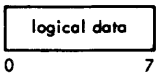


The left four bits of each byte in this format are called zone bits, and are not used except for the low-order (rightmost) byte which uses them for the sign. Zone bits do not affect the value of the numeric digit contained in the right four bits of the byte.

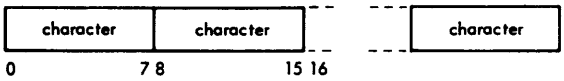
Logical Data

Logical data may be contained in the instruction or it may reside in core storage as an operand (an operand is data contained in, or addressed by the instruction and used to execute the instruction). Logical data may have a fixed length of one byte or may be variable in length up to a maximum of 256 bytes.

Fixed-Length Logical Data



Variable-Length Logical Data



INSTRUCTION FORMAT

The instruction format specifies the length of the instruction and the type of operation to be performed. The length of the instruction can be one, two, or three halfwords. The types of instruction formats are RR, RX, SI, and SS (Figure 1-2). RR Format: Denotes a register-to-register operation.

RX Format: Denotes a register-tc-storage or a storage-to-register operation. In this format, bits 12 through 15 must be zero.

SI Format: Denotes a stoage-immediate operation. In this format the I2 field of the instruction is the second operand.

SS Format: Denotes a storage-to-storage operation.

In each format, the first instruction halfword consists of two parts. The first byte contains the operation code (op code). The length and format of an instruction are specified by the first two bits of the operation code. The second byte may be used to contain data, specify operand lengths, or specify registers to be used by the operation.

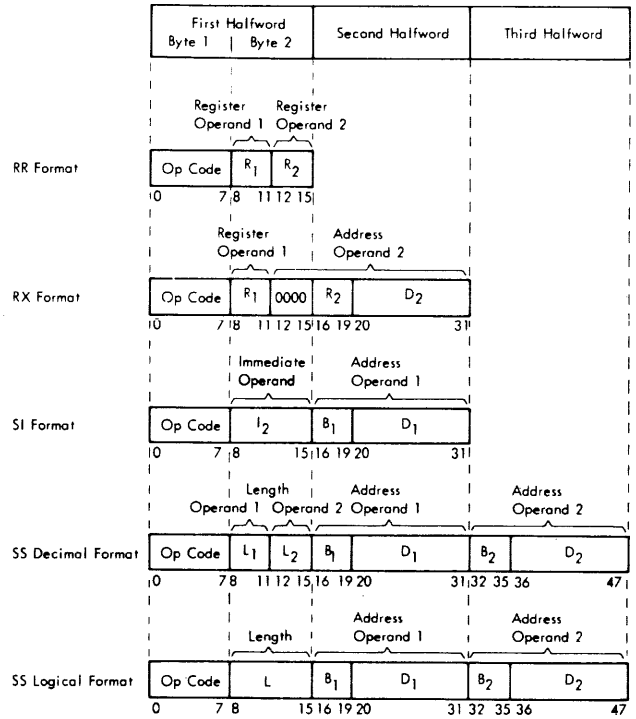


Figure 1-2. Instruction Formats

INSTRUCTION EXECUTION

For purposes of describing the execution of instructions, operands are designated as first and second operands.

These names refer to the manner in which the operands participate. The operand to which a field in an instruction format applies is generally denoted by the number following the code name of the field, for example, R1, B1, L2, D2. Normally, the operation of the terminal is controlled by instructions taken in sequence. An instruction is fetched from a location specified by the current instruction address. The current instruction address itself is located in the Program Status Word (PSW). After the fetch operation, the current instruction address is increased by the number of bytes in the fetched instruction to enable addressing of the next instruction in sequence.

The instruction is then executed by adding, subtracting, multiplying etc. The result that is thus obtained usually replaces operand 1. Upon the execution of certain instructions, a condition code, which reflects the nature of the result, is set into the PSW.

Subsequently, the updated address in the PSW is used to read out the next instruction from main storage and the processing continues.

INFORMATION POSITIONING

Byte locations in storage are consecutively numbered from 0, each number being considered the address of the corresponding byte. Bytes may be handled singly or strung together in fields. A group of two consecutive bytes is called a 'halfword'. The location of any field or group of bytes is specified by the address of the leftmost byte.

Information positioned in storage may be in fixed-length format or variable-length format. The length of fields is either implied by the operation to be performed or stated explicitly as part of the instruction. When the length is implied, the information is said to have a fixed length, which can be one, two, or four bytes.

Fixed-length fields must be located in main storage on an integral (halfword) boundary (Figure 1-3) for that unit of information. A boundary is called integral for a unit of information when its storage address is a multiple of the length of the unit in bytes. For example, a halfword (two bytes) must have an address that is a multiple of the number 2.

In the control unit unit, all instructions and all data with fixed word length (operands addressed by RX-format instructions) must begin on a halfword boundary. An instruction (or data with fixed word length) is properly located at a halfword boundary when its address is even or, in other words, when the low-order bit of the address is zero. An improperly placed instruction causes an error stop.

When the length of a field is not implied by the operation code, but is stated explicitly, the information is said to have a variable-length field.

Within any instruction format or any fixed-length operand format, the bits are consecutively numbered from left to right starting with bit number 0.

Variable-length fields are not restricted to halfword boundaries, and may contain up to 256 bytes. Length is variable in increments of one byte.

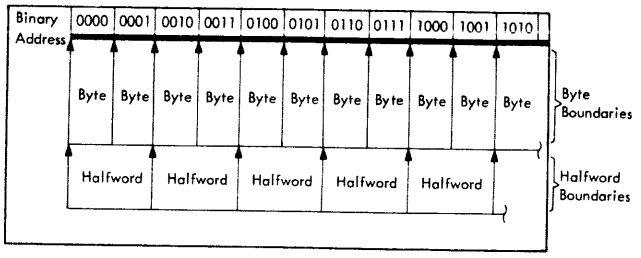


Figure 1-3. Integral Boundaries

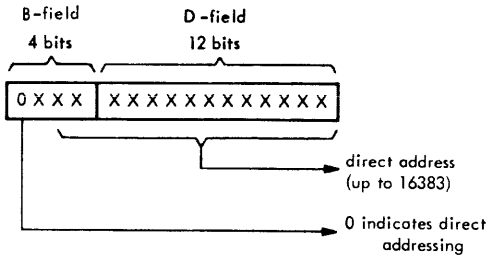
ADDRESSING

Byte locations in storage are expressed in binary form and consecutively numbered from 0000 to the upper limit of available storage. Appendix D explains the binary number system, and Appendix E contains the hexadecimal representation for addresses 0000 to 4095. The first 144 bytes (bytes 0000-0143) are reserved for internal control and are not available to the program. The location of any field or group of bytes is specified by the address of the leftmost byte.

An address used to refer to main storage may be specified by either of two methods; direct addressing or effective address generation.

Note: Neither a direct address nor an effective address can be zero, pertain to the protected area, or exceed the storage capacity.

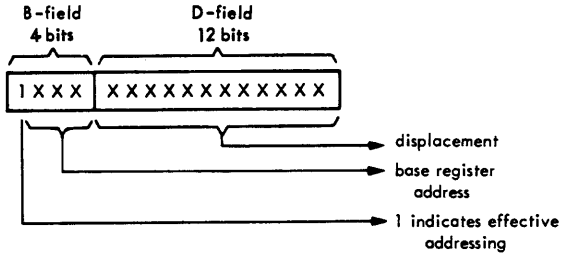
Direct Addressing: Direct addressing is used when the high-order bit in the B-field of an instruction is zero.



When the direct addressing method is employed, the low-order 14 bits of the combined B- and D-fields are used to refer directly to byte locations in main storage. The 12 binary bits in the D-field allow an address specification of up to 4095. To address additional

storage, the adjacent two bits in the low-order position of the B-field are used, allowing address specification of up to 8192.

Effective Addressing: Effecitve addressing is used when the high-order bit in the B-field of an instruction is one.



In the effective-address generation method, the contents of the general register specified by the B-field of an instruction, are added to the contents of the D-field of the instruction to form the effective address. The content of the general register specified by the B-field is referred to as the 'base address'. The content of the D-field is referred to as the 'displacement". (This type of address modification is commonly referred to as indexing.) Effective addressing may be either in a positive or a negative direction, and is determined by the sign of the base address. Caution should be exercised because a resultant effective address that is negative or that refers to the first 144 bytes of main storage results in an error condition.

Any of the eight general registers, 8 through 15, may be specified in the B-field as the location of the base address for effective-address generation.

If there are zeros either in the general register specified, or in the displacement field of the instruction when effective-address generation is specified, the effective address generated is the same as direct addressing with the nonzero component.

Effective addressing is useful for program routines which require address modification.

Upper Limit of Main Storage: Data can be stored up to (and including) the last available byte of main storage. However, the user must exercise caution when loading a program into the upper available space of main storage. The last instruction must be located so that at least one byte of main storage remains. (Actually, two bytes of main storage have to remain unused, otherwise the last instruction would not be located on a halfword boundary.) The remaining byte is required to avoid a storage overflow which can occur because, when the instruction is read out, the internal controls automatically update the current

instruction address by an increment equal to the length of this instruction.

OPERANDS

For addressing purposes, operands can be grouped in three classes: explicitly addressed operands in main storage, immediate operands placed in main storage as part of the instruction stream, and operands located in the general registers.

EXPLICITLY ADDRESSED OPERANDS

An explicitly addressed operand is selected from a main-storage location not related to the location of the instruction referring to it. It is always specified by means of a storage address. When the operand contains more than one byte, the address gives the location of the first byte of the field, subsequent bytes being located in higher addresses. Both the first and second operands of an instruction can be explicitly addressed.

Explicitly addressed operands can be of fixed length or variable length. The length of variable-field-length operands is specified in the L-field of the instruction. The L-field, either four or eight bits long, specifies the length in terms of the number of bytes to the right of the addressed byte, and thus can specify a maximum field length of 256 bytes.

IMMEDIATE OPERANDS

An immediate operand consists of one 8-bit byte of data which is located in the instruction itself. Only the instructions in the SI format contain immediate data. The immediate data is always the second operand; the first operand is located in the main-storage location specified by the B1-D1 field.

OPERANDS IN REGISTERS

Information referred to by an instruction may be located in one of eight general registers. The registers are identified by numbers 8-15 and are selected by the four-bit R or B field of an instruction. The registers are not designated by main-storage addresses.

An operand located in a register has a fixed length of one halfword, or 16 bits.

TIME SHARING

The 2922 has the ability to operate in a mode referred to as 'time sharing'. Time sharing is a means of overlapping input/output operations with each other and with processing. Time sharing is based on a system of monitoring the operation of input/output devices and sequencing the transfer of data to or from the I/O devices so as to make the most efficient use of processing time.

Processing operations in the control unit are time shared with the transfer of data between main storage and the I/O devices. When an I/O device requests service, processing is suspended only for the time required to send or to accept the input/output data.

Time sharing allows the control unit to perform useful processing functions while card or forms movement is taking place. A system of signaling that the I/O device is finished with the data transfer to or from the control unit is referred to as an 'interruption' system.

PROGRAM STATUS WORD (PSW)

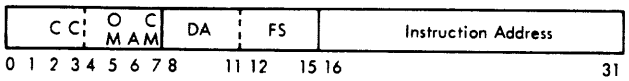
The PSW contains the inforamtion necessary for proper program execution. It is located in an internal register in the control unit and is not directly addressable. The programmer can change the PSW by means of a Set PSW instruction. The PSW has a fixed-length format of two halfwords (Figure 1-4).

The PSW is used to control instruction sequencing and to hold and indicate the status of the system in relation to the program being executed. The active or controlling PSW is called the 'current PSW'.

By storing the current PSW during an interrupt, the status of the control unit can be preserved for subsequent inspection. By loading a new PSW or part of a PSW, the state of the control unit can be

initialized or changed. The functions of the PSW are further described under "Branching".

When the current PSW is stored during an input/output interruption, the status of the control unit and the next sequential address are preserved for use after the interruption is serviced. An instruction to load a new PSW is equivalent to an unconditional branch to the instruction address contained in the new PSW.



0 - 1	Not Used
2 - 3	Condition Code
4	Zero
5	
6	ASCII Mode Bit
7	Channel Mask
8 - 11	Device Address
12 - 15	Function Specification
16 - 31	Instruction Address

Figure 1-4. PSW Format

Operation of the Program Load key or System Reset key causes the condition code (bits 2-3), the overlap mode bit (bit 5), the ASCII mode bit (bit 6), and the channel mask bit (bit 7) of the current PSW to be reset to zero.

INTERRUPTION

In the 2922, an automatic interrupt system is provided to make optimum use of the I/O devices and available processing time. The I/O devices signal the control unit to store an end condition when a data transfer has been terminated. Upon completion of the execution of each instruction, during the time that the control unit is in an interruptible mode, the control unit checks the various end conditions according to a built-in priority sequence. The first end condition thus found causes the actual interrupt. The interrupt is an automatic branch from the main program to a subroutine. This subroutine may be designed to test the received data for validity or to perform some other action. Since the interrupt occurs at the earliest moment possible after a data transfer has been completed, it is the primary means of controlling I/O operations.

The branch to the subroutine is acccmplished by replacing the current PSW with a new PSW which contains the start address of the subroutine in its instruction address portion. The former current PSW is stored into a particular core-storage location and thus becomes the old PSW. During the transfer, the old PSW is furnished with the device address and the function specification of the I/O unit that caused the interrupt.

To initiate a branch back to the main program, the old PSW must be fetched from storage to function as the current PSW again. This is done by addressing the old PSW in the B1-D1 fields of a Set PSW instruction placed at the end of the subroutine. The entire exchange of the program status words is fully automatic; however, the channel mask bit in the PSW, which determines whether the control unit is interruptible or not, may be set or reset under program control. The control unit is interruptible when the channel mask bit is set to 1; it is not interruptible if the channel mask bit is 0. The channel mask bit is zero when the control unit is in a reset state.

ARITHMETIC AND LOGICAL OPERATIONS

The arithmetic and logical operations are employed to process binary numbers of fixed length, decimal numbers of variable length, and logical information of either fixed or variable length.

Arithmetic and logical operations performed by the control unit fall into three classes: binary arithmetic, decimal arithmetic, and logical operations. These three classes differ in the data formats used, the registers involved, the operations provided, and in the way the field length is stated.

BINARY ARITHMETIC

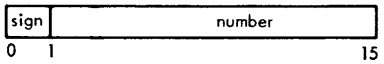
The binary arithmetic operand is the 16-bit binary halfword. The binary instruction perform binary arithmetic on operands serving as addresses, index quantities, and counts, as well as binary data. One operand is always in a general register. The second operand may be either in main storage or in a general register.

Data Format

Binary numbers occupy a fixed length format consisting of a one-bit sign followed by the 15-bit binary field. When held in one of the general registers, a binary quantity has a 15-bit binary field and occupies all 16 bits of the register.

Binary data in main storage also occupies a 16-bit halfword, with a binary field of 15 bits. This data must be located on integral storage boundaries for this unit of information, that is, the low-order binary address bit is zero.

Halfword Binary Number



NUMBER REPRESENTATION

All binary operands are treated as signed numbers. Positive numbers are represented in true binary notation with the sign bit set to 0. Negative numbers are represented in two's-complement notation with a 1 in the sign bit. The two's complement of a number is obtained by inverting each bit of the number and adding a one in the low-order bit position.

The number obtained by inverting each binary bit represents the one's complement of the original number. To represent the two's complement, a one must be added to the low-order position of the one's complement.

```
0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 0 = original number
1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 1 = one's complement
1 0 0 0 1 0 1 0 1 1 0 1 0 0 1 0 = two's complement
|
|
└-------sign position
```

Two's-complement notation does not include a negative zero. It has a number range in which the set of negative numbers is one larger than the set of positive numbers. The maximum positive number consists of an all-one integer field with a sign bit of zero, whereas the maximum negative number consists of an all-zero integer field with a 1 bit for sign.

The control unit cannot represent the complement of the maximum negative number. When an operation, such as a subtraction from zero, produces the complement of the maximum negative number, the number remains unchanged, and a binary overflow condition is recognized. An overflow does not result, however, when the number is complemented and the final result is within the representable range. An example of this case is a subtraction from minus one.

The sign bit is leftmost in a number. An overflow carries into the sign-bit position and changes the sign.

Figure 1-5 illustrates some sample values of 16-bit binary integers and their equivalents in decimal form.

Number		Signed Decimal		S--------- INTEGER ---------
$2^{15}-1$	=	32767	=	0111 1111 1111 1111
2^{8}	=	256	=	0000 0001 0000 0000
2^{0}	=	1	=	0000 0000 0000 0001
0	=	0	=	0000 0000 0000 0000
-2^{0}	=	-1	=	1111 1111 1111 1111
-2^{1}	=	-2	=	1111 1111 1111 1110
-2^{8}	=	-256	=	1111 1111 0000 0000
$-2^{15}+1$	=	-32767	=	1000 0000 0000 0001
-2^{15}	=	-32768	=	1000 0000 0000 0000

Figure 1-5. Binary Representation

CONDITION CODE

The results of binary add, subtract, and compare operations are used to set the condition code in the Program Status Word (PSW). All other binary operations leave this code undisturbed. The condition code can be used for decision-making by subsequent branch-on-condition instructions.

The condition code is set to reflect three types of results for binary arithmetic. For most operations, the status 0(00), 1(01), or 2(10) indicate a zero, less than zero, or greater than zero content of the result register. An overflow results in a Binary Overflow Check condition.

For a compare operation, the states 0, 1, or 2 indicate that the first operand is equal, low, or high compared to the second operand. Condition code settings are shown in Figure 1-6.

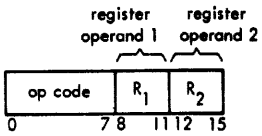
	0	1	2
Compare Halfword	equal	low	high
Add Halfword	zero	<zero	>zero
Subtract Halfword	zero	<zero	>zero
Add	zero	<zero	>zero
Subtract	zero	<zero	>zero

Figure 1-6. Condition Code Settings for Binary Arithmetic

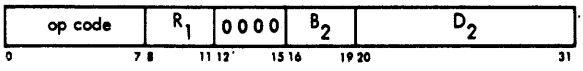
INSTRUCTION FORMAT

Binary instructions use the following two formats:

RR Format



RX Format



In these formats, R1 specifies the address of the general register containing the first operand. The second operand location, if any, is defined differently for each format.

In the RR format, the R2 field specifies the address of the general register containing the second operand.

In the RX format, the address which designates the storage location of the second operand is derived from the contents of the B2 and D2 fields of the instruction. This address may be taken directly from the B2 and D2 fields, or an effective address may be formed by adding the contents of the general register specified by the B2 field to the contents of the D2 field. See "Addressing."

Bits 12 thorugh 15 of the RX format must be zero. The second operand, addressed by B2-D2, must begin on a halfword boundary, that is, its address, must be even, otherwise a program error stop occurs.

Results of operations replace the first operand excpet for a store operation. The result replaces the second operand for the store operation.

An instruction can specify the same general register both for address modification and for operand location. Address modification is always completed before execution of the operation.

The contents of all general registers and storage locations participating in the addressing or execution part of an operation remain unchanged, except for the storing of the final result.

Instructions

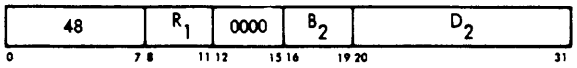
The binary arithmetic instructions and their mnemonics, formats, and operation codes are listed in Figure 1-7.

Name	Mnemonic	Format	Operation Code
Load Halfword	LH	RX	48
Add Halfword	AH	RX	4A
Subtract Halfword	SH	RX	4B
Compare Halfword	CH	RX	49
Store Halfword	STH	RX	40
Add	AR	RR	1A
Subtract	SR	RR	1B

Figure 1-7. Binary Instructions

Load Halfword (RX)

LH R1, D2 (0, B2)

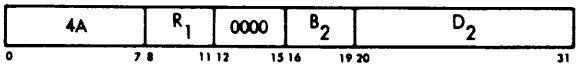


The second operand is placed in the first operand location. The instruction uses RX format and provides for loading a halfword from

the main-storage location specified by the B2-D2 fields into the register specified by the R1 field. The condition code is not changed.

Add Halfword (RX)

AH R1, D2 (0, B2)

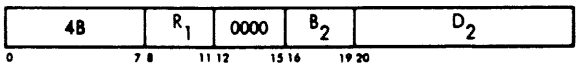


The second operand is added to the first operand and the sum is placed in the first operand location. Both operands are halfword length. The first operand is contained in the register specified by the R1 field. The second operand is located at a main-storage location specified by the B2-D2 fields of the instruction.

Addition is performed by adding all 16 bits of both operands. If the 'carries' out of the sign-bit position and the high-order numberic bit position agree, the sum is satisfactory; if they disagree, an overflow occurs. The sign bit is not changed after the overflow. A positive overflow yields a negative final sum, and a negative overflow results in a positive sum. The overflow causes a binary-overflow-check condition. The condition code is set to 00, 01, or 10 to indicate that the result is zero, less than zero, or greater than zero.

Subtract Halfword (RX)

SH R1, D2 (0, B2)

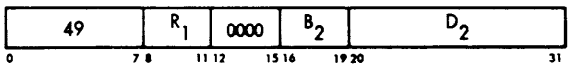


The second operand is subtracted from the first operand and the difference is placed in the first operand location. Both operands are halfword length. The first operand is contained in the register specified by the R1 field. The second operand is located at a main-storage location specified by the B2-D2 fields of the instruction.

Subtraction is performed by adding the two's complement of the second operand to the first operand. All 16 bits of both operands participate, as in ADD. If the 'carries' out of the sign-bit position and the high-order numeric bit position agree, the difference is satisfactory; if they disagree an overflow occurs, resulting in a bianry-overflow-check condition. The condition code is set to 00, 01, or 10 to indicate that the result is zero, less than zero, or greater than zero.

Compare Halfword (RX)

CH R1, D2 (0, B2)



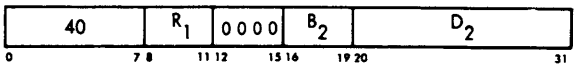
The first operand is compared with the second operand, and the result of the compare is indicated by the setting of the condition code in the PSW. The first operand is contained in the register specified by the R1 field. The second operand is located at a main-storage location specified by the B2-D2 fields of the instruction.

Comparison is algebraic, treating both comparands as 16-bit signed integers. Operands in registers or storage are not changed as a result of the operation.

The condition code is 00 if the operands are equal, 01 if the first operand is lower than the second operand, and 10 if the first operand is higher than the second.

Store Halfword (RX)

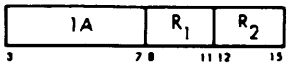
STH R1, D2 (0, B2)



The first operand is stored at the second operand location. The instruction uses RX format and provides for storing a halfword from the register specified by the R1 field into the main-storage location specified by the B2-D2 fields of the instruction. The condition code remains unchanged.

Add (RR)

AR R1, R2



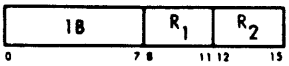
The second operand is added to the first operand, and the sum is placed in the first operand location. The halfword contained in the register

specified by the R1 field is added to the halfword contained in the register specified by the R2 field and the sum replaces the contents of the register specified by R1.

Addition is performed by adding all 16 bits of both operands. If the 'carries' out of the sign-bit position and the high-order numeric bit position agree, the sum is satisfactory; if they disagree, an overflow occurs. The sign bit is not changed after the overflow. A positive overflow yields a negative final sum, and a negative overflow results in a positive sum. The overflow causes a binary-overflow-check condition. The condition code is set to 00, 01, or 10 to indicate that the result is zero, less than zero, or greater than zero.

Subtract (RR)

SR R1,R2



The second operand is subtracted from the first operand, and the difference is placed in the first operand location.

The halfword contained in the register specified by the R2 field is subtracted from the halfword contained in the register specified by the R1 field and the difference replaces the contents of the register specified by R1.

Subtraction is performed by adding the two's complement of the second operand to the first operand. All 16 bits of both operands participate, as in ADD. If the 'carries' out of the sign-bit position and the high-order numeric bit position agree, the difference is satisfactory; if they disagree, an overflow occurs. The overflow causes a binary-overflow-check condition. The condition code is set to 00, 01, or 10 to indicate that the results is zero, less than zero, or greater than zero.

BINARY ARITHMETIC ERROR CONDITIONS

Error conditions which may occur during the instruction or exeuction phase of binary operations are:

1. Operation code invalid.

2. Addressing error.

   a. An insturciton address or an operand address refers to the protected first 144 bytes of main storage (addresses 0 through 143).

   b. An instruction address or an operand address is outside available storage.

   c. In the 2922-1, the last (highest) main-storage position contains any part of an instruction to be executed.

   d. The R1 or R2 fields of a binary instructicn contain values 0 through 7.

3. Specification error.

   a. The low-crder bit of an instruciton address is one.

   b. The halfword seccnd cperand is not located on a 16-bit boundary.

   c. Bits 12 through 15 of an RX fcrmat instuction are not all zero.

4. Binary overflow check.

5. Control Unit parity error.

DFCIMAL ARITEMETIC

Decimal arithmetic operaticns include addition, subtraction, multiplication, division, and comparison. Both operands and results are located in storage.

All decimal arithmetic cperaticns are performed with data in the packed mode for cptimum use cf stcrage. Since data is often communicated to or from peripheral devices in the zoned form, operations are provided for changing frcm zoned to packed format and vice versa.

Operand fields can be located on any byte boundary, except for the first 144 bytes of main stcrage, which are protected. They can have a length of up to 31 digits and sign, except for multiplier and divisor operands which are limited to 15 digits.

Each address specifies the leftmost byte of an operand. Associated with each address is a length field which indicates the number of additional bytes that the operand extends beyond the first byte.

DATA FORMAT

Decimal operands reside in main storage only. They occupy fields that may start at any byte address and are composed of one to 16 eight-bit bytes. All operaions use a two-address format.

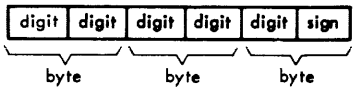
Lengths of the two operands specified in the instruction need not be the same. Regardless of length, the two operands are treated as if the rightmost integers were aligned before the operation begins. For example, in an add operation, the rightmost integers must be aligned to ensure that units are added to units, tens are added to tens, hundreds to hundreds, etc. The programmer may add high-order zeros as needed, to ensure that the result field is always large enough to contain all the digits of the result.

Processing takes place from right to left, as in simple manual arithmetic operations, treating the rightmost position of each operand as the units position, the next position to the left as the tens position, and so on.

Lost 'carries' or lost digits from arithmetic operations are signaled by a decimal-overflow condition. Operands are either in the packed or zoned format. Negative numbers are carried in true form.

Packed Decimal Number

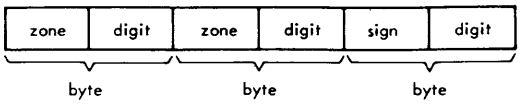
In the packed format, each eight-bit byte can contain the binary equivalents to two decimal digits, except for the rightmost four bits of the rightmost byte of the field, which represent the sign. Decimal digits 0 through 9 are equivalent to binary values 0000 through 1001.



The decimal digits range from 0 to 9 while any value from 10 to 15 (in hexadecimal representation 'A' to 'F') is regarded as a valid sign. In decimal arithmetic, all numbers must be arranged in the packed format. By means of the Pack instruction, it is possible to change zoned decimal numbers into packed numbers. Decimal number fields may overlap only when their rightmost bytes are identical. During the execution of all decimal arithmetic instructions, all data is checked for validity. A sign code found in any other but the sign position leads to an error stop as does a decimal value that is found in the sign position.

Zoned Decimal Numbers

In the zoned format, each eight-bit byte can contain one decimal digit in the right four bits and a zone code in the left four bits. However, the rightmost byte of a zoned decimal number field contains the digit in the right half and the sign in the left half.



The zone bits act only as fill characters and do not affect the value of the decimal digits. The zoned format is needed for decimal data that is sent to character-set sensitive I/O devices. Similarly, all decimal data that is received from card I/O devices comes in the zoned format. The zone is either 15 (F) in the Extended Binary-Coded-Decimal Interchange Code (EBCDIC) or 5 in the American National Standard Code for Information Interchange (ASCII). Since the zone F is identical with one of the general plus signs, all data received from punched cards is recognized as positive unless it is explicitly specified as negative data. Data can be changed from the packed to the zoned format by means of the Unpack instruction. During unpacking, the correct zone is automatically supplied by the control unit. The zone used depends on the mode indicated in bit 6 of the PSW. If the mode bit is zero, the EBCDIC-zone F is used; if the mode bit is one, the ASCII-zone 5 is used.

AUTOMATIC SIGN STANDARDIZATION

During the processing of all arithmetic instructions that handle decimal data, the control unit supplies the final result with the correct sign, according to the rules of algebra. The resultant sign is either a ASCII plus or minus sign, or an EBCDIC plus or minus sign (Figure 1-8). Whether the result sign is standardized in the ASCII mode or in the EBCDIC mode depends on mode bit 6 in the PSW. For example, if the result sign is positive and the mode bit is zero, the sign code 1100 is selected. If the result sign is negative and the mode bit is 1, the sign code 1011 is selected. Thus, the result can have only a ASCII or an EBCDIC sign.

Valid Sign Codes	ASCII	EBCDIC
(A) 1010 +	+	
(B) 1011 -	-	
(C) 1100 +		+
(D) 1101 -		-
(E) 1110 +		
(F) 1111 +		

Figure 1-8. Sign standardization

CONDITION CODE

The results of all add-type and comparison operations are used to set the condition code in the PSW. All other decimal arithmetic operations leave the code unchanged. The condition code can be used for decision-making by subsequent Branch-on-Condition instructions.

The condition code is set to reflect the types of results for decimal arithmetic. Four different result types can occur:

1. The result is zero, indicated by condition code 00.

2. The result is less than zero (negative), indicated by condition code 01.

3. The result is more than zero (positive), indicated by condition code 10.

4. The result does not fit into the result field (overflow) indicated by condition code 11.

For the comparison operation, the condition code settings 0, 1, and 2 indicate that the first operand compared equal, low, or high as shown in Figure 1-9.

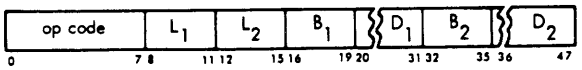
	0	1	2	3
Add Decimal	zero	< zero	> zero	overflow
Compare Decimal	equal	low	high	
Subtract Decimal	zero	< zero	> zero	overflow
Zero and Add	zero	< zero	> zero	

Figure 1-9. Condition Code Settings for Decimal Operations

INSTRUCTION FORMAT

Decimal instruction use the following format:

SS Format



In this format, the address which specifies the core-storage location of the leftmost byte of the first operand field is derived from the contents of the B1 and D1 fields of the instruction. The number of operand bytes to the right of this byte is specified by the L1 field of the instruction. Therefore, the length in bytes of the first operand field is 1 to 16 corresponding to a length code in L1 of 0000 to 1111. The second operand field is specified similarly by the L2, B2, and D2 instruction fields.

As previously described in the section "Addressing", the address of each operand may be taken directly from the respective B and D fields of the instruction, or effective addresses may be formed by adding the contents of the general register specified by the B field to the contents of the D field.

Results of operations replace the first operand field. The result is never stored outside the field specified by the address and length. The second operand field remains unchanged, except in those cases where overlapping fields, which are permitted, actually occur. The contents of the general registers remain unchanged.

INSTRUCTIONS

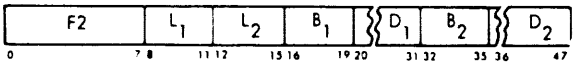
The decimal arithmetic instructions and their mnemonics and operation codes are shown in Figure 1-10.

Name	Mnemonic	Format	Operation Code
Pack	PACK	SS	F2
Unpack	UNPK	SS	F3
Move with Offset	MVO	SS	F1
Zero and Add	ZAP	SS	F8
Add Decimal	AP	SS	FA
Subtract Decimal	SP	SS	FB
Compare Decimal	CP	SS	F9
Multiply Decimal	MP	SS	FC
Divide Decimal	DP	SS	FD

Figure 1-10. Decimal Instructions

Pack (SS)

PACK D1 (L1, B1), D2 (L2, B2)



The format of the second operand is changed from zoned to packed, and the result is placed in the first operand location.

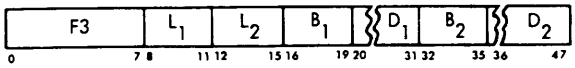
The second operand is assumed to have the zoned format. All zones are ignored, except the zone over the low-order digit, which is assumed to represent a sign. The sign is placed in the right four bits of the low-order byte, and the digits are placed adjacent to the sign and to each other in the remainder of the result field. The sign and digits are moved unchanged to the first operand field and are not checked for valid codes.

The fields are processed right to left. If the second operand does not fill the first operand completely, the remaining high-order positions of the first operand are filled with zeros. If the first operand field is too short to contain all the significant digits of the second operand field, the remaining digits are ignored. Overlapping fields may occur and are processed by storing each result byte immediately after the necessary operand bytes are fetched.

The condition code remains unchanged.

Unpack (SS)

UNPK D1 (L1, B1), D2 (L2, B2)



The format of the second operand is changed from packed to zoned, and the result is palced in the first operand location.

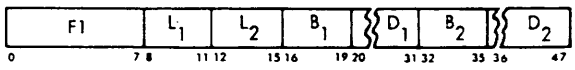
The digits and sign of the packed operand are placed unchanged in the first operand location, using the zoned format. Zones with coding 1111 in EBCDIC mode and coding 0101 in the ASCII mode are supplied for all bytes except the rightmost byte, which receives (in its high-order four bits) the sign of the packed operand. The operand sign and digits are not checked for valid codes.

The fields are processed right to left. If the second operand does not fill the first operand completely, the remaining high-order bytes of the first operand are each filled with a zero and a zone. If the first operand field is too short to contain all significant digits of the second operand, the remaining digits are ignored. The first and second operand fields may overlap and are processed by storing a result byte immediately after the necessary operand bytes are fetched. Caution must be exercised when overlapping the operands, otherwise bytes which have already been unpacked may be unpacked a second time during a single operation, thus giving an incorrect result.

The condition code remains unchanged.

Move with Offset (SS)

MVO D1 (L1, B1), D2 (L2, B2)



The second operand is placed to the left of and adjacent to the low-order four bits of the first operand.

The low-order four bits of the first operand are attached as low-order bits to the second operand, the second operand bits are offset by four bit positions, and the result is placed in the first operand location. Thus, the second operand is effectively offset by half a

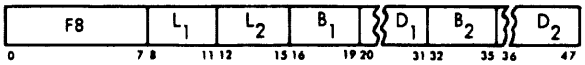
byte and, after the operand, the two halves of each original byte are located in two adjacent bytes. The first and second operand bytes are not checked for valid codes.

The fields are processed right to left. If necessary, the second operand is extended with high-order zeros. If the first operand field is too short to contain all bytes of the second operand, the remaining information is ignored. Overlapping fields may occur and are processed by storing a result byte as soon as the necessary operand bytes are fetched.

The condition codes remains unchagned.

Zero and Add (SS)

ZAP D1 (L1, B1), D2 (L2, B2)



The second operand is placed in the first operand location.

The operation is equivalent to an addition to zero. The sign code is made 1100 for positive and 1101 for negative results in EBCDIC and 1010 for positive and 1011 for negative results in the ASCII mode. A zero result is always positive.

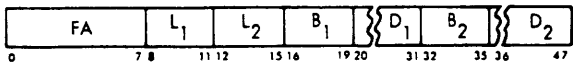
When the length of the second operand (L2) is greater than the length of the first operand (L1) a specification error stop occurs. The instruction is not executed.

The second operand is checked for valid sign and digit codes. Extra high-order zeros are supplied if needed. The first and second operand fields may overlap when the rightmost byte of the first operand field is coincident with or to the right of the rightmost byte of the second operand.

The condition code is set to 00, 01, or 10 to indicate that the result is zero, less than zero, or greater than zero.

Add Decimal (SS)

AP *D1 (L1, B1), D2 (L2, B2)*



The second operand is added tc the first operand, and the sum is placed in the first operand lccatien.

When the length of the seccnd operand (L2) is greater than the length of the first operand (L1), a specification-error stop occurs. The instruction is not executed.

Addition is algebraic, taking into account the sign and all digits of both operands. All signs and digits are checked for validity. If necessary, high-order zeros are supplied for the seccnd operand.

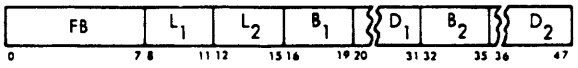
The first and second operand fields may overlap when their low-order bytes coincide; therefcre, it is possible to add a number to itself.

The sign of the result is determined by the rules cf algebra. A zero sum is always positive. When high-order digits are lost because of an overflow, a zero result has the sign cf the correct sum.

The condition code is set tc CC, C1, 10 or 11 to indicate that the result is zero, less than zerc, greater than zero, or overflow. A decimal overflow is not considered to be an error; that is, no machine stop occurs.

Subtract Decimal (SS)

SP *D1 (L1, B1), D2 (L2, B2)*



The second operand is subtracted frcm the first operand, and the difference is placed in the first operand location.

When the length of the second operand (L2) is greater than the length of the first operand (L1) a specification error stop occurs. The instruction is not executed.

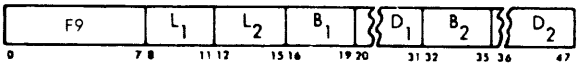
Subtraction is algebraic, taking into account the sign and all digits of both operands. The Subtract Decimal instruction is similar to the Add Decimal instruction, except that the sign of the second operand is changed from positive to negative or from negative to positive after the operand is obtained from storage and before the arithmetic operation.

The sign of the result is determined by the rules of algebra. A zero difference is always positive. When high-order digits are lost because of overflow, a zero result has the sign of the correct difference.

The operands of a Subtract instruction may overlap when their low-order bytes coincide even when their lengths are unequal. This property may be used to make an entire field or the low-order part of a field zero. The condition code is set to 00, 01, 10, or 11 to indicate that the result is zero, less than zero, greater than zero, or an overflow condition. A decimal overflow is not considered to be an error; that is, no machine stop occurs.

Compare Decimal (SS)

CP D1 (L1, B1), D2 (L2, B2)



The first operand is compared with the second, and the condition code indicates the comparison result.

When the length of the second operand (L2) is greater than the length of the first operand (L1), a specification-error stop occurs. The instruction is not executed.

Comparison is right to left, taking into account the sign and all digits of both operands. All signs and digits are checked for validity. If the second operand field is shorter in length than the first operand field, the second operand field is extended with high-order zeros. A positive zero compares equal to a negative zero. Neither operand is changed as a result of the operation. Overflow cannot occur in this operation.

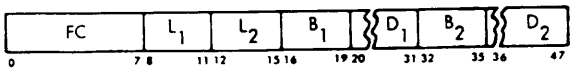
The first and second fields may overlap when their low-order bytes coincide. It is possible, therefore, to compare a number to itself.

A Decimal Compare instruction differs in several respects from a Logical Compare instruction. Signs, zeros, and invalid characters are considered and fields are extended when unequal in length. The

condition code is made 00 if the operands are equal, 01 if the first operand is low, and 10 if the first operand is high.

Multiply Decimal (SS)

MP D1 (L1, B1), D2 (L2, B2)



The product of factor one (the first operand) times factor two (the second operand) replaces the first operand. A multiplication can be performed only on data in the packed format. The length codes contained in the L1 and L2 fields specify the number of bytes that extend to the left of the units byte. The units byte is the rightmost byte of a packed decimal operand and it contains one digit and the sign. Factor two is limited to a size of 15 digits plus the sign, or in other words, the length code of factor two must not be greater than 7. Furthermore, the length code of factor two must always be smaller than the length code of factor 1. If L2 is greater or equal to L1, or if L2 exceeds 7, a program error stop occurs.

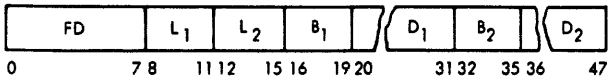
Factor one is limited to a size of 16 bytes (length code 15). The main storage location of factor one is specified by the B1-D1 field according to the rules for direct or effective-address generation. Similarly, the location of factor two is specified by the B2-D2 field.

Since the number of digits in the product (the result) is the sum of the number of digits in both operands, factor one must have as many leading (high-order) zero bytes as factor two has data bytes (L2+1). This requirement for the factor one field ensures that no product overflow can occur. If factor one has insufficient high-order zeros, a program error stop occurs. The sign of the product is developed from the signs of both operands according to the rules of algebra. This is true also when one or both operands are zero. If the larger of the two factors is 15 digits or less, the most efficient use of this instruction is realized when the larger factor is used in the second operand.

The factor two field may overlap the product field when the low-order bytes of both fields coincide. The condition code is not changed by a Multiply Decimal instruction.

Divide Decimal (SS)

DP D1 (L1, B1), D2 (L2, B2)



The dividend (the first operand) is divided by the divisor (the second operand) and is replaced by the quotient and remainder. A divide operation must be performed on data in the packed format only. Length codes, contained in the L1 and I2 fields, specify the number of bytes to the left of the units byte. Each byte contains two arithmetic digits, except the units byte which contains one digit and a sign. The location of the dividend is specified by the B1-D2 field, and the location of the divisor is specified by the B2-D2 field. The result consists of the quotient and the remainder.

The quotient field is placed leftmost in the first operand field. The remainder field is placed rightmost in the first operand field and has a size equal to the divisor size. Together, the quotient and remainder occupy the entire former dividend field; therefore, the address of the quotient field is the address of the first operand. The size of the quotient field in eight-bit bytes is L1-L2, and the length code for this field is one less (L1-L2-1). When the divisor length code is larger than seven (15 digits and sign) or larger than or equal to the dividend length code, the operation is not executed and an error stop occurs.

The dividend, divisor, quotient, and remainder are all signed integers, right-aligned in their fields. The sign of the quotient is determined by the rules of algebra from dividend and divisor signs. The sign of the remainder is the same as the dividend sign. These rules are true even when quotient or remainder is zero.

A divide check occurs when the quotient is larger than the number of digits allowed for it, or if the dividend does not have at least one leading zero. If a divide check occurs, the operation is not executed and a decimal-divide-error stop occurs. Divisor and dividend remain unchanged in their storage locations.

The divisor and dividend fields may overlap only if their low-order bytes coincide.

The condition code remains unchanged for division, and overflow cannot occur.

Programming Notes

The maximum dividend size is 30 digits and sign (16 packed bytes). Since the smallest remainder size is one digit and sign, the maximum quotient size is 29 digits and sign.

The condition for a divide check can be determined by a trial subtraction. The leftmost digit of the divisor field is aligned with the leftmost-plus-one digit of the dividend field. When the divisor, so aligned, is less than or equal to the dividend, a quotient overflow is indicated.

By programming a Divide Decimal instruction so that the dividend field (the first operand) has a number of high-order zero digits equal to the field size of the divisor field (the second operand), a divide check occurs only when a division by zero is attempted.

DECIMAL ARITHMETIC ERROR CONDITIONS

The following error conditions may occur during the instruction or execution phases of decimal arithmetic operations:

1. Operation code invalid.

2. Addressing error.

   a. An instruction address or an operand address refers to the protected first 144 bytes of main storage.

   b. An instruction address or an operand address is outside available storage.

   c. An instruction occupies the last two (highest) main storage positions.

3. Specification error.

   a. The low-order bit of an instruction address is 1.

   b. For Zero and Add, Compare Decimal, Add Decimal, and Subtract Decimal instructions, the length code L2 is greater than the length code L1.

   c. For Multiply Decimal and Divide Decimal instructions, the length code L2 is greater than 7 or greater than or equal to the length code L1.

4. Data error.

   a. A sign or digit code of an operand in the Zero and Add, Compare Decimal, Add Decimal, Subtract Decimal, Multiply Decimal, or Divide Decimal instruction is incorrect, or the operand fields in these instructions overlap incorrectly.

   b. The factor one field (first operand) in a Multiply Decimal instruction has insufficient high-order zeros.

5. Decimal divide check. The resultant quotient in a Divide Decimal instruction exceeds the specified data field instruction (including division by zero) or the dividend has no leading zero.

6. Control Unit parity error.

LOGICAL OPERATIONS

A set of operations is provided for the logical manipulation of data. Generally the operands are treated as characters, eight bits at a time. In a few cases the left or right four bits of a byte are treated separately. The operands are either in storage or are introduced from the instruction stream.

Processing of data in storage proceeds left to right through fields which may start at any byte position.

Except for the Edit instruction, data is not treated as numbers in this section. Editing provides a transformation from packed decimal digits to alphameric characters.

The set of logical operations includes move, comparison, editing, testing, and bit-connective operations.

The condition code is set as a result of all logical connective, comparison, editing, and testing operations.

DATA FORMAT

Data resides in storage, or is introduced from the instruction itself. The data size may be a single character, or variable in length. When two operands participate they have equal length, except in the Edit instruction.

In storage-to-storage operations, data has a variable field-length format, starting at any byte address (except for the first 144 storage-protected bytes) and continuing to a maximum of 256 bytes. Processing is left to right.

Operations which introduce data from the instruction into storage are restricted to an eight-bit byte. Only one byte is introduced from the instruction, and only one byte in storage participates.

Editing requires a packed decimal field; in other operations no internal data structure is required and all bit configurations are considered valid.

In storage-to-storage operations, the operand fields may be defined in such a way that they overlap. The effect of this overlap depends upon the operation. When the operands remain unchanged, as in compare, overlapping does not affect the execution of the operation. In the case of move and edit operations, one operand is replaced by new data, and the execution of the operation may be affected by the amount of overlap and the manner in which data is fetched or stored. For purposes of evalauting the effect of overlapped operands, it can be considered that data is always handled one eight-bit byte at a time. All overlapping fields are considered valid, but in the edit operations overlapping fields give unpredictable results.

CONDITION CODE

The results of most logical operations are used to set the condition code. The move operations leave this code undisturbed. The condition code can be used for deicision-making by subsequent branch-on-condition instructions.

There are four types of condition code settings for logical instructions. For the Edit instruction the codes 00, 01, and 10 indicate a zero, less than zero, and greater than zero content of the last result field.

For the logical connective operations, the codes 00 and 01 indicate a zero or nonzero result field.

For the Compare Logical instrucitcn, the codes 00, 01, and 10 indicate that the first operand compared equal, low or high.

Figure 1-11 shows the condition code for logical operations.

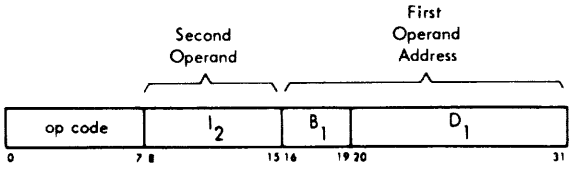
	00	01	10	11
Test under Mask	zero	mixed	- -	one
AND	zero	not zero	- -	- -
Compare Logical	equal	low	high	- -
OR	zero	not zero	- -	- -
Edit	zero	< zero	> zero	- -

Figure 1-11. Condition Code Settings for Logical Operations

INSTRUCTION FORMAT

Logical instructions use the SI or SS formats.

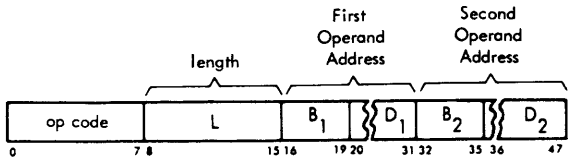
SI FORMAT



In the SI format, the address which specifies the core-storage location of the first operand field is derived from the contents of the B1-D1 fields of the instruction. The address may be taken directly from the B1-D1 fields (direct addressing) or an effective address may be formed by adding the contents of the general register specified by the B1 field to the contents of the D1 field (effective addressing).

Results replace the first operand. The contents of the general registers are not changed.

SS Format



The address which specifies the core-storage location of the first operand field is derived from the B1-D1 fields of the instruction. The B2-D2 fields specify the leftmost byte of the second operand field. The first and second operand fields are the same length, except in the Edit instruciton. The number of bytes extending to the right of the first byte is specified by the I field of the instruction.

The address of each operand may be derived by either direct or effective addressing. The result of the operation replaces the first operand, and is never stored outside the field specified by the address and length fields of the instruction. The contents of the general registers remain unchanged.

INSTRUCTIONS

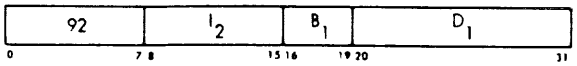
The logical instructions, their mnemonics, formats, and operation codes are shown in Figure 1-12.

Name	Mnemonic	Format	Operation Code
Move	MVI	SI	92
Move	MVC	SS	D2
Move Numerics	MVN	SS	D1
Move Zones	MVZ	SS	D3
Compare Logical	CLI	SI	95
Compare Logical	CLC	SS	D5
Edit	ED	SS	DE
AND	NI	SI	94
OR	OI	SI	96
Test under Mask	TM	SI	91
Halt and Proceed	HPR	SI	99
Translate	TR	SS	DC

Figure 1-12. Logical Instructions

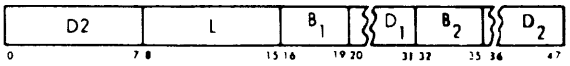
Move (SI)

MVI D1 (B1), I2



Move (SS)

MVC D1 (L, B1), D2 (B2)



The second operand is placed in the first operand location.

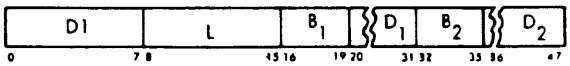
The SS format is used for a storage-to-storage move. The SI format introduces one eight-bit byte from the instruction stream.

In storage-to-storage movement, the fields may overlap in any desired way. Movement is left to right through each field, a byte at a time.

The bytes to be moved are not changed or inspected. The condition code remains unchanged. It is possible to propagate one character through an entire field by having the first operand field start one character to the right of the second operand field.

Move Numerics (SS)

MVN D1 (L, B1), D2 (B2)



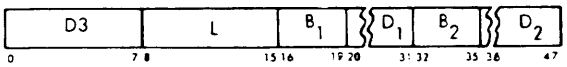
The low-order four bits of each byte in the second operand field, the numerics, are placed in the low-order bit positions of the corresponding bytes in the first operand field.

The instruction is storage-to-storage. Movement is left to right through each field, one byte at a time, and the fields may overlap in any desired way.

The numerics are not changed or checked for validity. The high-order four bits of each byte, the zones, remain unchanged in both operand fields. The condition code remains unchanged.

Move Zones (SS)

MVZ D1 (L, B1), D2 (B2)



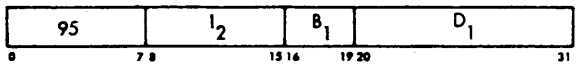
The high-order four bits of each byte in the second operand field, the zones, are placed in the high-order four bit positions of the corresponding bytes in the first operand field.

The instruction is storage-to-storage. Movement is left to right through each field one byte at a time, and the fields may overlap in any desired way.

The zones are not changed or checked for validity. The low-order four bits of each byte, the numerics, remain unchanged in both operand fields. The condition code remains unchanged.

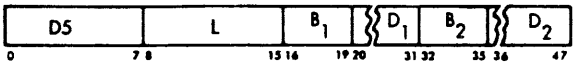
Compare Logical (SI)

CLI D1 (B1), I2



Compare Logical (SS)

CLC D1 (L, B1), D2 (B2)



The first operand is compared with the second operand, and the result is indicated in the condition code.

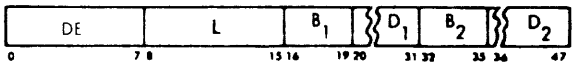
The instructions allow comparisons that are instruction-to-storage, and storage-to-storage.

Comparison is binary, and all codes are valid. The operation proceeds left to right and terminates as soon as an inequality is found. The condition code is made 00 if the operands are equal, 01 if the first operand is low compared to the second operand, and 10 if the first operand compares high.

In the Compare Logical instruction, all bits are treated alike as part of an unsigned binary quantity. In the variable length storage-to-storage operation, comparison is left to right and may extend to field lengths of 256 bytes. This instruction may be used for alphameric comparison.

Edit (SS)

ED D1 (L, B1), D2 (B2)



The format of the source (the second operand) is changed from packed to zoned and is edited under control of the pattern (the first operand). The edited result replaces the pattern.

Editing includes sign and punctuation control and the suppression and protection of leading zeros. It also facilitates programmed blanking of all-zero fields. Several numbers may be edited in one operation, and numeric information may be combined with text.

The length field applies to the pattern (the first operand). The pattern has the unpacked format and may contain any character. The source (the second operand) has the packed format and must contain

valid decimal digit and sign codes. The left four bits of a byte must be 0000-1001. The right four bits are recognized as either a sign or a digit.

Both operands are processed left to right, one character at a time. Overlapping pattern and source fields give unpredictable results.

The character to be stored in the first operand field is determined by three things; the digit obtained from the source field, the pattern character, and the state of a trigger, called the S trigger. One of three actions may be taken:

1. The source digit is expanded to zoned format and is stored into the first operand.

2. The pattern character is left unchanged.

3. A fill character is stored into the first operand.

S Trigger: The S trigger is used to control the storage or replacement of source digits and pattern characters. Digits to be stored in the result, whether zero or not, are termed significant. Pattern characters are replaced or stored when they are significance-dependent (such as punctuations) or sign-dependent (such as credit symbols). The S trigger is also used to record the sign of the source and it sets the condition code accordingly.

The S trigger is set to the 0 state at the start of the operation and is subsequently changed depending upon the source number and the pattern characters.

Pattern Character: Three pattern characters have a special use in editing. They are the digit-select character, the significance-start character, and the field-separation character. These three characters are replaced, either by a source digit or by a fill character; their encoding is shown in Figure 1-13.

The functions of these characters are as follows:

1. The digit-select character causes either a source digit or the fill character to be inserted in the result field.

2. The significance-start character has the same function but also indicates, by setting the S trigger, that the following digits are signifcant.

3. The field-separation character identifies individual fields in a multiple-field editing operation. The character is replaced by the fill character. The S trigger is set to zero, and testing for a zero-field is reinitiated.

4. All other pattern characters are treated in a common way; if the S trigger is 1, the pattern character is left unchanged; if the S trigger is 0, the pattern character is replaced by the fill character.

If the pattern character is either a digit-select or a significance-start character, the source digit is examined. The source digit replaces the pattern character if the S trigger is 1 or if the source digit is nonzero. If a nonzero digit is inserted when the S trigger is 0, the S trigger is set to 1 to indicate that the subsequent digits are significant. If the S trigger and the source digit are both 0, the fill character is substituted for the pattern character.

Source Digit: When the source digit is stored in the result, its code is expanded from the packed to the zoned format by attaching a zone. The zone code is 1111 in the binary-coded-decimal mode and 0101 in the ASCII mode. The type of zone used depends on the mode bit 6 of the PSW. For example, if the mode bit is 0, 1111 (the EBCDIC zone) is used.

The source digits are examined only once during an editing operation. They are slected eight bits at a time from the second operand field. The leftmost four bits are examined first, and the rightmost four bits remain available for the next pattern character which calls for a digit examination. However, the rightmost four bits are inspected for a sign code immediately after the leftmost four bits are examined.

Any of the plus-sign codes 1010, 1100, 1110 or 1111 sets the S trigger to 0 after the digit is inspected, whereas the minus-sign codes 1011 and 1101 leave the S trigger unchanged. When one of these sign codes is encountered in the four rightmost bits, these bits are not longer treated as a digit, and a new character is fetched from storage for the next digit to be examined.

A plus sign sets the S trigger to 0 even if the trigger was set to 1 for a nonzero digit in the same source byte or by a significance-start character for that digit.

Fill Character: The fill character is obtained from the pattern as part of the editing operation. The first character of the pattern is used as a fill character and is left unchanged in the result field, except when it is the digit-select or significance-start character. In the latter cases, a digit is examined and, when nonzero, inserted.

Result Condition: To facilitate the blanking of all-zero fields, the condition code is used to indicate the sign and zero status of the last field edited. All digits examined are tested for the code 0000. The presence or absence of an all-zero source field is recorded in the condition code at the termination of the editing operation.

1. The condition code is made 0 for a zero source field, regardless of the state of the S trigger.

2. For a nonzero source field and an S trigger of 1, the code is made 1 to indicate less than zero.

3. For a nonzero source field and an S trigger of 0, the code is made 2 to indicate greater than zero.

The condition-code setting pertains to fields as specified by the field-separator characters, regardless of the number of signs encountered.

For the multiple-field editing operations, the condition-code setting reflects only the field following the last field-separator character. When the last character of the pattern is a field-separator character, the condition code is made 0.

Figure 1-13 gives the details of the edit operation. The leftmost columns give the pattern character and its code. The next columns show the states of the digit and the S trigger used to determine the resulting action. The rightmost column shows the new setting of the S trigger.

The example which follows shows the step-by-step editing of a packed field with a length specification of four against a pattern 13 bytes long. The following symbols are used:

Symbol	Meaning
b	blank character
((hexadecimal 21)	significance-start character
) (hexadecimal 22)	field-separation character
d (hexadecimal 20)	digit-select character

Assume:

```
Loc 1000-1012 (first operand)          bdd, dd(.ddbCR
Loc 1200-1203 (second operand)       02 57 42 6+
Reg 12 (decimal equivalent)                1000
```

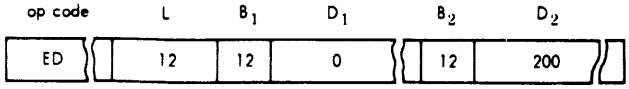
Character Code	Name and Purpose	Examine Digit	Trigger Status	Digit Status	Result Character	Trigger Set
0010 0000	digit select	yes	s = 1		digit	
			s = 0	d not 0	digit	s = 1
			s = 0	d = 0	fill	
0010 0001	significance start	yes	s = 1		digit	
			s = 0	d not 0	digit	s = 1
			s = 0	d = 0	fill	s = 1
0010 0010	field separator	no			fill	s = 0
other	message insertion	no	s = 1		leave	
			s = 0		fill	

Notes:

d	Source digit
s	S trigger (1: minus sign, digits, or pattern used; 0: plus sign, fill used)
digit	A source digit replaces the pattern character
fill	The fill character replaces the pattern character
leave	The pattern character remains unchanged

Figure 1-13. Edit Characteristics

The instruction is:



and it provides the following:

Pattern	Digit	S Trigger	Rule	Location 1000-1012
b		0	leave (1)	bdd, dd (.ddbCR
d	0	0	fill	bbd, dd (.ddbCR
d	2	1	digit	bb2, dd(.ddbCR(2)
,		1	leave	same
d	5	1	digit	bb2, 5d(.ddbCR
d	7	1	digit	bb2, 57(.ddbCR
(	4	1	digit	bb2, 574.ddbCR
.		1	leave	same
d	2	1	digit	bb2, 574,2dbCR
d	6+	0	digit	bb2, 574.26bCR(3)
b		0	fill	same
C		0	fill	bb2, 574.26bbR
R		0	fill	bb2,574.26bbb

Notes:

1. This character is saved as the fill character.
2. First nonzero digit sets S trigger to 1.
3. Plus sign in this same byte sets S trigger to zero.

Condition code = 2; result greater than zero.

Thus:
Loc 1000-1012 (after) bb2, 574.26bbb

If the second operand in location 1200-1203 is 00 00 02 6-, the following results are obtained:

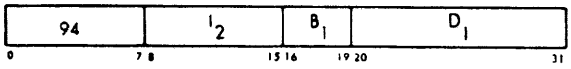
Loc 1000-1012 (before) bdd,dd(.ddbCR
Loc 1000-1012 (after) bbbbb.26bCR

Condition code=1; result less than zero

In this case the significance-start character in the pattern causes the decimal point to be left unchanged. The minus sign does not reset the S trigger so that the CR symbol is also preserved.

AND (SI)

NI *D1 (B1), I2*



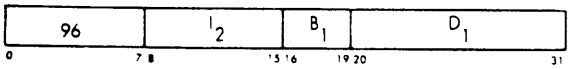
This instruction uses the SI format, which provides a single character instruction-to-storage operation. This instruction may be used to set a bit to 0.

The logical product (AND) of the bits of the first and second operands is placed in the first operand location.

Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. All operands and results are valid. The condition code is set to zero (00) or not zero (01) according to the result of the operation (See Figure 1-11).

OR (SI)

OI *D1 (B1), I2*



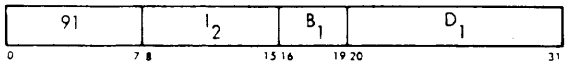
This instruction uses the SI format, which provides a single character instruciton-to-storage operation. This insturction may be used to set a bit to 1.

The logical sum (OR) of the bits of the first and second operand is placed in the first operand location. The condition code is set to zero (00) or not zero (01) according to the result of the operation (See Figure 1-11).

Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by bit. All operands and results are valid.

Test Under Mask (SI)

TM D1 (B1), I2



This instruction inspects the first operand bits and compares them with the I2 field, which is the mask. The mask determines which operand bits are to be compared, as follows: each zero bit in the mask excludes the corresponding operand bit from the comparison. Thus only the 1-bits in the mask are compared with the corresponding operand bits. The comparison leads to one of three different results, neither the first operand nor the mask being changed:

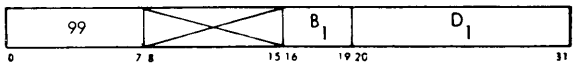
1. If each 1-bit in the mask is matched by a corresponding 1-bit in the first operand, the condition code is set to 11.

2. If only some of the 1-bits in the mask are matched by 1-bits in the first operand, the condition code is set to 01.

3. If none of the 1-bits in the mask are matched, or if the mask is zero, the condition code is set to 00.

Halt and Proceed (SI)

HPR D1 (B1), I2



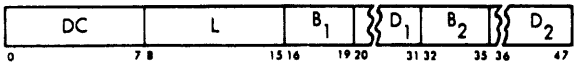
The Halt and Proceed instruction is used to stop the control unit. All input/output operations in progress are continued to completion.

When the control unit has been stopped by the Halt and Proceed instruction, the program may be resumed with the next sequential instruction by operating the Start key on the control unit console.

This instruction uses the SI format in which the I2 field is ignored. The direct or effective address derived from the B1-D1 fields is displayed in the E-S-T-R registers on the console to identify the Halt and Proceed instruction. The condition code remains unchanged.

Translate (SS)

TR D1 (L, B1), D2 (B2)



The first operand contains the data that is to be translated. The second operand represents the translating list. The first operand is selected byte by byte proceeeding from left to right. The binary value of each oprand 1 byte (the argument) is added to the second operand (left byte) address. The resultant new address is used to select an appropriate byte from the translating list (second operand) which contains the function bytes. The selected function byte replaces the original argument byte in the first operand.

All data is valid. The operation proceeds until the first operand field is exhaused. It is permissible for the list and the first operand field to overlap. The condition code remains unchanged.

ERROR CONDITIONS

Error conditions which may occur during the instruction or execution phase of logical operations are:

1. Operation code invalid.
2. Addressing error.
   a. An instruction address or an operand address refers to the protected first 144 bytes of main storage (addresses 0 to 143).
   b. An instruction address or an operand address is outside available storage.
   c. The last (highest) main-storage position contains any part of an instruction to be executed.
3. Specification error. The low-order bit of an instruction address is 1.
4. Data error. An invalid digit code is contained within the second operand field of an Edit operation.
5. Control Unit parity error.

BRANCHING

Instructions are performed by the control unit primarily in the sequential order of their location. A departure from this normal sequential operation may occur when branching is performed. The branching instructions provide a means for making a two-way choice, to reference a subroutine, or to repeat a segment of coding, such as a loop.

Branching is performed by introducing a branch address as a new instruction address.

The branch address may be obtained from one of the general registers or it may be the address specified by the instruction. The branch address is independent of the updated instruction address. Branching may be conditional or unconditional. Unconditional branches always replace the updated instruction address with the branch address. Conditional branches may use the branch address or may leave the updated instruction address unchanged. When branching takes place, the instruction is called successful; otherwise, it is called unsuccessful.

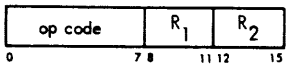
Whether a conditional branch is successful depends on the result of operations preceding the branch. An example is the Branch on Condition instruciton, which inspects the condition code that reflects the result of a previous arithmetic or logical operation.

During a branching operation, the rightmost half of the PSW, the updated instruction address, may be stored before the instruction address is replaced by the branch address. The stored information may be used to link the new instruction sequence with the preceding sequence.

Instruction Format

Branching instructions employ the RR, RX, and the SI formats.

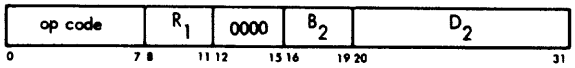
RR Format



The R1 field may specify a general register into which the updated instruction address is to be stored as link information, or may contain a mask which is employed to identify the bit values of the condition code.

The R2 field specifies the general register which contains the branch address.

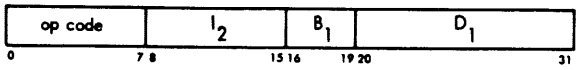
RX Format



The R1 field may specify a general register into which the updated instruction address is to be stored as link information, or may contain a mask which is employed to identify the bit values of the condition code.

The direct or effective address derived from the B2-D2 fields in the branch address.

SI Format



The SI format is employed by only one branching instruciton, Set PSW. The direct or effective address derived from the B1-D1 fields specifies the location of a word in main storage which is to replace the rogram Status Word (PSW); the contents of the I2 field are ignored.

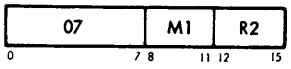
Instructions

The branch instructions, their operation codes, formats, and mnemonics are shown in Figure 1-14.

Name	Mnemonic	Format	Operation Code
Branch on Condition	BCR	RR	07
Branch on Condition	BC	RX	47
Branch and Store	BASR	RR	0D
Branch and Store	BAS	RX	4D
Set PSW	SPSW	SI	81

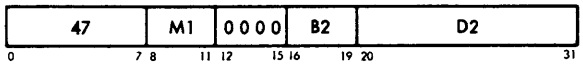
Figure 1-14. Branch Instructions

Branch on Condition (RR)



Branch on Condition (RX)

BC M1, D2 (0, B2)



The updated instruction address is replaced by the branch address if the state of the condition code is as specified by M1; otherwise, normal instruction sequencing proceeds with the updated instruction address.

The M1 field is used as a 4-bit mask. The four bits of the mask correspond, left to right, with the four condition codes, 0, 1, 2, and 3, as shown in the following table. The branch is successful whenever the condition code has a corresponding mask bit of one. The condition code is not changed.

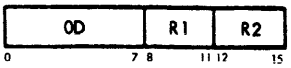
Condition Code	Instruction Bits
0 (0 0)	8
1 (0 1)	9
2 (1 0)	10
3 (1 1)	11

When all four mask bits are ones, the branch is unconditional. When two mask bits are ones (for example, bits 8 and 9) the branch will occur when the condition code is either 0 or 1. When all four

mask bits are zero or when the R2 field in the RR format contains zero, the branch instruction is equivalent to a no-operation. Refer to Appendix G for a table of all operations which affect the condition codes.

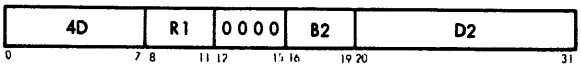
Branch and Store (RR)

BASR R1,R2



Branch and Store (RX)

BAS R1,D2(0,B2)



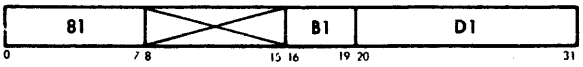
The rightmost 16 bits of the PSW, the updated instruction address, are stored as link information in the general register specified by R1. Simultaneously the instruction address is replaced by the branch address.

The condition code remains unchanged.

When in the RR format, the R2 field contains zero, the link infomration is stored without branching.

Set PSW (SI)

SPSW D1(B1)



The 32-bit word (four 8-bit bytes), located in main storage with the leftmost byte at the first operand address, replaces the Program Status Word.

Bits 8-15 in the Set PSW instruction are ignored.

The PSW has a fixed length format of one word (see Figure 4). It is lcoated in an unaddressable register in the control unit and is employed as an internal control. Since the PSW contains the address

of the next sequential instruction, the Set PSW instruction is equivalent to a branch operation.

Programming Notes:

1. The instruction address portion of the word which is transferred from main storage to the PSW by the Set PSW instruction should:

   a. Not refer to the protected first 144 bytes of main storage.

   b. Have the least significant bit zero, and

   c. Be within the limits of available storage.

   If these conditions are not satisfied, an addressing or specification error stop will occur.

2. The condition code is set by the Set PSW instruction to the value contained in the word transferred from main storage to the PSW.

3. Main-storage boundaries are not required of the first operand address in the Set PSW instruction.

4. The condition code, ASCII mode bit, channel mask and overlap mode bit in the PSW are zero when the control unit is in the reset state. The instruciton address protion of the PSW is not changed when the control unit is reset.

Error Conditions

Error conditions which may occur during a branching operation are:

1. Operation code invalid.

2. Addressing error.

   a. An instruction address or a branch address either refers to the protected first 144 bytes of main storage, or is outside available storage. If an incorrect branch address has been specified, the control unit indicates the error during the execution of the branch instruction.

      Note: When the branching condition is not met and the branch address is invalid, either an addressing error indication or a no-operation may result.

b. The R1 field of a Branch and Store instruction contains binary values 0 through 7, or the R2 field of an RR format branch instruction contains binary values 1 through 7.

c. An instruction part is located in the highest two main-storage positions.

3. Specification error.

   a. The low-order bit of an instruction address is 1.

   b. Bits 12 through 15 of an RX format instruction are not all 0.

4. Control Unit parity error.

TERMINAL CONTROL UNIT CONSOLE

The console (Figure 1-15) provides the switches, keys, and lights necessary to operate and control the terminal.

OPERATING KEYS AND INDICATORS

Register Display Indicators

The eight primary data registers in the control unit are displayed on the console.

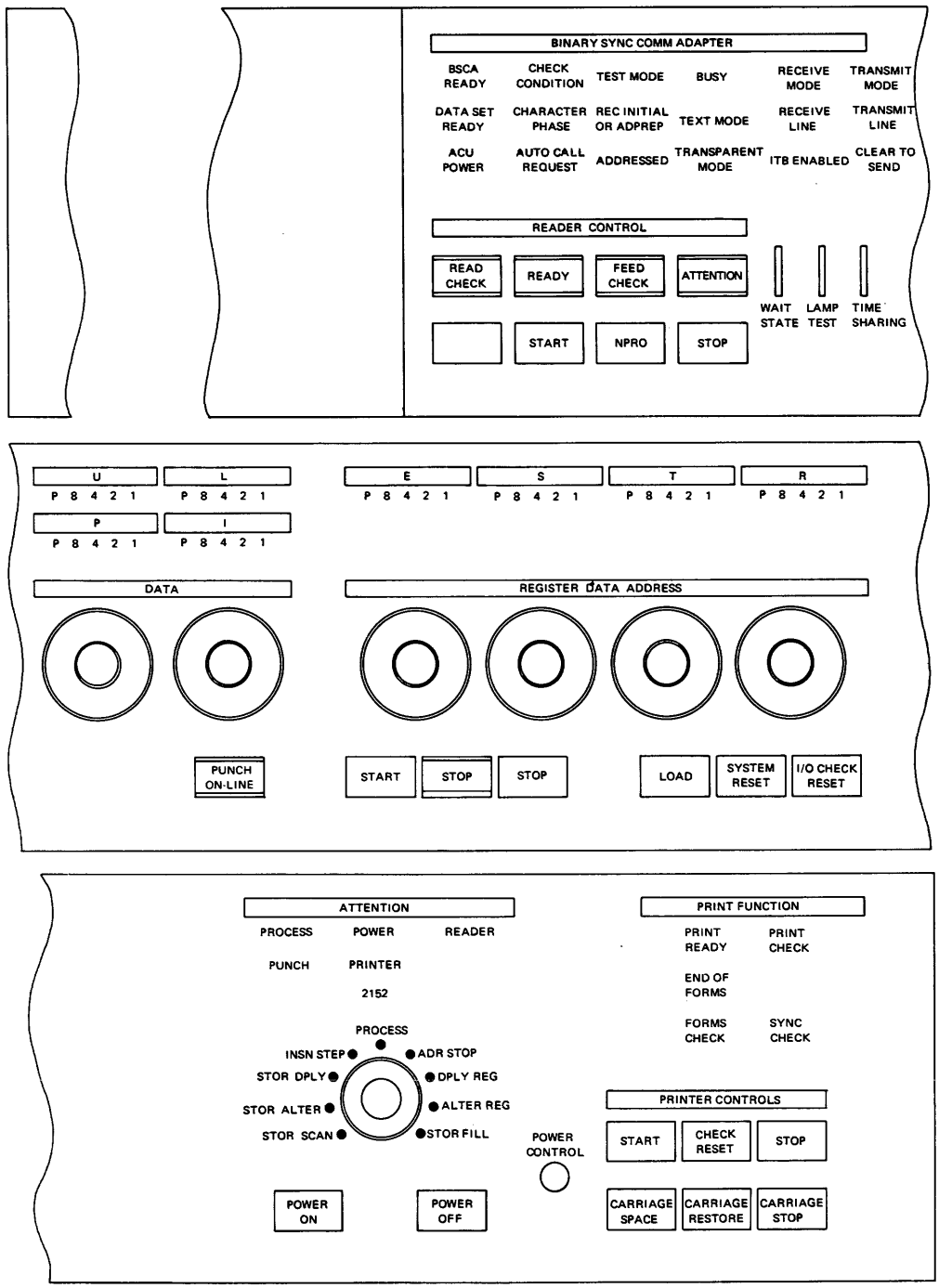


Figure 1-15. Console

System Reset Key

Operation of the System Reset key stops the control unit immediately, including all I/O operations which may be in progress. All error

conditions are reset. The condition code, channel mask, ASCII mode bit, and overlap mode bit in the PSW are reset to zero. A system reset sets the mode to EBCDIC. The instruction address portion of the PSW is not changed. All registers displayed on the console are reset to zero, that is, all bit lights are off except the parity bits (P). The system reset function is also performed when the power on or program load sequences are initiated.

Note: If successful operation cannot be resumed after the System Reset key has been pressed and the interrupted program has been reloaded, the attention of a Service Representative is normally required.

Punch On-Line

This indicates that the 1442 Card Punch is ready to accept instructions from the control unit. The following conditions must be satisfied:

1. Terminal must be in communications mode.
2. Ready indicator must be lit on the 1442 console.

Start Key

The start key starts or resumes operation of the control unit.

Stop Key and Indicator

Pressing the stop key stops the control unit at the completion of the execution of the instruction in progress. All time-shared I/O operations in progress continue to completion. The stop key indicator is on when the control unit is stopped by any stop condition.

Program Load Key

The following conditions must be established before the program load function is operable:

1. The mode switch on the control unit console must be in the Process, Address Stop, or Instruction Step position.
2. The 2922-3 Terminal Reader must be in a ready condition.

When these conditions are met, operating the Program Load key initiates the system reset function, followed by a read-card operation starting at the main-storage location set in the Address/Register Data switches.

If the mode switch is in any position other than Process, Address Stop, or Instruction Step when the Program Load key is operated, the control unit performs the system reset function and then stops.

If an error has occurred during loading of the control program, however, a code is displayed in the U-L registers. The operator can clear the errors listed in Figure 1-16 as shown in the "Next Action" column. Any other errors require the attention of the Service Representative.

Display in U-L Registers	Meaning of Display	Next Action
00	Successful load	Set Mode switch on CU to PROCESS
01	Read error and/ or feed check	Restart the control program load procedure
02	Card missing, out of sequence, extra, or duplicate	Check columns 73-80 for correct card sequence. Incorrect numeric order means a card is out of sequence; incorrect alphabetic order means a section is out of sequence. Correct the sequence error and restart the load procedure.
03	Section out of sequence	

Figure 1-16. Control Program Load Errors

I/O Check Reset Key

The I/O Check Reset key resets all program testable I/O error indicators. See "Operating Conditions."

Power Control

When this key-operated, power-on control switch is turned to the ON position, the Power-On key is operative. When the power is on, the key may be removed. The Power-Off key does not depend on the power control switch.

Emergency Pull

This switch (not shown) is only present if the optional 1442-5 card punch is installed. In an emergency, this switch may be pulled to immediately disconnect all power from the system. This switch must be manually reset by a service representative before power may be restored.

Power On Key

The power-on sequence is initiated when this key is operated if power is not on. Operation of the Power On key has no effect if power is on.

Power Off Key

Operation of the Power Off key removes power from the terminal.

Attention Indicators

The Attention indicators show the particular condition or the particular I/O device that has caused the terminal to stop. In this manner the operator is guided quickly to the respective device and an inspection of its console defines the stop condition in detail.

Process

This indicator shows that an error has occurred within the control unit. All operations are halted immediately and usually the attention of a Service Representative is required. However, operations can be resumed by first pressing the System Reset key and then loading the program again.

Power

This indicator shows that a power failure has occurred either in the control unit or in an I/O device. A power failure is defined as the loss of any of the voltages that are generated by the power supply or as an overheating condition in any of the gates in the control unit or I/O devices. Restore power by first pressing the System Reset key and then the Power On key.

Reader

This indicator shows an unusual condition in the 2922-3, such as a filled stacker, an empty hopper, an open machine cover, a read check, or a feed check. Machine checks usually require action from a Service Representative.

2152

This indicator shows an unusual condition in the 2152 Printer-Keyboard, such as an end-of-forms condition, a P1 check or a P2 check, a 2152 power supply failure, or disconnected cables.

Printer

This indicator shows an unusual condition in the 2922-2 Terminal Printer, such as a hammer misfire, a sync check, or a forms check.

Punch

This indicator shows an unusual condition in the IBM 1442 Card Punch Model 5, such as a feed check, an empty hopper, or a full stacker.

CONTROL SWITCHES

Mode Switch

Process

In the Process position, the control unit operates under control of the stored program. See "Operating Conditions."

Address Stop

The control unit stops when the program has reached the instruction which is located at the main-storage address indicated on the Address/Register Data switches. Otherwise it operates in the same way as in process mode. When an interrupt occurs and the stop address set up on the Address/Register Data switches is identical with the instruction address contained in the new PSW, the control unit does not stop on this address.

Instruction Step

In this mode of operation, the control unit executes one complete instruction for each operation of the Start key. Since the interrupt condition is always tested prior to the execution of an instruction, it is possible that an interrupt occurs when the Start key is pressed; in this case, the control unit stops after the instruction that is designated by the address in the new PSW is executed.

Storage Display

The eight-bit byte of data located in main storage at the address indicated on the Address/Register Data switches is displayed in the U-L register when the Start key is operated. The address of this byte of data is displayed in the E-S-T-R registers.

Storage Alter

To alter the contents of a main-storage byte, the operator must set the Address/Register Data switches to the desired location and select the bit configuration of the eight-bit byte of data to be entered by setting the two Data switches. When the Start key is operated, the byte indicated on the Data switches is entered. The byte which was entered is displayed in the U-L registers and the address of this byte is displayed in the E-S-T-R registers.

The eight-bit byte of data is represented in hexadecimal notation on the Data switches. Data switch 1 select the configuration of bits 0 1 2 3 and Data switch 2 selects the configuration of bits 4 5 6 7

Register Display

The contents of the eight general registers and also the contents of certain address registers may be displayed in this mode of operation. The register to be displayed must first be selected by Data switch 1 according to Figure 17.

The operator may then display the contents of the selected register in the E-S-T-R registers by operating the Start key. The number of the register selected is displayed as a four-bit binary number in the P register.

The first halfword of the last instruciton executed (OP code plus byte two of the instruciton) may be displayed in the E-S-T-R registers in this mode when Register 0 is selected (Figure 1-17).

Data Switch 1		Register Selected
0	(0000)	First two bytes of previous instruction
1	(0001)	I-Recall Address Register
2	(0010)	PSW (bits 0-15)
3	(0011)	PSW (bits 16-31)
4	(0100)	- -
5	(0101)	- -
6	(0110)	- -
7	(0111)	- -
8	(1000)	General Register 8
9	(1001)	General Register 9
A	(1010)	General Register 10
B	(1011)	General Register 11
C	(1100)	General Register 12
D	(1101)	General Register 13
E	(1110)	General Register 14
F	(1111)	General Register 15

Figure 1-17. Data Switch 1

Register Alter

The contents of the eight general registers and certain address registers may be altered in this mode of operation. The register to be altered must first be selected by Data switch 1 as shown in Figure 17. The data to be entered into the selected register is set on the four Address/Register Data switches. When the Start key is operated, the data is entered into the selected register. The data which was entered is displayed in the E-S-T-R registers and the number of the selected register is displayed as a four-bit binary number in the P register.

Storage Scan

In this mode of operation, the control unit starts at the address indicated on the Address/Register Data switches when the Start key is operated. It scans through main storage (from low to high) until a parity error is detected or the Stop key is operated. If the end of main storage is reached during the scan, the scan continues from the beginning of main storage.

Note: The only correct method of terminating a scan operation is to press the Stop key.

Storage Fill

In this mode of operation, the control unit enters the contents of Data switches 1 and 2 into all positions of main storage, starting at the address indicated by the Address/Register Data switches. Operation of the Start key causes the loading of storage to begin. The loading proceeds from low address locations to high address locations, and wraps around from the high end of main storage to the low end of main storage. The loading operation is terminated by operation of the Stop key.

Note: The only correct method of terminating a fill operation is to press the Stop key.

Data Switches

Data switches 1 and 2 are physically located on the control unit console below the U and L registers. They are 16-position rotary switches with hexadecimal notation. Data swtich 1 selects the configuration of bits 0 1 2 3 and Data switch 2 selects the configuration of bits 4 5 6 7 in an eight-bit byte of data in Storage Alter mode. Data switch 1 is also used for register selection in the Register Dsplay and Register Alter modes. The hexadecimal representation for all bit combinations of an eight-bit byte is shown in Appendix E.

The position of the Data switches may be changed without disturbing control unit operation.

Address/Register Data Switches

The four 16-position rotary Address/Register Data switches are numbered 1, 2, 3, 4 from left to right and are physically arranged on the control unit console directly below the E-S-T-R registers. These switches have hexadecimal notation. A hexadecimal-decimal conversion table is presented in Appendix E. Refer to the Address Stop, Register Display, and Register Alter modes for the functions of the Address/Register Data switches.

The position of the Address/Register Data switches may be changed without disturbing control unit operation, except that a stop may occur if in the Address Stop mode.

Time-Sharing Switch

When this switch is on, the execution of input/output operations is time-shared with other control unit operations. When the switch is off, each input/output operation is completely executed before the control unit continues with the next sequential instruction.

Lamp Test Switch

This switch may be employed to visually check for burned out indicator lamps. All indicators on the Terminal Console should be on when the Lamp Test switch is operated.

OPERATING CONDITIONS

During normal operation of the control unit under control of the stored program, the Stop indicator on the console is turned off. This indicator is on for all stop conditions of the control unit. Three types of stop conditions may occur:

1. Process Check: A process check indicates some fault in the control unit such as an even parity. It can be reset by operating the System Reset key.

2. A Normal Stop: In this case, only the Stop indicator is on.

3. A Programming Error Stop: In this case, the Stop indicator as well as an error number display in data register I is on. The error number in data register I is displayed in the binary coded form. For example, the code 0111 represents the number 7 and indicates that the second operand of a decimal instruction is equal to or greater in size (field length) than the first operand.

For normal stop conditions other than a programmed halt operation, the operation code of the Next Sequential Instruciton (NSI) is displayed in the U-L registers and the address of the NSI is displayed in the E-S-T-R registers on the console.

For a programmed halt operation, the halt operation code is displayed in the U-L registers and the direct or effective address derived from the B1 and D1 fields of the instruction is displayed in the E-S-T-R registers.

For programming error stop conditions, the operation code of the instruction in which the error condition occurred is displayed in the U-L registers, and the address of this instruction is displayed in the E-S-T-R registers.

Normal Stop Conditions

A normal stop of the control unit occurs as a result of:

1. Operation of the Stop key.
2. A programmed halt operation.
3. An address stop in the Address Stop mode.
4. Operation in the Instruction Step mode.

The control unit is also in a normal stop condition following operation of the System Reset key or when power is first applied, but all register displays are blank (0000).

The control unit starts with the instruction located at the address contained in the program status word (bits 16-31) when the Start key is operated. The PSW contains the address of the NSI when the control unit stops as a result of one of the conditions listed above.

All input/output operations in progress when the control unit stops as a result of the conditicns listed above are completely executed. All input/output operations in progress are terminated when the System Reset key is operated.

Programming Error Stop Conditions

A programming error stop occurs as a result of the conditions listed below. With each condition is a four-bit binary number which is displayed in the I register when a stop due to the error condition occurs.

1. Operation code invalid (0001).

2. Addressing error.

   a. An instruction address or an operand address refers to the protected first 144 bytes of main storage (0100).

      Note: As an exception, the error display 0100 may also occur when a data error has arisen during the execution of a Zero and Add Packed instruction. This exception applies when the data error is caused by a low-order half-byte not containing a valid digit code.

   b. An instruction address or an operand address is outside available storage (0101).

   c. The R1 or R2 fields of an RR cr an RX format binary instruction contain binary values 0 through 7, the R1 field in a Branch and Store instruction contains binary values 0 through 7, or the R2 field of an RR format branch instruction contains binary values 1 through 7 (0101).

   d. Any instruction part is located at the last available storage position (0101).

3. Specification error.

   a. An instruction address is not located on a halfword boundary of main storage (0110).

   b. A binary operand is not located on a specified boundary of main storage (0110).

   c. For Decimal Add, Decimal Subtract, Zero and Add, and Decimal Compare instructions, the length code L2 is greater than the length code L1 (0110).

   d. For Decimal Multiply and Decimal Dvide instructions, the length code L2 is greater than 7 or greater than or equal to the length code L1 (0110).

   e. Bits 12 through 15 of an RX format instruction are not all zero (0110).

f. The field length specified in an input/output instruction is zero.

Note: For the BSCA field length specifications which exceed the maximum do not cause an error indication, but the bytes specified in excess are not processed. The program continues undisturbed.

4. Binary overflow check (1000).

5. Data error.

a. A sign or digit of an operand in the decimal instructions Zero and Add, Add, Subtract, Compare, Multiply, or Divide is incorrect or the operand fields in these operations overlap incorrectly (0111).

Note: Care must be taken when displaying error locations after a data error has arisen during a Compare Decimal operation. If the error was caused by an incorrect digit code in the first four bits of a byte, displaying this half-byte will cause a change in the last four bits of the byte. Depending on the bit configuration, incorrect parity can arise and cause a process error.

b. The multiplicand field (first operand) in a Decimal Multiply instruction has insufficient high-order zeros (0111).

c. An invalid digit code is contained within the second operand field of an Edit operation (0111).

6. Decimal divide check (1011).

If the Start key is operated before the error condition is corrected, the control unit again attempts to execute the instruciton in which the error occurred.

If the error is corrected without altering the contents of the PSW, the control unit executes the instruction in which the error occurred and continues with the program when the Mode switch is placed in the Process or Address Stop position and the Start key is operated. Caution must be exercised in the case of error conditions that occur during the actual processing of data. A second atempt to execute an instruction which has previously been partially executed without reconstructing the first operand causes erroneous results.

All input/output operations in progress when the control unit stops as a result of a programming errcr condition are completely executed.

If an address which is outside available storage is encountered during an input/output transfer, the transfer and the input/output

operation are terminated as if the complete data field as specified in the input/output instruction had been transferred; an error condition does not occur.

Process Error Stop Conditions

When an internal parity error occurs in the control unit, the control unit stops immediately. All input/output operations in progress are terminated.

The Process indicator on the console will be on. The process error condition must be reset by operation of the System Reset key on the console.

After a process error, the restart procedure cannot be used. The program must be loaded again and started either from the beginning or at a check point.

INPUT/OUTPUT OPERATIONS

Transfers of information to main storage from sources external to the control unit and from main storage to external destinations are referred to as input/output operations. For card I/O, the communication adapter and printer, three types of instructions are provided: Transfer I/O, Control I/O, and Test I/O and Branch.

A Transfer instruction (XIO) controls the transfer of data between main storage and the input/cutput device.

A Control instruction (CIO) directs an input/output device to perform a specified function, such as select a stacker pocket or initiate a carriage skip.

A Test I/O and Branch instruction (TIOB) causes an inquiry to an input/output device for a particular condition (for example, reader busy or end of form); if the tested indicator is on, the program branches to the specified address.

If the Time-Sharing switch is on, processing operations in the control unit are time-shared with the transfer of data between main storage and the input/output devices. When an input/output device requests service, processing is suspended only for the time required to send or accept the input/output data.

DATA FORMAT

Input/output data, located in eight-bit bytes in main storage in variable-length fields, may be in the zoned, binary, or packed format. The data format in which an input/output device may accept or present data is, however a characteristic of the device.

Input data is translated from the code form of the input device to the Extended Binary-Coded-Decimal Interchange Code employed by the control unit internally as the data is received. Output data is translated from the internal control unit code to the code form of the output device as the data is transferred.

CONDITION CODE

The status of an I/O device addressed by an XIO instruction, and under certain conditions, a Control I/O instruction, sets the condition code of the PSW at the time the execution of the instruction is completed. The condition code indicates if the I/O device has initiated the operation specified, and if not, the reason for the rejection. The condition code can be used for decision-making by subsequent branching operations. The condition code is set to 00, 01, 10, or 11 by an instruction to indicate the status of the I/O device addressed.

Condition Code	Status
00	Available
01	Working
10	Not Used
11	Not Operational

Available

Indicates that the addressed I/O device is operational, does not contain data or error check conditions, and is not busy with a previously initiated operation.

Working

Indicates that the addressed I/O device is executing a previously initiated operation.

Not Operational

Indicates that the addressed I/O device is in a not-ready status, or an error or a data check condition exists on the device.

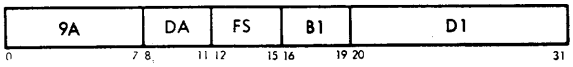
The operation specified by an XIO instruction is initiated only when the addressed I/O device is in the available state. If an I/O device is specified which is not a part of the system, a no-operation occurs, and the condition code is not changed.

INSTRUCTION FORMAT

The three input/output instructions and their formats are as follows:

Test I/O and Branch

TIOB D1 (B1), UF



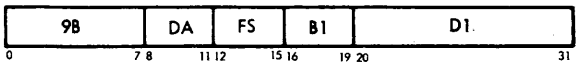
The Device Address (DA) specifies the I/O device in which a condition is to be tested.

The Function Specification (FS) specifies the particular condition or indicator to be tested in the I/O device addressed.

If the condition tested in the addressed I/O device is on, the updated instruction address is replaced by the branch address dervied from the B1-D1 fields; otherwise, normal instruction sequencing continues with the updated instruction address.

Control I/O

CIO D1 (B1), UF



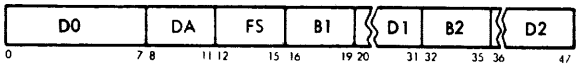
The Device Address (DA) specifies the I/O device in which a control function is to be performed.

The Function Specification (FS) specifies the particular component (it may also specify the primary function of that component) in the I/O device addressed.

A detailed specificiation of the control function to be performed is derived from the contents of the B1-D1 fields, according to the rules for direct or effective address generation. If the detailed specification derived from the B1-D2 fields is all zero, a no-operation occurs.

Transfer I/O

XIO D1,(UF,B1),D2(B2)



The Device Address (DA) specifies the I/O device to which output data is to be transmitted, or from which input data is to be received.

The Function Specification (FS) specifies the input or output function to be performed on the I/O device addressed, and also the particular component of the addressed device (when required).

The main-storage location of the first byte in the input or output data field is derived from the contents of the B1-D1 fields according to the rules for direct or effective address generation.

The field or record length of the input or output data in main storage is derived from the contents of the B2-D2 fields.

The field length specifications for input or output data fields in main storage is the acutal number of bytes in the field, whereas for variable field length processing operations, the field length specification is the number of bytes extending beyond the first byte.

ERROR CONDITIONS

Error conditions which may occur in the control unit during the instruction or service phases of input/output operations are listed below.

1. Operation code invalid.

2. Addressing error.

   a. An instruction address or the address of an input/output data field refers to the protected first 144 bytes of main storage.

   b. An instruciton address, the address of an input/output data field, or a branch address is outside available storage. If an incorrect branch address has been specified, the control unit indicates the error during the execution of the branch instruction itself.

   c. Any instruction part is located in the last two main-storage positions.

3. Specification error.

   a. The low-order bit of an instruction address is one.

   b. The field length specified for an input/output data field is zero or is greater than the maximum allowable number for the I/O device addressed. Exception to this rule is the BSCA, where no specifidation error occurs when a field length exceeding the allowable maximum (4095 bytes for BSCA) is specified; this is because, in BSCA instructions, the four high-order bits of the field length specification are ignored.

4. Control Unit parity error.

An input/output instruction containing an I/O device address which specifies a device not attached to the system is treated as a no-operation.

To ensure correct function, only valid specifications as defined in this manual should be used.

INTERRUPTION

Interruption is the general term applied to an automatic branch in the control unit program. The branch is automatic in the sense that it occurs when the conditions exists, and is independent of a programmed branch instructicn.

In the control unit, interruption is provided only for the channel end input/output condition. Channel end condition of an input/output device is defined as the time (in the mechanical cycle of the device) at which the data transfer has been completed.

In the time-shared mode of operation of the control unit and the input/output devices, the transfer of data between the input/output devices and main storage in the control unit is asynchronous with respect to processing operations. Thus, the channel end condition of an input/output data transfer operation may occur at any time in the instruction or execution phase of any processing operation. It is also possible that more than one input/output device would reach the channel end condition during a specific processing operation. The channel end conditions are stored in the form of interrupt bits. These bits are reset when the respective interrupts occur or are reset by a system reset or load operation.

The control unit is in an interruptible state when the channel mask bit in the PSW is 1, and is in a non-interruptible state when this bit is 0. The channel mask bit is reset to 0 by operation of the System Reset or Program Load key. The channel mask bit may be altered by a Set PSW instruction or by an interruption.

When the control unit is in the interruptible state (channel mask bit is 1), a test is performed by the control unit at the end of each processing operation to determine whether or not a channel end condition exists for any input/output device. (The instruction and start phases of input/output operations are also considered as processing operations.)

The test for a channel end condition is performed in an established priority sequence. The program continues with the next sequential instruction if no channel end conditions exist. When one or more end conditions exist, the first one encountered in the priority sequence causes an interruption to occur. The channel end condition which causes the interruption is reset. The interruption is performed by storing the PSW in fixed main-storage location 144 and obtaining a new PSW from another fixed main-storage location (148) before the program continues. Since the new PSW contains the address of the next sequential instruction, the interrupt is equivalent to a branch operation. The PSW which is stored in main storage location 144 is referred to as the old PSW.

The old PSW (stored at main storage location 144 when an interrupt occurs) contains the device address of the I/O device which caused the interruption (bits 8-11); the primary function which the device was performing (bits 12-15); the address of the next sequential instruciton, and the condition code. Primary functions are read, or print.

Note: The term channel in the above context is the card read or printer adapter and the BSCA adapter.

The following is a list of device addresses and function specifications contained in bits 8 through 15 of the old PSW stored at main-storage location 144 when an interruption due to the

corresponding channel end condition occurs. The list is in the priority sequence for these interruptions.

Channel-End Condition	DA	FS
2922-3 Read	1	2
2922-2 Print	4	0
BSCA Any XIO	5	6
BSCA ITB Received	5	2
2152 Read	E	1
2152 Write or Carrier Return	E	2
2152 Inquiry Request	E	3
1442-5 Punch	3	4

The channel mask bit in the new PSW (obtained from main-storage location 148) may disable further interruptions in the routine which begins at the next sequential instruction address specified in the new PSW. As a means of returning to the point in the program at which the interruption occurred, a Set PSW instruction, in which the specified address is 144, may be used as the last instruction in the routine.

LAST CARD CONTROL

The card I/O device has a card reading unit and is provided with a testable Last Card indicator.

The last card condition in the 2922-3 Terminal Card Reader is set when a Read Card instruction for the 2922-3 is encountered in the program and no cards is at the read station. The channel-end condition is set, and the 2922-3 is placed in a not-ready status. The Last Card indicator may be tested and is reset by a Test I/O and Branch instruction in which the device address is 1 and the function specification is 4. The Last Card indicator is also reset when a new deck of cards is run in on the 2922-3 and the first Read Card instruction is encountered in the program; or it can be reset by the system reset function of the control unit.

COMPATIBILITY

The organization of the 2922 terminal is, in most respects, identical with that of the System/360. The data and instruction formats are a compatible subset. The majority of operations in the control unit instruction set are compatible. Some instructions and features in the 2922 terminal are different from those in the System/360. These differences are described in this section.

The control unit exerts direct control over all the I/O devices attached to it. However, I/O operations are initiated, halted, or tested by program instruction, which select the unit to be used. The I/O instructions determine what operation is performed (read, write, and so on) and where the data is stored.

GENERAL REGISTERS

The control unit has eight general registers. They are numbered 8-15 and correspond to the same registers in a System/360. They are one halfword in length compared to fullword length on System/360 models. Results of valid control unit binary operations are identical to corresponding operations in the System/360 except for the recognition of an overflow condition.

MAIN STORAGE ADDRESSING

The first 144 bytes of main storage are protected and program reference to this area results in an error condition.

ADDRESSING

An address used to refer to main storage in the 2922-1 may be specified by either of two methods: direct addressing or effective-address generation. Direct addressing is indicated when the high-order bit in the B-field of an instruction is zero. The address from this method is derived from the binary value of the low-order 14 positions (for addresses up to 16383) of the combined B and D fields. Effective-address generation is automatically performed when the B-field refers to general registers 8-15; in forming this address, the control unit operates identically to the System/360.

INPUT/OUTPUT DEVICES

2922-3 TERMINAL CARD READER

INTRODUCTION

This section contains the operating principles and features of the IBM 2922-3 Terminal Card Reader as used with the 2922-1 Terminal Control Unit. Descriptions of machine timings and operator procedures are also covered.

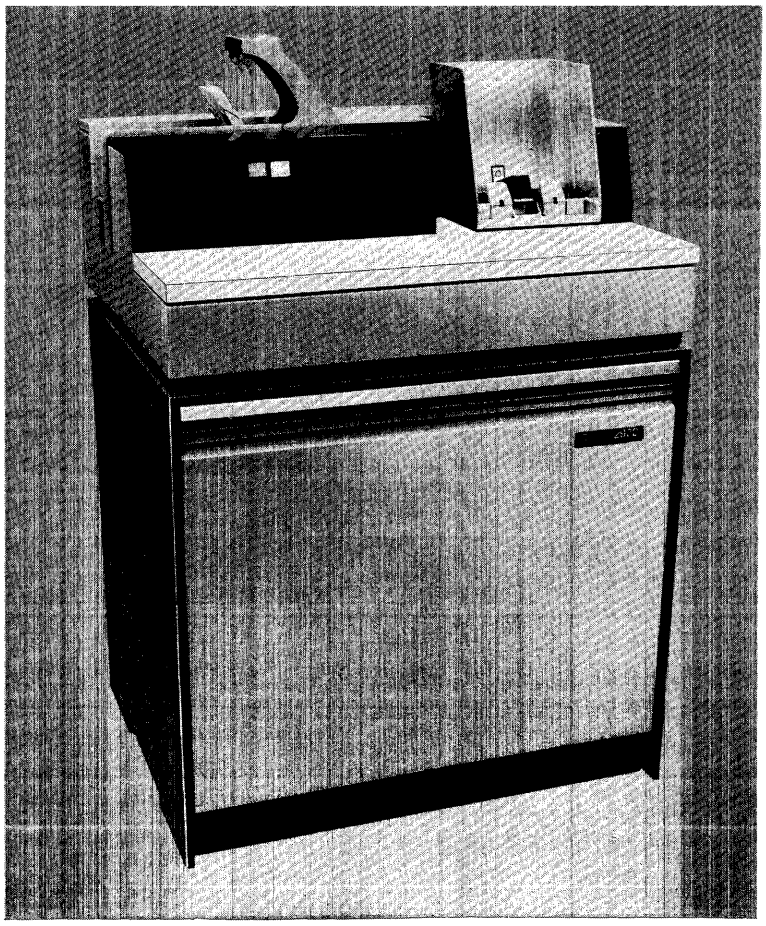


Figure 2-1. 2922-3 Terminal Card Reader

The 2922-3 Terminal Card Reader provides punched card input for the IBM 2922 Programmable Terminal. Cards are read at a maximum rate of 500 cards per minute (cpm).

The control unit program controls card reading. Cards are read serially, that is, column-by-column, beginning in column 1. Each column is read twice and the two readings are compared to check reading accuracy. The control unit detects off-punched or mispositioned cards.

MACHINE FEATURES

CARD PATH

Cards travel from a card feed hopper, through the read station, and into the stacker (Figure 2-2). The hopper and stacker have capacities of approximately 1200 and 1300 cards, respectively.

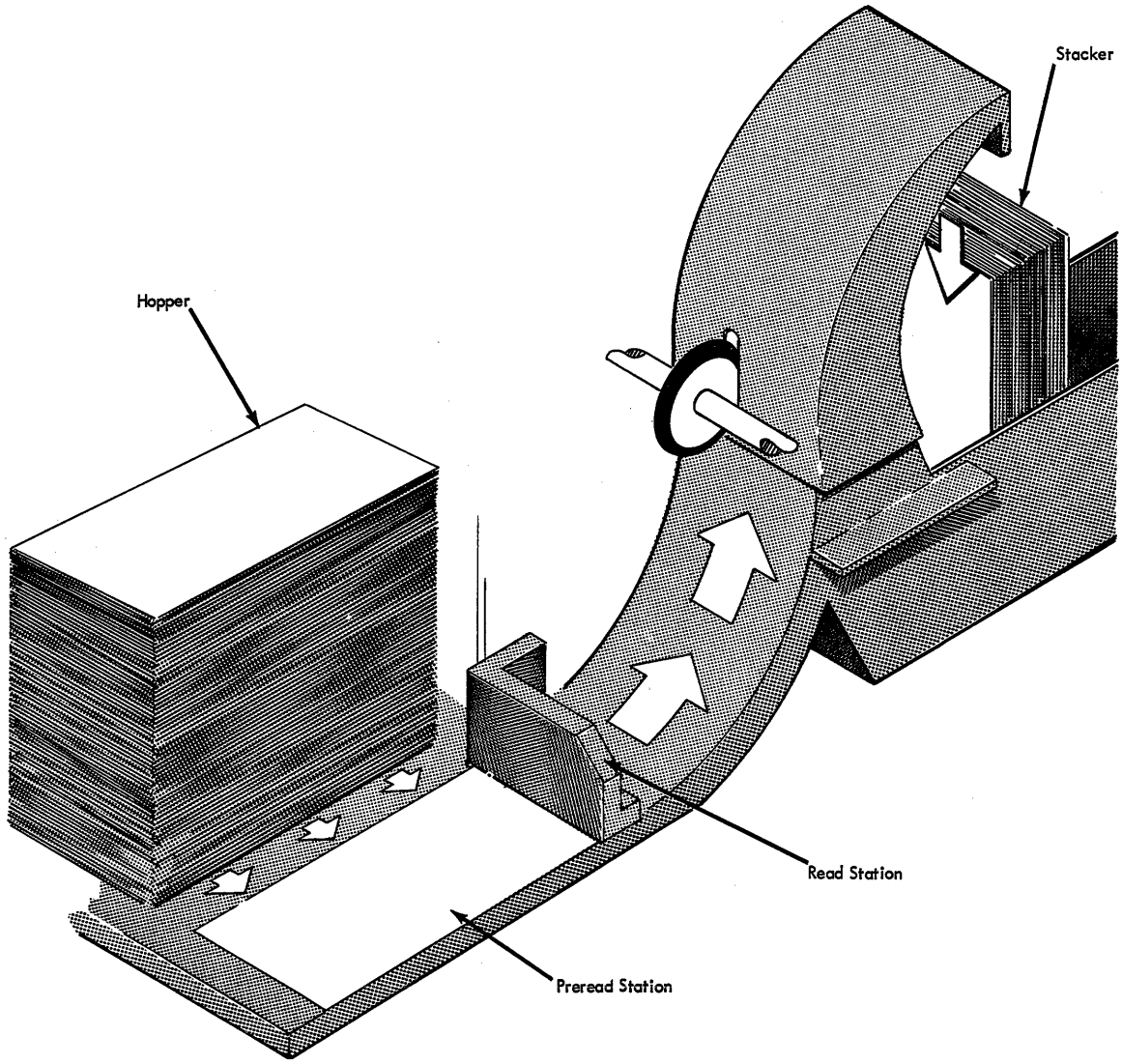


Figure 2-2. 2922-3 Card Path

Reading is done by light cells, which convert light energy into electrical energy. Twelve read cells, one for each row of punching positions in the card, are exposed to light as holes in the card pass by them (Figure 2-3).

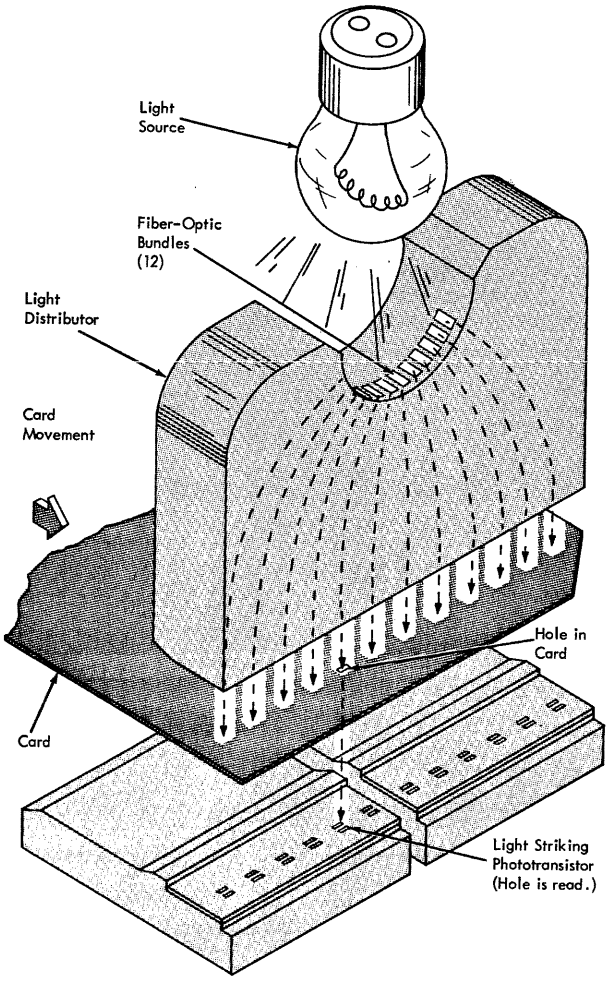


Figure 2-3. Schematic of 2922-3 Read Station

OPERATING KEYS AND INDICATORS

The operating keys and indicators, Figure 2-4, are strategically located on the left end of the terminal console.

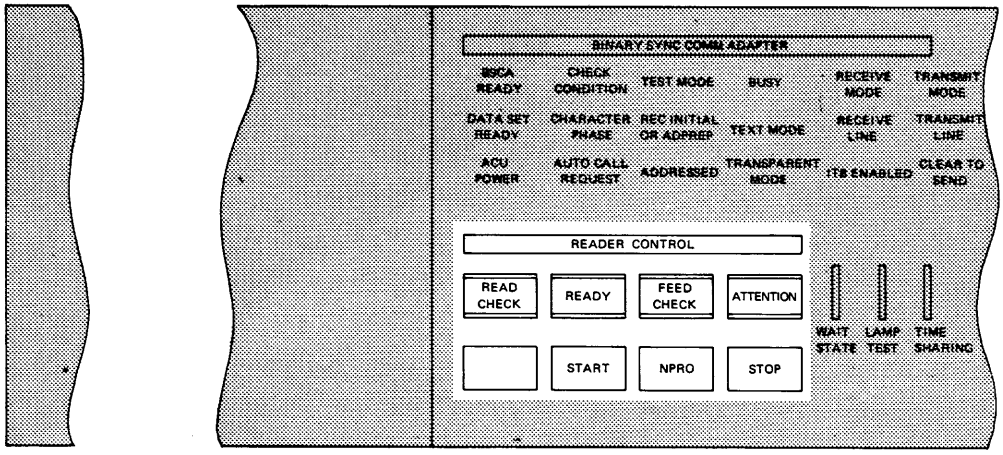


Figure 2-4. Console Keys and Indicators

Indicators

Ready

This indicates that the 2922-3 is ready to accept instructions from the control unit. The following conditions must be satisfied for the ready indicator to be on:

1. Power on
2. Cards in hopper, except during last card sequences
3. Cards in pre-read, except during last card sequences
4. None of the following errors active

   Read Check
   Feed Check
   Attention

5. Machine not stopped with Stop key

6. Cards not being run out with the NPRO key

7. Last card indicator not set.

If the 2922-3 is addressed by a Transfer instruction when the Ready indicator is off, the Reader indicator on the console is turned on. The not-ready condition is indicated to the 2922-1 program through the Device Status Word, which should be monitored before a Read instruction is issued to the 2922-3.

Feed Check

This indicator turns on when a card is mispositioned in the card path or when certain equipment malfunctions occur. When turned on by a mispositioned card, the Feed Check indicator can be turned off by the following procedure:

1. Empty the hopper.

2. Raise the machine cover.

3. Clear the card path of all cards.

4. Close the machine cover.

5. Press the NPRO key.

NOTE: Restart the program according to procedures described in the Operating Procedures manual for the type of program being run.

If repeated feed checks occur, notify your Service Representative.

The feed check condition turns off the Ready indicator and turns on the Reader indicator on the 2922-1 when the next Transfer instruction is issued to the 2922-3.

Read Check

Indicates that a card is sufficiently mispositioned to impair reading. For the Programmable Terminal, this indicator may also indicate that an invalid code has been detected.

The read check condition also turns off the Ready indicator and turns on the Reader indicator on the console.

When the 2922-3 is operating with the Terminal Control Unit, the read check is turned off by:

1. Pressing the I/O Check Reset key on the console, or
2. Pressing the System Reset key on the console, or
3. Pressing the Load key on the console, or
4. Pressing the NPRO key on the console, or
5. Testing the read check indicator by a Test I/O and Branch instruction.

When the 2922-3 is operating with the Terminal Control Unit, the NPRO key must be used to turn off the Read Check indicator.

Attention

Indicates an open cover or a full stacker. The indicator is reset by correcting the condition. The Attention indicator condition turns off the Ready indicator and turns on the Reader indicator on the Terminal Control Unit when the next Transfer instruction is issued to the 2922-3. Operator intervention is required to restore the 2922-3 to the ready condition.

NOTE: If the cover is opened during an operation in progress, the 2922-3 motor stops and the Feed Check indicator is turned on, indicating the possibility that an operation was not completed.

KEYS

The operating keys are used to start and stop the 2922-3 and to clear the card path without processing the cards.

Start

Pressing the Start key places the 2922-3 in ready status, providing all the conditions listed under Ready Light are satisfied. If no card is at the preread station, one card is fed to that position.

Stop

Pressing this key stops the 2922-3 and removes it from ready status. If a card read operation is in progress, the operation is completed before the machine stops. Pressing the Start key restarts the 2922-3.

NPRO (Nonprocess Runout)

Pressing this key ejects all cards in the card path to the stacker without being read. Before the NPRO key will function, any card jam must be removed and the hopper must be empty.

DATA FLOW

The data flow from the 2922-3 to the control unit storage is as follows:

The normal IBM card code is translated into the Extended Binary Coded Decimal Interchange Code (EBCDIC). The Terminal Control Unit also has the added ability of reading in column binary mode. Figure 2-5 shows the data flow and format for the Programmable Terminal.

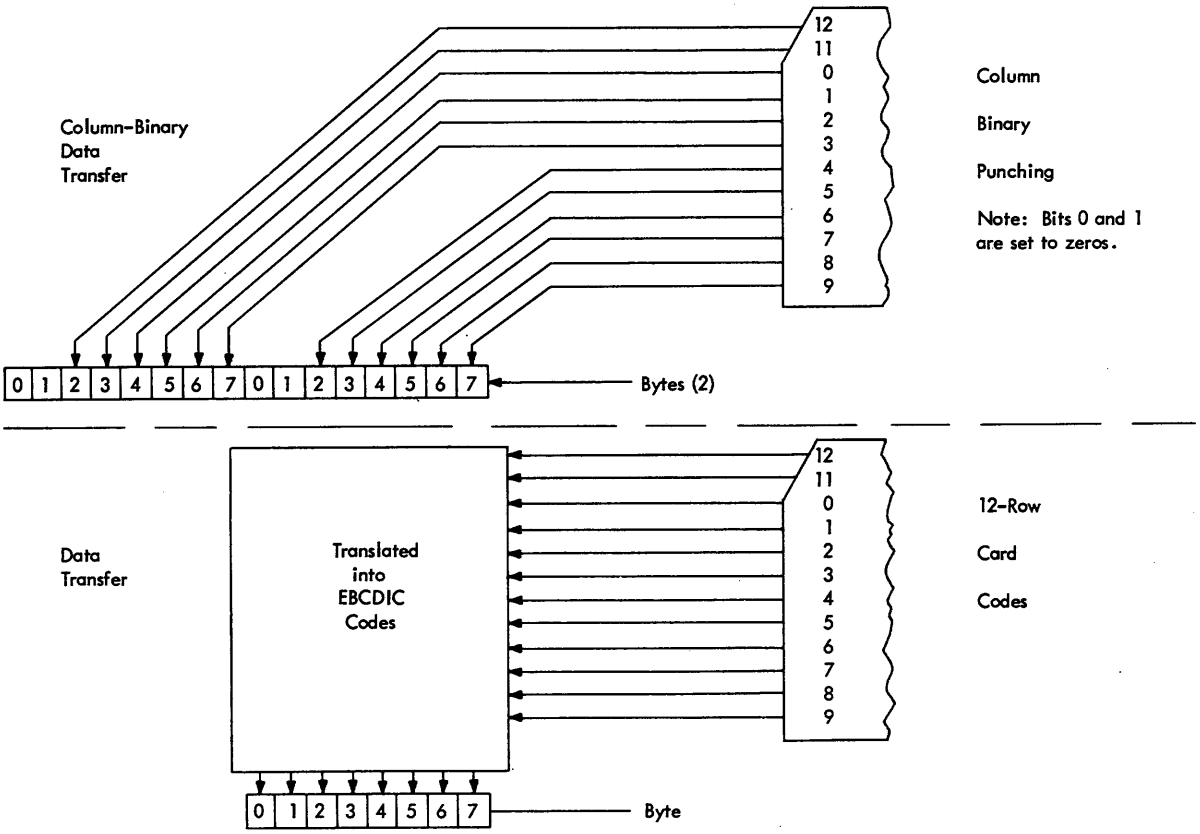


Figure 2-5. Data Storage Formats

PROGRAMMING

CONDITION CODES

Three condition codes (working, available, and not operational) indicate the status of the 2922-3 to the Terminal Control Unit. Transfer of data from the 2922-3 to the control unit can occur only when the 2922-3 is in the available state. The 2922-3 is in the available state whenever the Ready indicator is on, the Read Check indicator is not on, and a previous read operation is not in progress.

The 2922-3 is working during the execution of a previously initiated Read Card instruction. A new instruction cannot be accepted until the previous instruction has been executed and the 2922-3 has returned to the available state.

The not-operational state exists whenever the 2922-3 Ready indicator is off or when the Feed Check or Read Check indicator is on. The feed check condition is not program testable and is indicated to the program by the not-operational condition code. Operator intervention is required to restart from either condition.

TRANSFER INSTRUCTION

The Transfer instruction (Read Card) governs the transfer of data from the 2922-3 to the Terminal Control Unit.

Data transfer begins with the data read from column 1 of the card and continues until the specified number of columns have been transferred from the card reader to the control unit storage.

NOTE: A scored card should be designed so that the last column read is to the left of the score to avoid the likelihood of the score causing a read check.

TEST I/O AND BRANCH INSTRUCTION

Test I/O and Branch instructions direct the control unit to interrogate the 2922-3 for a particular condition (indicator), such as read check indicator.

READER BUSY INDICATOR

The reader busy indicator (data transfer in progress) is tested; if the indicator is on, the program branches to the address specified by the instruction. The indicator is turned off when the data transfer has been completed.

READ CHECK INDICATOR

The read check indicator is tested; if the indicator is on, the program branches to the address specified by the instruction. The read check indicator is turned off by the branch test; by the I/O Check Reset, the System Reset, or the Load key on the terminal console; or by pressing the NPRO key on the 2922-1 Terminal Control Unit console.

LAST CARD INDICATOR

The last card indicator is turned on when a Read instruction is given when no card is at the preread station. The indicator is tested and, if it is on, the program branches to the address specified by the instruction. The indicator is reset by the branch test or when a new deck of cards is run in on the 2922-3 and the first Card Read instruction is encountered in the program.

NOTE: When the last card feeds from the hopper, the 2922-3 Ready indicator goes off. (The next Transfer instruction directed to the 2922-3 when the Ready indicator is off, turns on the Reader indicator on the console.) At this point the operator either refills the hopper and presses the 2922-3 Start key to continue the operation, or presses the Start key without refilling the hopper. This restores the 2922-3 to ready status and allows the Terminal Control Unit stored program to read the last card and initiate a last card routine. The Card Read instruction that turns on the last card indicator also turns off the Ready indicator.

OPERATING PROCEDURES

INITIAL RUN-IN

All power to the 2922-3 is supplied from the power supplies in the Terminal Control Unit to which it is attached. When the power is turned on in the control unit, a feed check automatically results in the 2922-3. This is a design feature to ensure that no cards are left in the card path from a previous operation.

To start operation, the operator:

1. Presses the NPRO key to clear the feed check condition.
2. Loads the hopper with cards.
3. Presses the start key.

This places the 2922-3 in ready status and completes the initial setup.

JAM REMOVAL AND RESTART

REMOVAL

A feed check indication may result from a hopper misfeed, a jam, or a mispositioned card in the read station. Most feed check conditions can be physically corrected by emptying the hopper, fanning the cards, replacing creased or torn cards, and pressing the NPRO key. If this procedure does not turn off the Feed Check indicator, check for a card jam.

The entire transport and the stacker are exposed by swinging the rear top cover to the rear. At least half of any card in the card path is now visible and can be easily removed. Any card partially under a feed roll may be moved either forward or backward by turning the handwheel in the appropriate direction.

The hand wheel is located just below and to the rear of the read station. A plastic card guide over the preread station can be raised to aid in jam removal from that station.

After the card path has been cleared, restore the card guide, close the rear top cover, and press the NPRO key.

RESTART

To restart the Terminal Control Unit program after an NRPO operation, follow the procedures in the Operating Procedures manual for the type of program being processed.

LAST CARD

When the last card feeds from the hopper, the 2922-3 Ready indicator is reset. At this point the operator either refills the hopper and presses the Start key to continue the operation, or presses the Start key without refilling the hopper. This restores the 2922-3 to ready status and allows the control unit stored program to read the last card and initiate a last card routine.

INPUT/OUTPUT CONSIDERATIONS

Two basic timing considerations are associated with a 2922-3 attached to the Terminal Control Unit.

1. Card throughput in cards per minute (cpm).

2. Time available for other Terminal operations.

The Programmable Terminal is capable of performing operations such as reading and processing, essentially simultaneously. This ability is called "time sharing". When all 80 columns of a card are read, the control unit is interlocked (busy) for only 5 ms. The remainder of the cycle is available for other operations.

This time can be used by the programmer to accomplish the data manipulation, for instance, add, subtract, print, etc., required to carry out a data processing operation. If fewer than 80 columns are specified by the Read instruction, the interlock time is reduced by approximately 0.06 ms for each column not read.

The 2922-3 has a maximum card read speed of 500 cards per minute (cpm).

INSTRUCTIONS

TRANSFER INSTRUCTIONS (XIO)

Function	Op Code	DA	FS
Read Card	DC	1	2

The direct or effective address (drived from the B1-D1 fields of the instruction) specifies the leftmost main-storage location of the field where the input data will be located. The length of the input field is derived from the B2-D2 fields of the instruction.

The field length specification is restricted to binary values equal to or less than 80.

TEST I/O AND BRANCH INSTRUCTIONS (TIOB)

Function	Op Code	DA	FS
Test Reader Busy	9A	1	0
Test Reader Error	9A	1	1
Test Last Card	9A	1	4

Test Reader Busy

The Reader Busy indicator is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address.

Test Reader Error

The Read Check indicator is tested. If this indicator is on, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. The Read Check indicator is reset by the branch test.

Test Last Card

The Last Card condition is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. The Last Card condition is reset by the branch test.

CONTROL I/O INSTRUCTIONS (CIO)

A control instruction which specifies the 2922-3 is treated as a no-operation.

CONDITION CODE

The condition code is set 00, 01, or 11 when the execution of an XIO instruction is initiated. At this time the status of the respective I/O device is checked and the result of the check determines the setting of the condition code; 00 indicates available status, 01 indicates working status, and 11 indicates a not-operational status.

A data transfer specified by an XIO instruciton for the 2922-3 is initiated only when the 2922-3 is in the available state. The 2922-3 is in the available state when the ready indicator is on, the read check indicator is not on, and a previously initiated Read Card instruction is not in progress.

The 2922-3 is working (condition code 01) during the execution of a previoulsy initiated Read Card XIO instruciton. It cannot accept a new instruction until the previous instruction has been executed and the 2922-3 has been returned to the available state (condition code 00).

The 2922-3 is not operational when the ready indicator is off or when the Feed Check or Read Check indicators are on.

COLUMN BINARY CODE

The IBM punched card is a permanent data storing device that contains 80 storage positions (card columns). When the standard IBM card code (Figure 2-6) is used, each punched column contains either one digit or one character. The capacity of a card is increased to 160 storage positions, however, when the column binary code (see Figure 2-6) is used; each punched column can then contain two digits or two characters in six-bit Binary Coded Decimal (BCD) code. A column binary card cannot be read like an ordinary punched card because the meaning of the usual zones and digits (12, 11, 0, 1, 2, and so on) is changed to correspond to the bits of the BCD code. The punches in one card column represent the BCD bits B, A, 8, 4, 2, 1, from top to bottom.

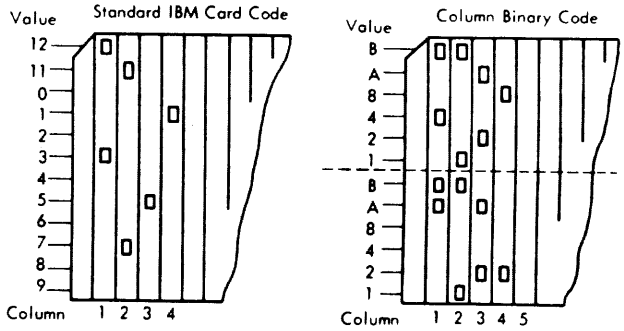


Figure 2-6. IBM Punched Card Codes

Column binary cards can be read by the card I/O device attached to the 2922-1 Terminal Control Unit. Each BCD character is transferred into one byte of main storage. Since the six-bit BCD character cannot fill a byte, the character is placed into bits 2-7 of the byte, and bits 0 and 2 of the byte are forced to zero. Column binary reading is performed from top to bottom of the card column, proceeding by columns from left to right (Figure 2-7).

The programmer must know that, when the 80 columns of a card punched in Column Binary mode are read, 160 adjacent bytes of core storage are needed. The field length specification derived from the B2 and D2 fields of the XIO instruciton (issued to read a column binary card) pertains to the number of BCD characters that are to be read. The field length must therefore be greater than zero and not more than 160.

The programmer can specify an odd number of characters (odd field length). When this occurs, reading always stops after the upper half of the last card column has been read. If the storage capacity is exceeded during a column binary read operation, reading stops when the highest storage position is reached, regardless of whether the field length is exhausted or not. No error indication is given.

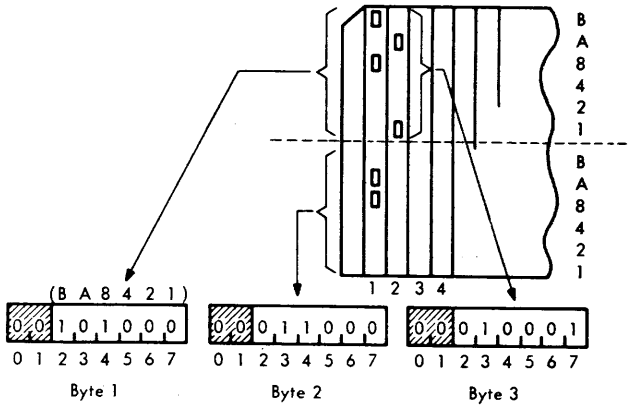


Figure 2-7. Column Binary Reading

The column binary mode is set by an additional bit in the FS field fo the XIO instruction. When bit 12 of the instruction (this is the 8-bit of the FS field) is present, the reader reads in column binary mode; when bit 12 of the instruction is zero (not present), the reader reads column-by-column. Thus, the column binary mode of reading is specified when the value of the FS is increased by eight.

The following list gives all read instructions, with the regular FS and the column binary FS, for all card I/O device which can operate in column binary mode.

2922-3 Card Reader

DA	FS	Function	Mode
1	2	Read Card	EBCDIC
1	10	Read Card	Column Binary

IBM 2922-2 TERMINAL PRINTER

INTRODUCTION

The IBM 2922-2 Terminal Printer (Figure 2-8) is an output device attachable to the IBM Terminal Control Unit. Fanfold paper, preprinted forms, or adding-machine type paper tape may be used. The 2922-2 can also be used to generate machine-readable input documents for all IBM optical character readers.

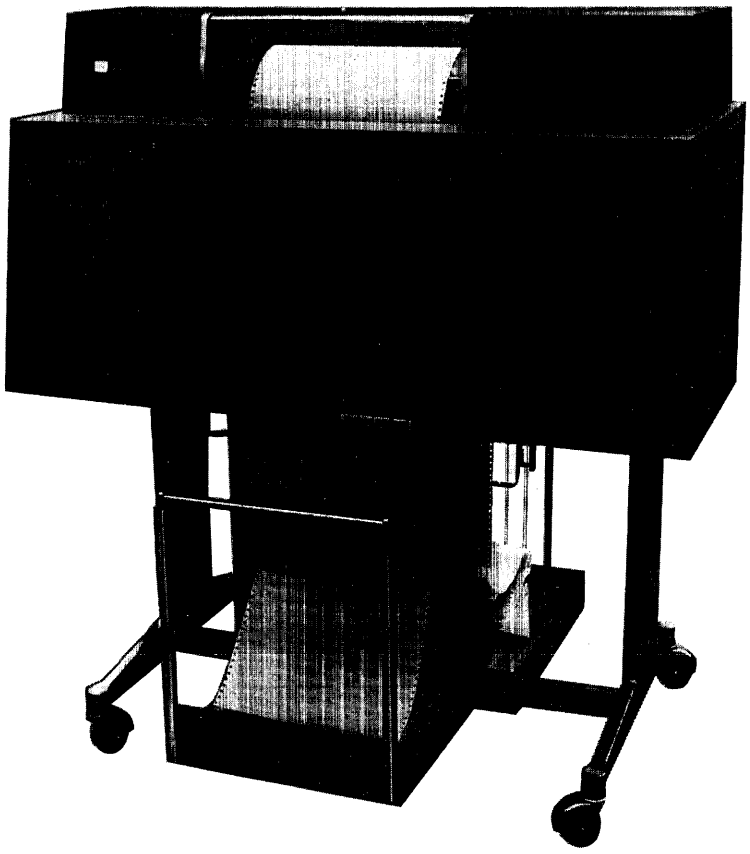


Figure 2-8. IBM 2922-2 Terminal Printer

A printer's rated speed is based upon the number of single-spaced lines that can be printed per minute. Actual printing speed depends also upon the character set used and the time required for processing and for moving paper.

The basic character arrangement for the printer is the fixed 60 character set (PL/1-60 Graphics). Each position can print 60 different characters: 26 alphabetic, 10 numeric, and 24 special characters.

Horizontal spacing is 10 characters per in.(25,4mm). Standard vertical spacing is six and eight lines per inch, controlled manually by the operator. Vertical spacing and skipping are initiated by the stored program. Standard skipping rate is about 33 in.(83cm) per second.

The production of machine-readable documents requires attention to print quality and format. The specific requirements for each optical reader are described in "Print Quality Requirements" and in manuals for particular optical character readers (See "References-OCR").

PRINTING METHOD

A schematic of the chain printing mechanism is shown in Figure 2-9.

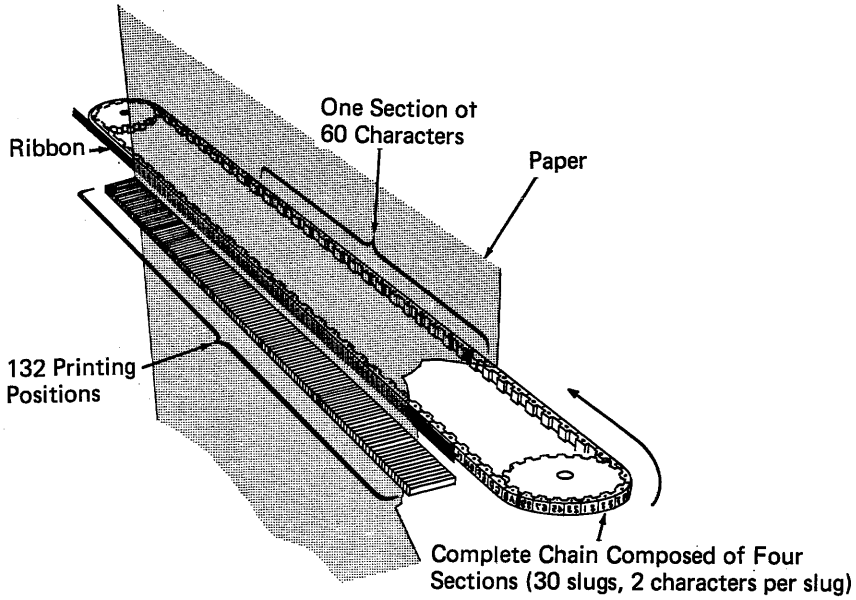


Figure 2-9. Chain Printing Mechanism

MACHINE COVERS AND SAFETY

The covers of the 2922-2 are designed to safeguard personnel against possible injury when the machine is in operation. Some hazards (such as moving mechanical parts) are obvious; others (such as electrical potentials and ascoustical noise) are not.

Although IBM maintains rigorous attention to safety on all its machines, the effectiveness of safeguards is decreased by failure to keep the covers closed when the machine is running.

The frames of all IBM equipment have been made electrically safe by standard grounding practices. The covers are acoustically designed to reduce the noise level below any possible hearing damage. Printer operation with the covers open, however, causes needless exposure to these unseen hazards. Because of this, IBM strongly recommends that all personnel associated with the equipment follow the simple safety first procedure of keeping the covers closed whenever the machine is in operation.

If necessary to open the side or rear covers of the 2922-2, use a coin or a key to operate the metal tab in the vertical gap between the covers.

OPERATING INFORMATION

OPERATOR CONTROLS

The keys and lights (Figure 2-10) are located at the right end of the terminal control unit console and provide operator control of the printer during setup and programmed interruptions that require operator attention.

PRINTING CONTROLS

Start

Pressing the start key places the printer in a ready status if the following conditions are met:

1. Power on
2. Forms guide plates closed
3. Feed clutch control properly positioned
4. Carriage control tape installed
5. Carriage brush assembly closed
6. No error conditions, such as print check, sync check, or end-of-form, exist.

The start key permits operating the printer after the end-of-form light is on, until channel 1 of the carriage tape.

A duplicate start key (Figure 2-11) is located at the rear of the printer for operator convenience.

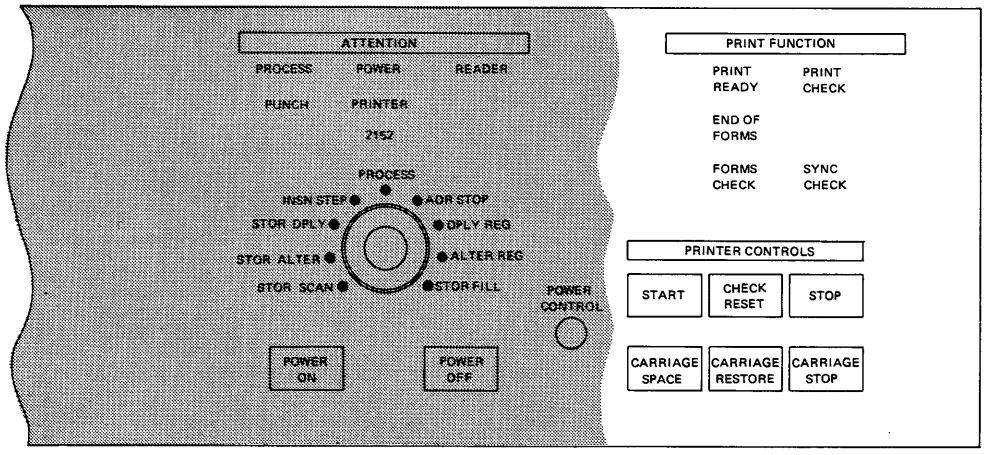


Figure 2-10. Operating Keys and Lights

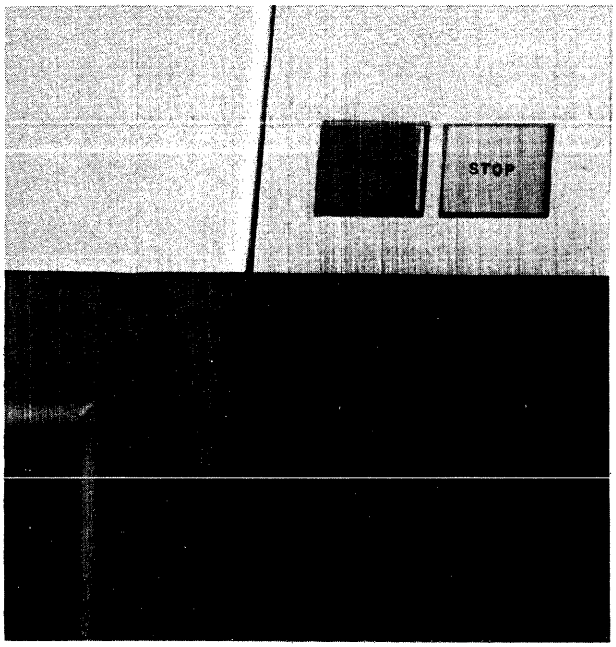


Figure 2-11. Printer Keys (Rear)

Check Reset

The check reset key resets a printer error indication. Pressing the start key restarts the operation.

Stop

The stop key stops the printer at the completion of the current operation. A duplicate stop key (Figure 2-11) is located at the rear of the machine for operator convenience.

Print Ready

The print ready light indicates that the printer has been conditioned by the operator to accept initial instructions and subsequent commands from the controller. This light turns off when:

1. The stop key is pressed
2. The carriage stop key is pressed
3. An end-of-form is indicated
4. An error condition (such as form check, sync check, or print check) occurs.

Print Check

The print check light indicates a malfunction in the printer circuits. The operation may be retried and, if unsuccessful, service may be required.

End of Form

The end-of-form light turns on and the machine stops when an end-of-form condition occurs.

If an end-of-form occurs during a skip or while spacing within the last form in the printer, the operator should single-cycle print until the next skip to a new form occurs.

Form Check

The form check light is turned on for any of the following conditions.

1. Forms not feeding properly through the forms tractor
2. Forms guide plate open
3. Carriage control tape not installed
4. Carriage brush assembly open
5. Feed clutch manual control not properly positioned
6. Carriage stop key pressed.

Sync Check

The sync check light turns on when the chain is not in synchronization with the compare circuits for the printer. The timing is automatically corrected. Pressing the check reset key turns off this light.

CARRIAGE CONTROLS

Carriage Space

Pressing the carriage space key advances the carriage form one line space if the clutch is engaged. On some systems, this key is operable only when the printer is in a not-ready condition.

Carriage Restore

Pressing the carriage restore key positions the carriage at channel 1 (home position) of the carriage tape, or at linecount 1 of the forms line counter. If the carriage feed clutch is disengaged, the form does not move.

NOTE: This key must not be pressed when the printer is printing. Printer operation is unpredictable; carriage runaway may occur.

Carriage Stop

Pressing The carriage stop key stops the carriage operation and turns on the form check light. The form may need to be realigned with the program. Press the check reset key to turn off the form check light.

INDICATOR PANEL LIGHTS

The indicator panel (Figure 2-12), located below the manual feed clutch control, can enable the operator to easily locate and rectify common trouble sources.

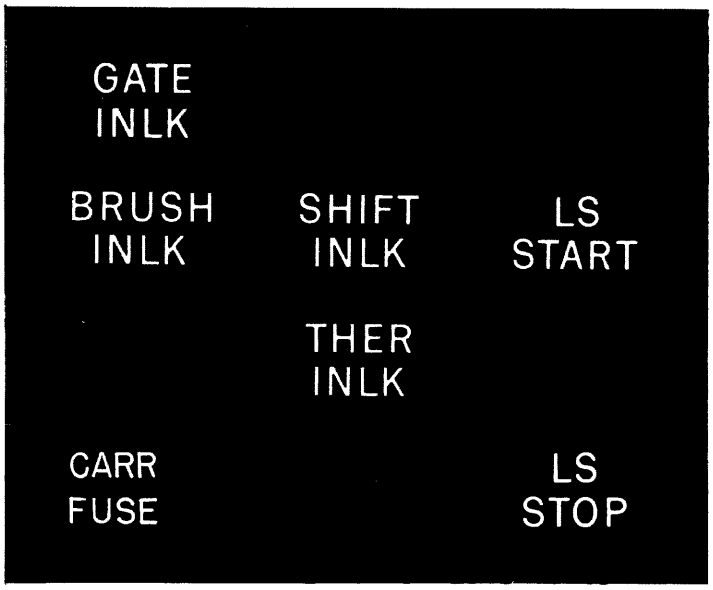


Figure 2-12. Indicator Panel

Gate Inlk

The gate interlock light indicates that the print unit is not in position. The print-unit release lever locks this unit in position.

Brush Inlk

The brush interlock light indicates that the carriage tape brushes are not latched in position for operation.

Carriage Fuse

The carriage fuse light indicates that a fuse in the carriage circuitry has burned out. Contact your service representative.

Shift Inlk

The shift interlock light indicates that the manual feed clutch control is not positioned properly.

Thermal Inlk

The thermal interlock light indicates that a fuse has burned out and that service is required.

LS Start

The low-speed start light indicates that a low-speed skip or line spacing has been initiated.

LS Stop

The low-speed stop light indicates that a low-speed skip stop has been initiated. The light is also on when the carriage is not in motion.

MANUAL CONTROLS

The manual controls are shown in Figures 2-13 through 2-16.

Feed Clutch

The feed clutch controls the carriage tape drive and form feeding mechanism, and selects the 6- or 8-line-per-inch spacing. When the feed clutch is set to neutral, automatic form feeding cannot occur.

Paper Advance

The paper advance knob positions the form vertically. The feed clutch must be disengaged.

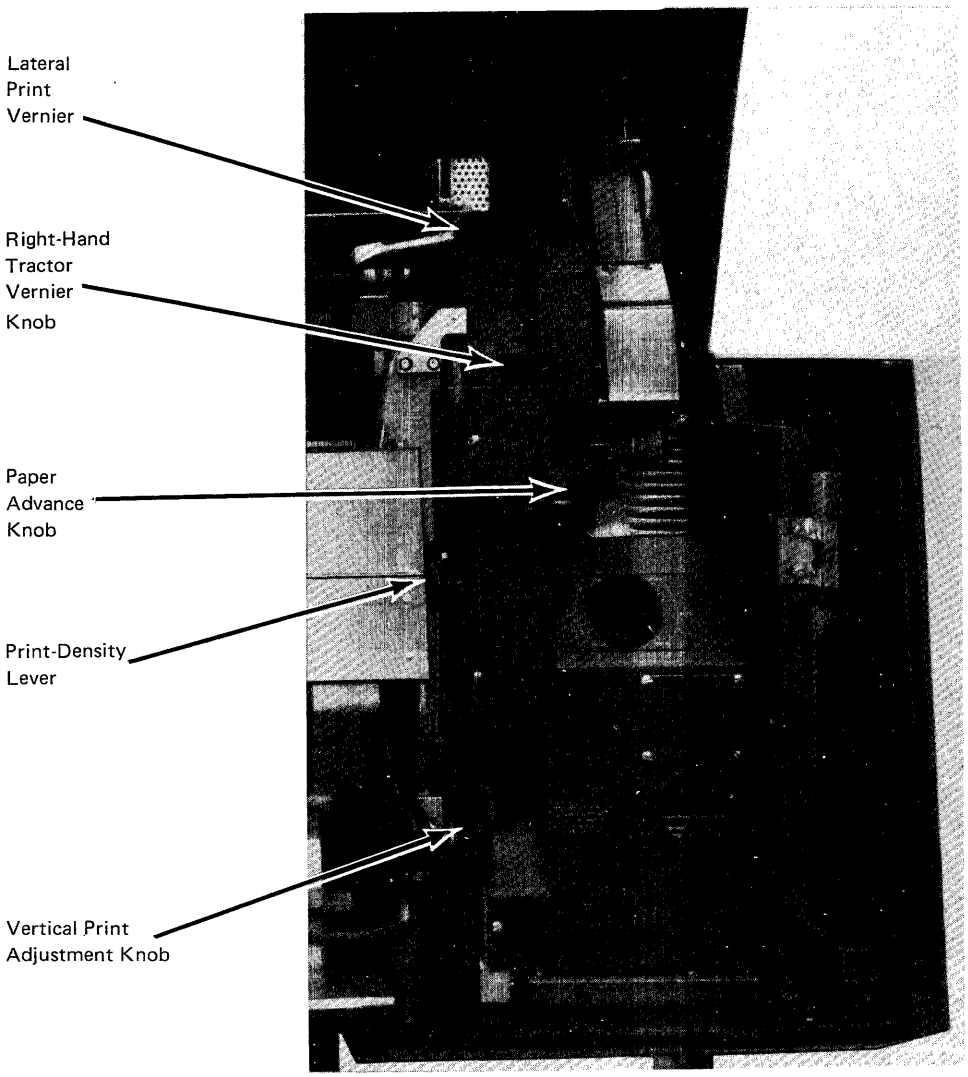


Figure 2-13. Manual Controls

Vertical Print Adjustment

The vertical print adjustment knob controls the fine spacing adjustment of forms at the print line. The carriage tape is not affected by this adjustment.

Print Unit Release

The print-unit release lever unlocks the print unit to allow it to be swung open to provide access to the form transport area (Figure 2-14).

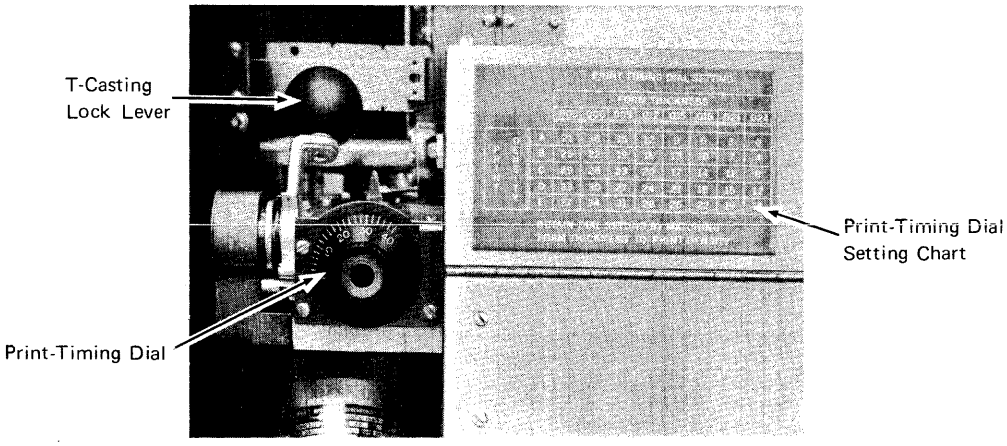


Figure 2-14. Print Unit Release Lever

Print-Line Indicator and Ribbon Shield

The print-line indicator and ribbon shield (Figure 2-15) pivot with the ribbon mechanism when the print unit is opened. This assembly may be unlatched from the print unit and pivoted independently.

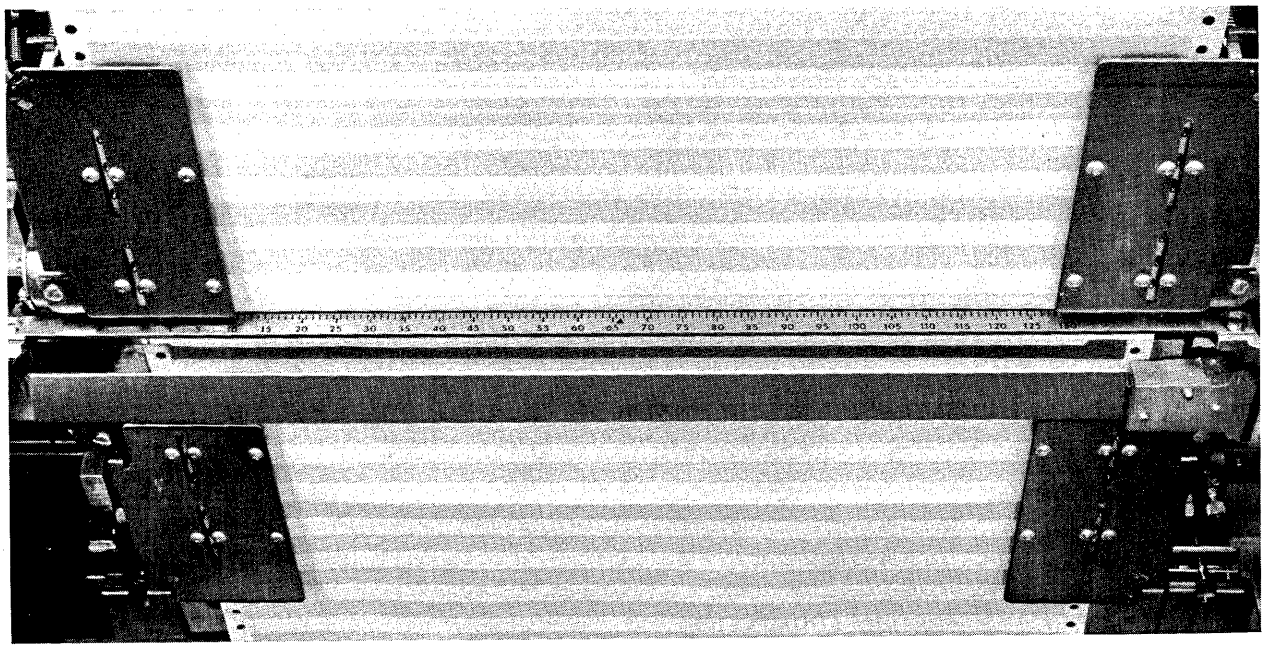


Figure 2-15. Print Line Indicator and Ribbon Shield

Lateral Print Adjustment Lever

This lever allows for horizontal positioning of the printing mechanism. When the lever is raised, the print mechanism unlocks and can be positioned horizontally within its 2.4 in. (61mm) travel limit.

Lateral Print Vernier Knob

The lateral print vernier controls the movement of the print mechanism. Movement is up to 1/2 in. (13mm).

RH Tractor Vernier

The right-hand tractor vernier knob controls fine adjustments in paper tension. Lateral movement is up to 1/2 in. (13mm).

Tractor Slide Bars

The forms tractors are mounted on two tractor slide-bars, upper and lower. To facilitate positioning the forms tractors, notches are provided in the tractor slide-bar.

The left tractor is locked in place by a spring-loaded latch in one of the nine notches located 1 in. (25,4mm) apart on the tractor slide-bar. The third notch from the left end is the normal location for most applications.

The first notch is used for forms from 5 1/2 to 18 3/4 in. (140 to 475mm) wide. When this notch is used, the lateral movement of the print unit is limited to .4 in. (10mm).

The second notch is used for forms from 4 1/2 to 17 3/4 in. (115 to 450 mm) in width. When this notch is used, the lateral movement of the print unit is limited to 1.4 in. (35mm).

The third notch is used for forms from 3 1/2 to 16 3/4 in. (90 to 425mm) wide. When this notch, or one of the notches 4 through 9 is used, a full lateral print-unit movement of 2.4 in. (61mm) is possible.

The ninth (last) notch can be used for forms from 3 1/2 to 10 3/4 in. (90 to 273mm) wide. When this notch is used, the first print position is No. 38.

The right-hand tractor is locked in place by spring-locked pins snapped into any one of 27 holes, located 1/2 in. (13mm) apart on the tractor slide-bar.

The movement of the tractor slide-bar is controlled by the right-hand tractor vernier.

Print Density Lever

The print-hammer unit accommodates different thicknesses of forms. The print-density lever provides a vernier control for print impression. When this lever is set at position E, print impression is lightest.

When this lever is set to position A, print impression is darkest. Between these two settings are intermediat settings. Position C is considered the normal setting. This lever moves the type chain closer to, or farther from, the hammer unit (See Figure 2-13).

Print-Timing Dial

A movable dial is set for fine adjustment of print quality (See Figure 2-14).

The proper dial setting is obtained from the print-timing dial chart (Figure 2-14) located on the ribbon cover. The setting of the print-density lever, in conjunction with the thickness of the form, gives a nominal setting of the print-timing dial.

The setting from the chart can be adjusted to a finer degree by the operator. For finer setting of the timing dial, turn the timing dial clockwise until the left side of the characters appears to be cut off. Then rotate the dial counterclockwise until the right side of the printing appears to be cut off. The optimum setting of the print-timing dial is halfway between the two readings.

Form-Thickness Lever

The form-thickness lever (Figure 2-16) is located at the right-hand end of the ribbon cover. This lever permits manual adjustment for the various forms (single or multiple copy) thicknesses. The adjustment range is from 0.003 in. (0,08mm) minimum to 0.019 in. (0,48mm) maximum graduated in increments of 0.004 in. (0,10mm).

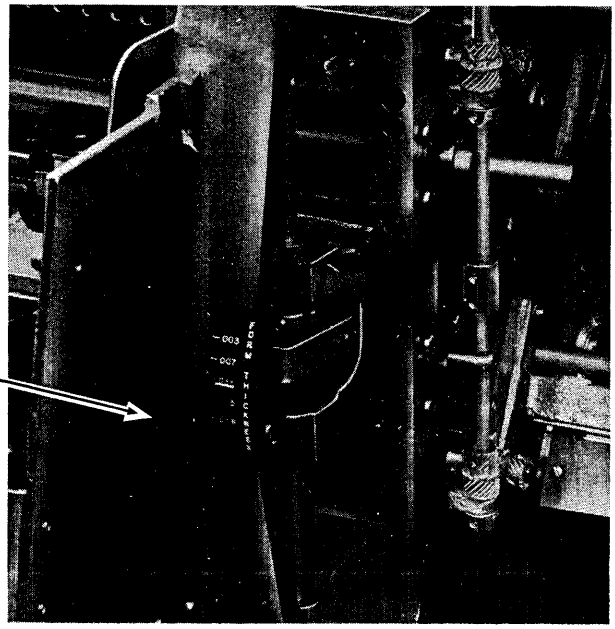


Figure 2-16. Forms Thickness Lever

FORMS CARRIAGE CONTROL

For tape control, each application has a control tape (Figure 2-17) corresponding in length to the length of one or more forms. This tape is punched with holes to stop the form when it reaches a pred termined position.

With the carriage-control tape, the carriage accommodates continuous forms, up to a maximum of 22 in. (559mm) in length (at 6 lines per in.). The minimum length is 1 in. (25,4mm) or 1 1/2 in. (38,1mm) at 8 or 6 lines per in. respectively.

Control Tape

The carriage-control tape (See Figure 2-7) has 12 columns indicated by vertical lines. These positions are called channels. Holes can be punched in each channel throughout the length of the tape. A maximum of 132 lines can be used to control a form, although for convenience the tape blanks are slightly longer. Horizontal lines are spaced six to the in. (25,4mm) for the entire length of the tape. Round holes in the center of the tape are prepunched for the pin-feed drive that advances the tape in synchronism with the movement of a printed form through the carriage.

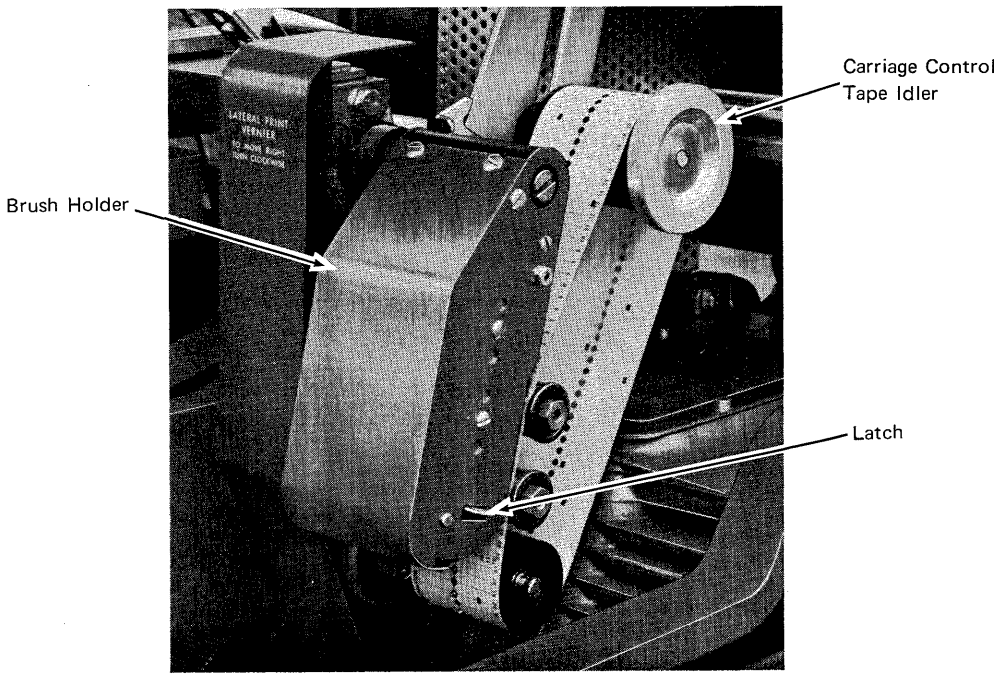


Figure 2-17. Forms Carriage Control

PUNCHING THE TAPE

Six-Lines-per-Inch Spacing

A small compact punch (Figure 2-18) is provided for punching the tape. The tape is first marked in the channels in which the holes are to be punched. This can be easily done by laying the tape beside the left edge of the form it is to control, with the top line (immediately under the glue portion) even with the top edge of the form. Then a mark is made in the first channel, on the line that correspon s to the first printing line of the form. Additional marks are made in the appropriate channels for each of the other skip stops, and for the overflow signal required for the form.

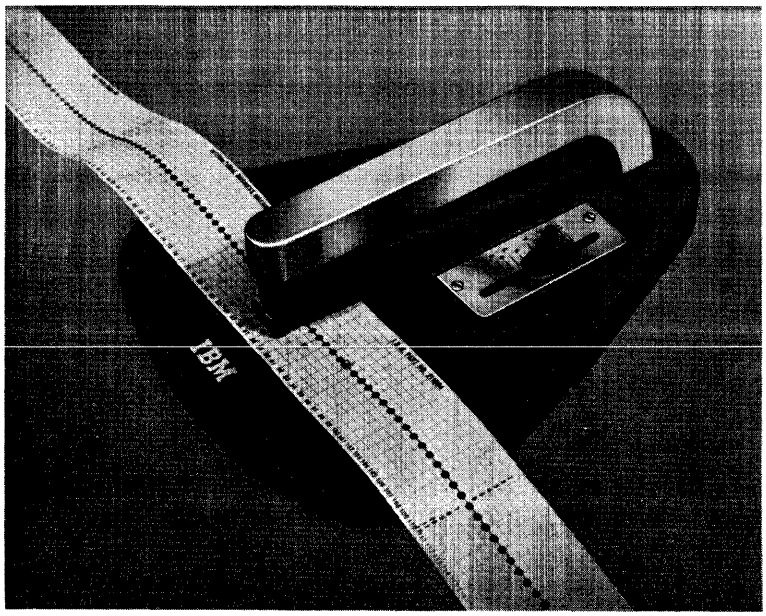


Figure 2-18. Tape Punch

The marking for one form should be repeated as many times as the length of the tape (22 in. or 559mm) allows. When the tape controls several forms in one revolution through the sensing mechanism, the life of the tape is increased. Finally, the line correspondi g to the bottom edge of the last form should be marked for cutting after the tape is punched.

The tape is inserted in the punch by placing the line to be punched over the guide line on the base of the punch and placing the center feed-holes of the tape over the pins projecting from the base. The indicator slide is then moved until the arrow points to the number of the channel to be punched. Pressing on the top of the punch, toward the back, cuts a rectangular hole at the intersection of a vertical and horizontal line in the required channel of the tape. The tape should <u>never</u> be punched in more than one channel on the same line. Holes in the same channel should not be spaced closer than eight lines apart. After the tape is punched, it is cut and looped into a belt. The bottom end is glued to the top section marked <u>glue</u>, with the bottom line coinciding with the first line. Before the tape is glued, the glaze on the tape should be removed with an ink eraser; if this is not done, the tape ends may separate. The center feed-holes should coincide when the two ends of the tape are glued together.

The last hole punched in the tape should be at least four lines from the cut edge, because approximately the last half in. (12,7mm) of the tape overlaps the glue section when the two ends are spliced. If a hole must be punched lower than four lines from the bottom of

the form, place the tape with the top line (immediately under the glue portion) four lines lower than the top edge of the form, before marking the channels. To compensate for the loss, cut the tape four lines lower than the bottom edge of the form.

Eight-Lines-per-Inch Spacing

Each line on the tape always equals one line on the form, regardless whether the form is six to eight lines per in. (25,4mm). For a document printed eight lines to the in., every 1/8 in. (3,2mm) on the form represents one line on the tape.

Carriage Tape Brushes

Reading brushes (Figure 2-19) sense holes in the carriage-control tape. A small contact roll is used for the brushes.

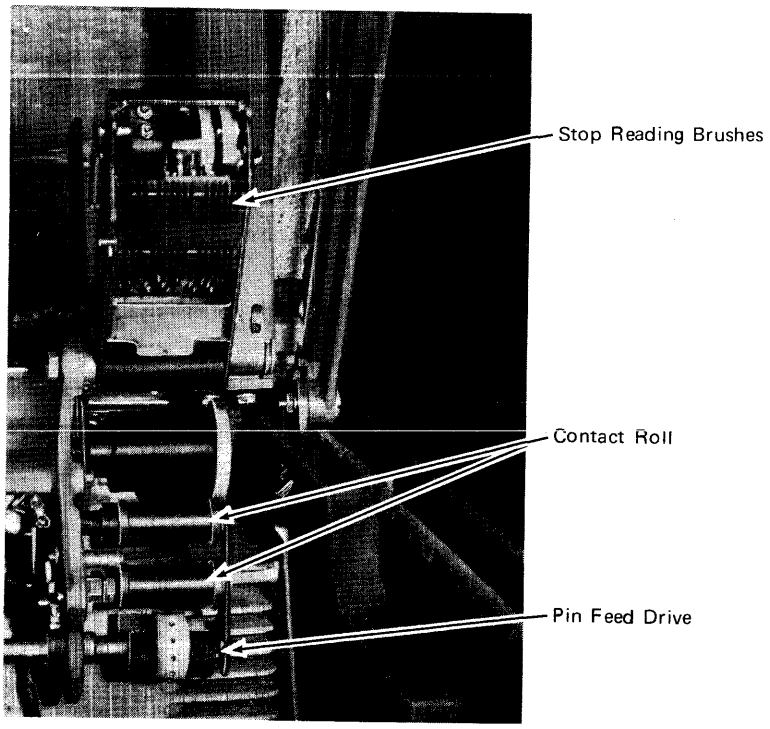


Figure 2-19. Carriage Tape Brushes

ACOUSTICAL DAMPENER

The acoustical dampener is a nylon and bronze brush on the print-unit frame. The dampener provides a drag on single-part forms to dampen high-frequency vibration of the paper. The brush is hinged so that it can be rotated out of the way when not needed. To prevent damage to single-part forms, rotate the brush away from the paper when the print unit is moved horizontally for alignment.

OPERATOR PROCEDURES

CARRIAGE TAPE INSERTION

1. Raise the printer cover.
2. Turn the feed clutch to neutral.

3. Press the latch on the side of the brush holder, and raise the assembly.
4. With the printing on the outside of the tape loop, place the loop over the pin-feed drive wheel so that the pins engage the holes in the tape. Be certain that the line position numbers are on the right side of the tape loop, as seen from the front of the printer.
5. Place the other end of the loop around the adjustable carriage-control tape idler.
6. Adjust the idler by loosening the locking knob and moving the idler in its track. No noticeable slack should be in the tape, but the tape should not be under tension. Test the tape by pressing the sides of the loop together. There should be some give. If the tape is too tight, the pin-feed holes will be damaged. Be sure to retighten the locking knob on the idler.
7. Lower the brush assembly. A click can be heard when the latch engages.
8. Press the restore key. When the tape has returned to the home (channel 1 or line count 1) position, engage the feed clutch.
9. Close the printer cover.

RIBBON CHANGING

To change the ribbon (Figure 2-20) on the 2922-2:

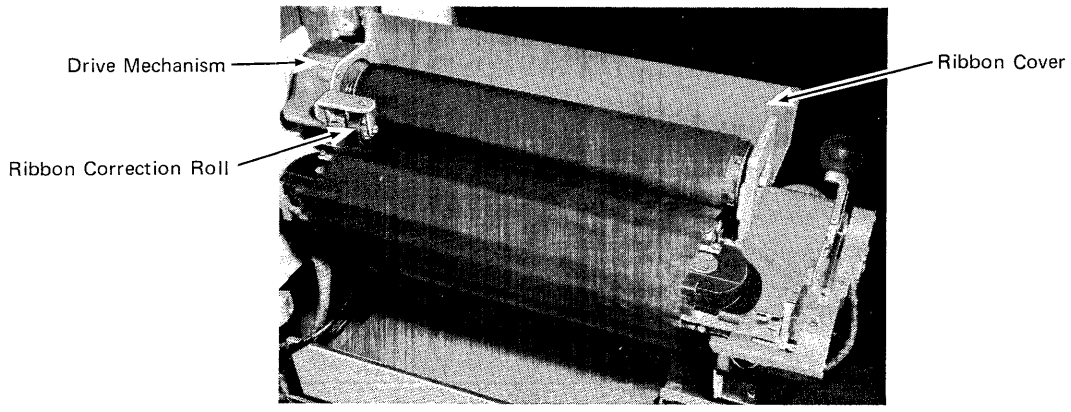


Figure 2-20. Ribbon Mechanism

1. Raise the printer cover.
2. Pull back and unlock the print unit release lever. Swing out the print unit.
3. Open the top ribbon cover.
4. Unlatch the print-line indicator ribbon shield and swing it away from the ribbon, against the forms area.
5. Push the top ribbon roll to the right (hinged side of print unit), lift out the left end of the ribbon roll, and remove the roll from the drive end of the mechanism. (Gloves may be provided with the ribbon.)
6. On printers without the auxliary ribbon feeding, slip the ribbon from under the ribbon correction roller.
7. To remove the bottom roll, press the ribbon roll to the right, lower the left end of the ribbon roll, and remove it from the mechanism.
8. When replacing the ribbon in the machine, hand-tighten the ribbon to remove slack from in front of the printing mechanism.

Ribbons are available in 11 inch (279,4mm) widths, in addition to the standard 14 in. (355,6mm). The ribbon width lever (Figure 2-21) can adjust the ribbon-feed mechanism to accomodate the various ribbon widths.

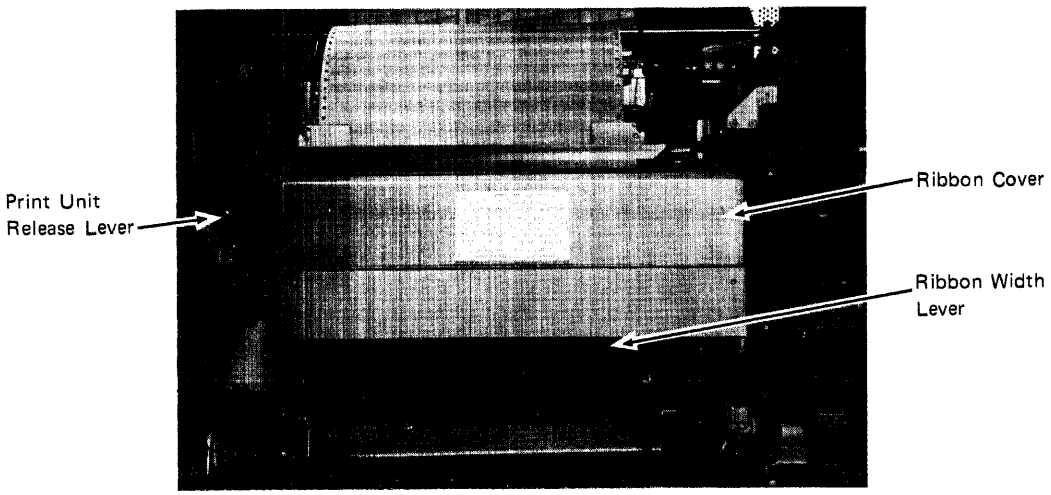


Figure 2-21. Front of Printer (Cover Raised)

NOTE: When installing a new ribbon in the printer, always load the full ribbon spool on the bottom spindle to assure proper ribbon skew on the first winding of the ribbon (Figure 2-22).

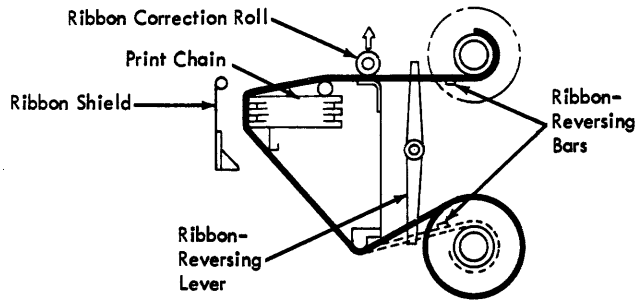


Figure 2-22. Installing a New Ribbon

FORMS INSERTION

1. Raise the front cover of the printer to gain access to the print unit and forms area.
2. Turn the clutch to neutral.
3. Unlock and swing back the print unit by pulling the print-unit release lever toward you.
4. Set both the left-hand forms tractors slightly to the left of the first printing position. Pull the tractor until until it latches in the appropriate notch (Figure 2-23).
5. Open the left-hand tractor covers and place the forms over the pins. Close the covers.
6. Open both the right-hand tractor covers.
7. Move the right-hand tractors to the desired location to line up the right side of the forms. Pull out the tractor pin latch, and slide the tractor until the pin snaps into the appropriate position (See Figure 2-15).
8. Place the forms over the tractor feed-pins and close the tractor covers.
9. Tighten the tension on the form, using the right-hand tractor vernier.

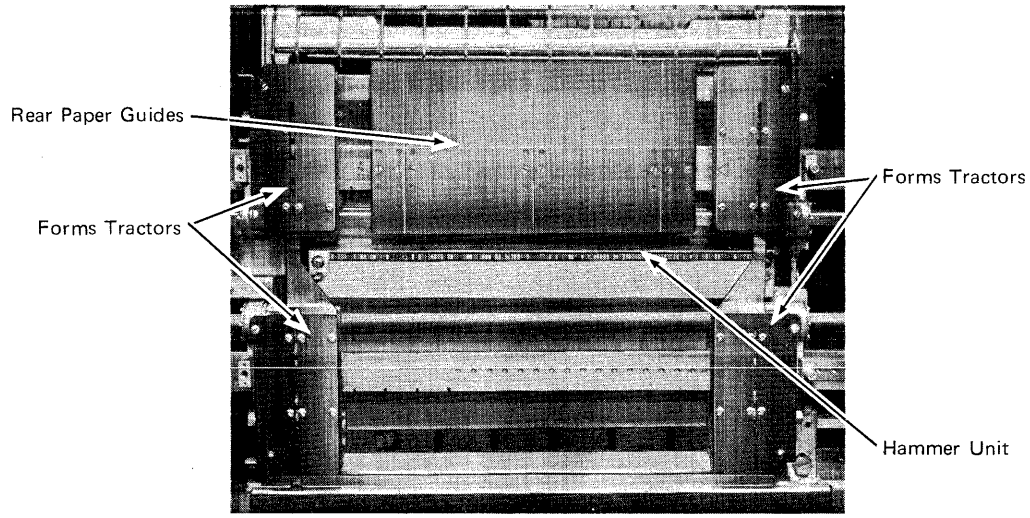


Figure 2-23. Forms Tractor

10. To position the form, turn the paper advance knob until the block, line, or area on which the first line of print is to occur is just visible above the ribbon guide bar. Align the desired hammer position to the form with the lateral print alignment lever and vernier. Observe the relationship to the form of the markings on the ribbon guide bar. Now, turn the paper advance knob backward three line spaces (if in six-line neutral, or or four line spaces if in eight-line neutral). The form is now properly positioned.
11. Close and lock the print unit. Be sure to push the print-unit release lever as far back as it will go.
12. Restore the carriage tape to the first printing position by pressing the carriage restore button.
13. Set the feed clutch to DRIVE. Set it for either six or eight lines per in. (25,4mm) depending on the form to be printed.
14. Close the cover of the printer.
15. Position the paper supply so that the forms feed straight up into the machine.
16. When the printing begins, operator attention is required behind the printer. The first form must be guided between the forms stacker-guide and the machine. Then, the first form must be adjusted in the stacker so they fold flatly. See "Forms Stacking".

FORMS STACKING

The forms stacking mechanism of the 2922-2 printer consists of two major parts: a set of stacker rolls and a paper guide. As the forms are printed, they move upward out of the print area, over the top of the printer, and downward into the stacker at the back of the machine. Here the forms are refolded into a flat stack.

As the forms enter the stacker, they pass between the powered stacker rolls and a paper-tension device.

The paper guide arrangement is used to help stack the paper evenly. Satisfactory stacking under power depends on the care with which the operator positions or adjusts and supervises the operation of the forms-feeding and stacking mechanisms.

Among the common operator actions recommended for good forms feeding and stacking are the following.

1. Remove the blank forms from their shipping cartons before installing them in the printer. Air suction produced as the forms are pulled from the carton increases the drag on the paper sufficiently to tear or distort the pin feed holes at the edges of the forms.
2. Be careful when adjusting the forms tractors. Never adjust the horizontal tractor vernier so that the form is streched too tightly. This can tear or distort the pin-feed holes in the forms.

Position the sliding paper guide (Figure 2-24) up or down as required to obtain the best folding and stacking condition. Use the paper-guide control knob to raise or lower the paper guide. This knob slides along a printed scale graduated from 0 through 6 for convenient operator reference. The lower edge of the paper guide assists in folding the paper; thus, as the pile of stacked forms rises, raise the guide correspondingly.

Gravity stacking is obtained by disengaging the paper-tension device from the powered stacker rolls. Use the stacker spring lift bar (Figure 2-24) to move the circular tension springs away from the stacker rolls.

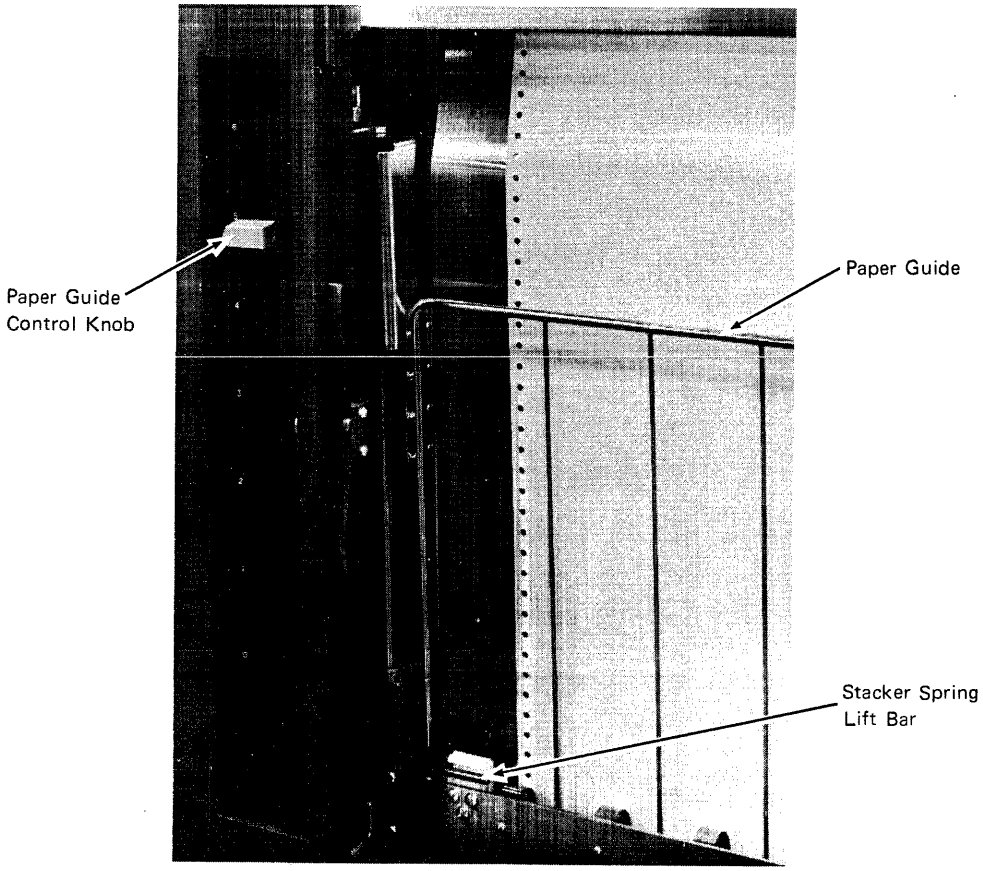


Figure 2-24. Forms Stacker

PRINT QUALITY REQUIREMENTS

When the 2922-2 printer is used in optical character recognition (OCR) applications, correct machine setup is important. The operator must take certain precautions to ensure acceptable print quality. These include:

1. Using recommended ribbon and paper-weight combinations.
2. Supervising and adjusting the print density as required.
3. Replacing the ribbon as required.
4. Cleaning the type faces.

RIBBONS AND PAPER

For applications in which OCR is used extensively, the IBM OCR ribbon (part 414486) or equivalent, is recommended. For ess extensive OCR applications, the IBM general-purpose ribbon (part 419098), or equivalent may be used, provided a 20- to 24-pound (75-to 90 gram per square meter) bond paper is used. Other weights of paper require the IBM OCR ribbon or equivalent.

RIBBON LIFE

Ribbon life depends upon the amount of ribbon usage. Because new ribbons contain more ink than used ones, the initial print density (darkness of impression) is heavy. The more the ribbon is used, the less ink it contains and the lighter the print density becomes. Therefore, the operator should check the print density at the beginning of the job and periodically throughout the run, and adjust the print-density control as required to maintain the best print quality.

When printing becomes so light that further adjustment of the print-density control produces no appreciable improvement, replace the ribbon. When an OCR ribbon is used, the condition does not occur usually until 250,000 lines have been printed.

Figures 2-25 and 2-26 illustrate acceptable and unacceptable OCR printing related to ribbon life. Lines one and two show acceptable printing, and lines three and four show marginal printing. Lines five and six represent unacceptable printing: both inadequate ink coverage and insufficient stroke width.

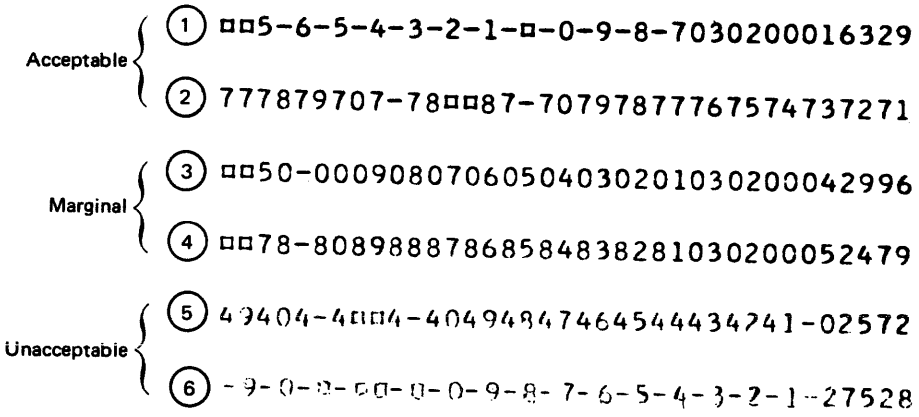


Figure 2-25. Ribbon Life Related to OCR Print Quality

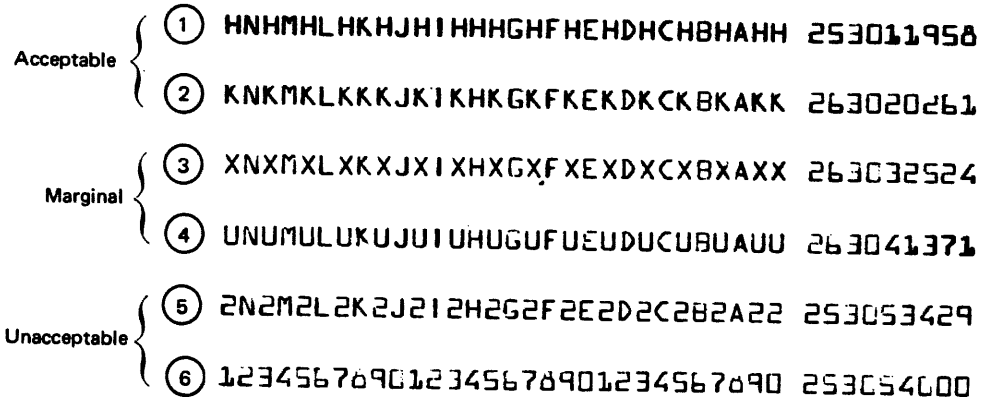


Figure 2-26. Ribbon Life Related to OCR Print Quality (1428 Font)

CHARACTER STROKE WIDTH

Except for the period or decimal, and comma, the character-stroke width for OCR should measure 0.010-0.018 in. (0,25-0,46mm). The samples of printing shown in this publication can be used for visual comparison. These samples show acceptable as well as unacceptable stroke widths.

Figure 2-27 shows two samples of ideal OCR printing with stroke widths of 0.013 in. (0,33mm) and 0,011 in. (0,28mm). A sample with stroke widths of 0.008 in. (0,20mm) is included for comparison.

PRINT DENSITY AND FORMS-THICKNESS ADJUSTMENTS

In applications in which print quality is critical, correct print-density and forms-thickness adjustment are important. The forms-thickness adjustment must correspond to the thickness of the forms used. If this thickness cannot be measured directly, secure this information from the paper supplier. The thickness of 20- to 24-pound (75-90-gram per square meter) paper is about 0.004 in. (0,10mm), and continous forms card stock is about 0.007 in. (0,18mm). The forms tractors should be adjusted to hold the forms as taut as possible without bursting or tearing the forms.

Because each printing format causes unique ribbon wear, the operator must experiment with a print-density adjustment schedule that provides optimum results for his particular application. A good quality OCR ribbon has a usage expectancy (for OCR printing) of 250,000 lines. As a starting point, the following schedule of print-density settings for the degree of ribbon wear is recommended:

Ribbon Usage	Print Density Setting
New to 50,000 lines	D
50,000 to 100,000 lines	C
100,000 to 150,000 lines	B
150,000 to 250,000 lines	A

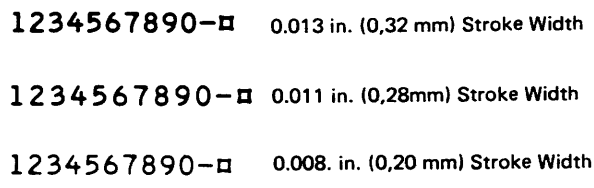


Figure 2-27. Stroke-Width Variations

Figure 2-28 shows print-density conditions (enlarged) that can be corrected by the 2922-2 operator.

Adjust the print density control lever to get the best impression with the thickness of the forms being used. Then set the print-timing dial (Figure 2-14) for that thickness, as prescribed by the chart on the print unit (also in Figure 2-14). If the left edges of the characters are cut off, turn the print-timing dial counterclockwise until they

reappear. If the right edges are cut off, turn the dial clockwise. (Refer to Figure 2-28.)

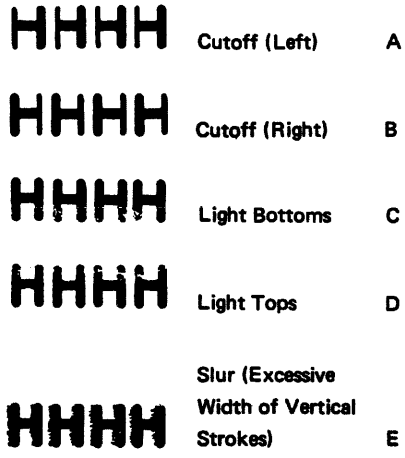


Figure 2-28. Print Conditions (Operator Adjustable)

PRINT QUALITY TEST PROCEDURE

To detect a maladjustment that affects print quality and to find the optimum print-density setting, perform the following procedure periodically before actually printing documents to be used for optical character reading.

1. Install ribbon to be used.
2. Insert documents in the form-feeding mechanism set the printer controls for the thickness of the document.
3. With the print-density control set at E, print a few lines of H's in all positions to check for fading across the print line (characters blacker on one side of the form than those on the other side). If fading occurs, notify a service representative.
4. With the print-density control set at D, continue printing and look for conditions shown in Figure 2-28.
5. Vary the print-density control through several positions to determine the optimum setting for the actual printing of documents. At different impression levels, observe the printing for light density, stroke sections, extraneous ink, and excessive stroke width.

TYPE-FACE CLEANING

For best results, the operator should periodically clean the type faces, first with a vacumn cleaner, then with the type cleaning paper (IBM part 451529, or equivalent). This is a crepe-like paper having a tacky surface. As the hammers drive the cleaning paper against the type (ribbon removed) dirt from the type adheres to its surface. Only one or two sheets are required to clean all the type. Discard the dirty sheets.

NOTE: Under no circumstances should the operator attempt to clean the chain by any means other than the following:

1. Open the print unit and remove the ribbon.
2. Vacumn-clean the cartridge, type faces, and print area.
3. Install the type-cleaning paper on the tractors, just like any other form. Be sure the crepe side is facing you.
4. Position the paper and tractors so that the entire print line is between the preforated margins.
5. Close and lock the print unit.
6. Set the print-density control to C, and the print-timing dial to 17.
7. Put the carriage drive in neutral and press CHECK RESET. (This turns on the check light.)
8. Leave the top cover open and load a suitable program, and press START. The machine starts printing a pattern to print every character in every print position many times.
9. Using the paper advance knob (Figure 2-13), manually space the paper every five or six lines while printing until the type faces are clean, as indicated by a light printed line. Avoid printing more than ten print lines on the same space as this tends to shred the crepe surface of the paper. The shreds that flake off may lodge between the type slugs and damage the chain.
10. Remove and discard the dirty cleaning paper. Reinstall the ribbon.

REFERENCES--OCR

For additional information, the operator should refer to the following manuals:

IBM 1231 and 1232 Optical Mark Page Readers, GA21-9012
IBM 1282 Optical Reader Card Punch, GA24-3106
IBM 1287 Optical Reader, GA21-9064
IBM 1288 Optical Page Reader, GA21-9081
IBM 1418 Optical Character Reader, IBM 1428

Alphameric Optical Reader, GA24-1473
Print Quality Considerations, IBM 1418 and IBM 1428, GA24-1452
OCR Input Preparation Guide, GC20-1686.

INSTRUCTIONS

TRANSFER INSTRUCTION (XIO)

Function	Op Code	DA	FS
Print	D0	4	0
Print & Space Suppress	D0	4	1

The direct or effective address (derived from the B1-D1 fields of the instruction) specifies the leftmost byte of the output data field in main storage. The length of the output field is derived from the B2-D2 fields of the instruction.

The field length specification is limited to binary values equal to or less than 132 for the 2922-2.

At the completion of the Print instruction, the carriage performs an automatic single space unless otherwise directed by the program. The automatic single space does not occur following a Print and Space Suppress instruction.

TEST I/O AND BRANCH INSTRUCTIONS (TIOB)

Function	Op Code	DA	FS
Test Printer Busy	9A	4	0
Test Printer Error	9A	4	1
Test Channel 9	9A	4	2
Test Channel 12	9A	4	3
Test Carriage Busy	9A	4	6

Test Printer Busy

The printer busy condition is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. The printer busy condition and the carriage busy condition (see "Test Carriage Busy") are of the same duration.

Test Printer Error

The Print Check indicator is tested. If this indicator is on, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. With the 2922-3 Printer attached to the Terminal, the print check indicates misfiring of hammers.

Test Channel 9

The carriage channel 9 condition is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. The carriage channel 9 condition is reset by the branch test, and is reset when the carriage moves to or beyond channel 1.

Test Channel 12

The carriage channel 12 condition is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceeds with the updated instruction address. The carriage channel 12 condition is reset by the branch test, and is reset when the carriage moves to or beyond channel 1.

If a print operation, which includes carriage motion after printing, is in progress when either the Test Channel 9 of the Test Channel 12 instructions are encountered in the program sequence, the control unit is interlocked until the completion of the print operation; at this time the test is performed and the program continues.

Test Carriage Busy

The carriage busy condition is tested. If this condition exists, the updated instruction address is replaced by the branch address; otherwise, normal instruction sequencing proceed wtih the updated instruction address. The carriage carriage busy condition and the printer busy condition are of the same duration.

CONTROL I/O INSTRUCTION (CIO)

Function	Op Code	DA	FS
Control Carriage, Immediate Space	9B	4	4
Control Carriage, Immediate Skip	9B	4	5
Control Carriage, Delayed Space	9B	4	6
Control Carriage, Delayed Skip	9B	4	7

A Control Carriage instruction in which bits 28, 29, 30, and 31 are all zero is treated as a no-operation.

The direct or effective detailed specification (derived from the B1-D1 fields of the Control Carriage instruction) indicates the particular line or carriage channel to which the carriage is to advance. Only the four lowest-order bits of the detailed specification (bits 28-31) are required; the remaining higher-order bits are ignored. Carriage channel selection is as follows:

Bits				
28	29	30	31	Specification
0	0	0	1	Single Space or Channel 1
0	0	1	0	Double Space or Channel 2
0	0	1	1	Triple Space or Channel 3
0	1	0	0	Channel 4
0	1	0	1	Channel 5
0	1	1	0	Channel 6
0	1	1	1	Channel 7
1	0	0	0	Channel 8
1	0	0	1	Channel 9
1	0	1	0	Channel 10
1	0	1	1	Channel 11
1	1	0	0	Channel 12

Note: After a skip operation (for example, a skip to channel 1), the form is stopped on the particular line to which it was to advance (for example, line 77). If the control tape contains a punch on line 78 in some other channel (for example, channel 2), the programmer can move the form to this next line by issuing a single-space instruction. However, if a Skip to Channel 2 instruction is issued instead, the skip operation is automatically altered into a single-space instruction and the form advances only to the next line.

CONDITION CODE

XIO Instruction

The condition code is set to 00, 01, or 11 at the time the execution of an XIO instruction for the printer is completed, to indicate that the printer is in the available, working, or not operational state.

The I/O operation and data transfer as specified in an XIO instruction for the printer is initiated only when the printer is in the available state.

The printer is available when the Ready indicator is on and a previous print operation, including carriage motion, is not in progress.

The printer is working or busy during the execution of a previous print operation, including carriage motion after printing.

The printer is not operational when the Ready indicator is not on.

CIO Instruction

The condition code is set to 00, 01, or 11 at the time the execution of a Control Carriage instruction is completed, to indicate that the carriage is in the available, working, or not operational state.

The carriage operation specified is initiated only when the carriage is in the available state.

The carriage is available when the printer Ready indicator is on if a previous print operation is not in progress.

The carriage is working or busy during the execution of a previous print operation which includes carriage motion after printing, or during the execution of a previous carriage operation.

The carriage is not operational when the printer Ready indicator is not on, or when the carriage control tape is not inserted.

IBM 2152 PRINTER-KEYBOARD (OPTIONAL FEATURE)

INTRODUCTION

The IBM 2152 Printer-Keyboard consists of an IBM Selectric Typewriter mounted on a table that is cable-connected to an attachment in the terminal control unit.

The 2152 Printer-Keyboard is used mainly as an inquiry station. For example, it allows an operator to retrieve information, and to print the information on paper; these inquiries are advantageous when jobs are being run sequentially and the operator needs to know the current working status. The 2152 can also be used for entering certain variable job parameters (such as program decision factors and calculation factors).

The 2152 may also be used as a secondary printer. For example, relatively low frequency messages (such as back-order or re-order information developed during billing or stock control operations) may be printed during a regular run without interfering with the operation of the primary printer of the system. Two separate reports may thus be produced by the same program.

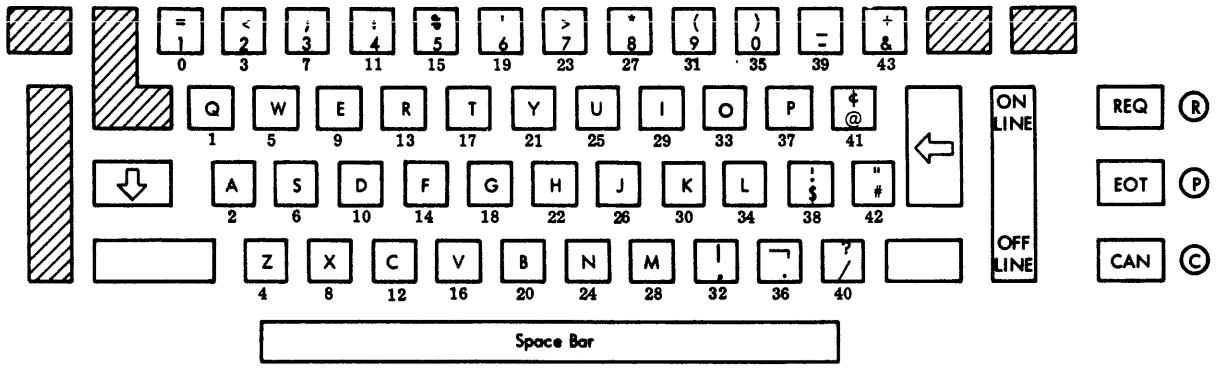
The width of the 2152 print line is 125 characters (maximum) and the vertical spacing is six lines per inch (standard), or eight lines per inch (optional). The 2152 prints at 15.5 characters per second, using a print element of standard System/360 layout, and producing one original print and up to four copies.

The standard equipment includes a pin feed platen (which allows perforated continuous forms to be used) and a paper guide. A friction platen is optional.

GENERAL OPERATION

The 2152 can operate only when the control unit has power on. In addition, if the device is to be used as an inquiry station or secondary printer, the keyboard On Line/Off Line switch (Figure 2-29) must be in the on-line position. In off-line mode, the system can neither respond to inquiries nor print unsolicited messages because the attachment is in a reset state.

In on-line mode, the 2152 is under stored program control, thus, the system may print unsolicited messages as well as replies o inquiries; therefore, the keyboard is locked mechanically during on-line mode. To gain access to the keyboard, the operator must press the request (REQ) key; this action is acknowledged by the keyboard Request (R) light coming on. The program eventually responds to the request by issuing a Read instruction, the keyboard Proceed (P) light comes on and, at the same time, the keyboard is unlocked.



Legend:

- Shift Keys
- Shift Lock Key
- Return Key
- Keys not used (permanently locked)
- (R) Indicator Light, confirming 'Request'
- (P) Indicator Light, meaning 'Proceed'
- (C) Indicator Light, meaning P1 or P2 Check, or Operator Error

NOTE: Numbers beneath keys are for identification

Figure 2-29. IBM 2152 Printer-Keyboard Layout

The operator can now enter a message and may key in as many characters as the field length of the Read instruction permits. The keyboard locks automatically when the allotted field has been filled.

Regardless of whether the field length has been used partially or wholly, the operator must terminate the message by depressing either the End of Transmission (EOT) key, for a valid message, or the Cancel (CAN) key, for an invalid message. When the Cancel key is depressed, a program-testable Cancel indicator is set; however, the message in main storage that has been declared invalid is not destroyed. The program may repeat the operation by issuing another Read instruction.

A special electronic circuit guards against accidental operation of the space bar together with another data key. Should this occur, no data can enter main storage and the keyboard locks. The operator is then forced to cancel the message.

In on-line mode, the program may issue Write instructions to print out messages on the 2152. During write operations, the keyboard is principally locked so that the operator cannot intervene accidentally; however, he can press the Request key at anytime.

INSTRUCTIONS

INSTRUCTION SET

The 2152 is programmed with instructions shown in Figure 2-30.

Format	Type	Op Code	DA	FS	Name
SS	XIO	D0	E	1	Read (from Printer)
SS	XIO	D0	E	2	Write without Carrier Return/Line Feed
SS	XIO	D0	E	3	Write with Carrier Return/ Line Feed
SI	CIO	9B	E	1	Carrier Return/Line Feed
SI	CIO	9B	E	2	Enable Request
SI	CIO	9B	E	3	Disable Request
SI	TIOB	9A	E	0	Test Printer Busy
SI	TIOB	9A	E	1	Test Inquiry Request
SI	TIOB	9A	E	2	Test Any Check
SI	TIOB	9A	E	3	Test Cancel
SI	TIOB	9A	E	4	Test P1-Check
SI	TIOB	9A	E	5	Test P2-Check

Figure 2-30. 2152 Instruction Set

XIO INSTRUCTIONS

The 2152 Transfer I/O (XIO) instructions are executed only when the 2152 is ready. If the device is not ready, the instruction is rejected and a program-testable indication of the reason for the rejection is given in the form of a condition code.

If the 2152 is ready, an XIO instruction is accepted and the device becomes busy for the duration of the operation. The ending of an XIO instruction is indicated to the program by a device end interrupt. The placing of the interrupt request ends the 2152 busy status.

The foregoing characteristics pertain to the Read instruction and both Write instructions, and to the Carrier Return/Line Feed instruction although this is a Control I/O (CIO) instruction. The Read and Write instructions are limited to a field length of 511 bytes; if this rule is violated, a program error stop occurs with a 6 in data register 1. The other programming rules (even boundary, protected area, main storage limits, direct or effective addressing) must also be observed.

CIO INSTRUCTIONS

The 2152 Control I/O (CIO) instructions are executed regardless of the status of the device. If the 2152 is off-line, the execution of a CIO instruction results in no operation. A specification error occurs if the function specification is 0 or greater than 3. The 2152 cannot become busy while a CIO instruction is being executed, and the termination of this instruction is not accompanied by a request for interrupt. However, these characteristics do not apply to the Carrier Return/Line Feed instruction, which is treated as an XIO instruction. CIO instructions contain a B1-D1 field that may be used to state a detailed function specification; however, this field is not used for the 2152. The B1-D1 field must, nevertheless, not be zero.

TIOB INSTRUCTIONS

The 2152 Test I/O and Branch (TIOB) instructions are executed regardless of the status of the device. If the tested condition is present, the program branches to the address defined in the B1-D1 field of the instruction. If the tested condition is not present, or if the 2152 is off-line or not connected, the program continues with the next sequential instruction. TIOB instructions cannot set the 2152 busy status and an interrupt is not requested when such instructions have been completed. The testing of a particular indicator (except Busy and Any Check) causes it to be reset; the Busy indicator is reset when the busy condition ends, and the Any Check indicator is reset when all conditions covered by this indicator have been reset.

READY CONDITION

The 2152 ready condition is a prerequisite for the successful execution of all Read, Write and Carrier Return instructions. The device is ready when all of the following requirements are met:

1. 2152 attached.
2. On-line mode.
3. No end-of-forms conditions.
4. Not busy.
5. 48-volt dc potential present.
6. No P1 or P2 check.

CONDITION CODE

Before a Read, Write, or Carrier Return instruction is executed, the status of the 2152 is examined and a condition code is set.

Condition Code 00 (binary)

Indicates that the 2152 is ready.

Condition Code 01 (binary)

Indicates that the 2152 is busy executing a previously-issued Read, Write, or Carrier Return instruction.

Condition Code 11 (binary)

Indicates that the 2152 is not operational because of one of the following conditions:

1. 2152 not attached (cables disconnected).

2. Off-line mode.

3. 48-volt dc failure.

4. End-of-forms condition.

5. P1 or P2 check present.

The not-operational condition is also indicated by the 2152 light in the attention field of the console; this light comes on when a 2152 XIO instruction is issued while the device is not ready.

2152 INTERRUPTS

The 2152 requests one of two interrupts; device end or, inquiry. The device and interrupt has priority.

Device End Interrupt

The device end interrupt is requested whenever a Read, Write, or Carrier Return instruction has ended. For this interrupt, the device address 'E' and one of the two function specifications are stored in the old PSW to indicate which operation has ended. Identification is as follows:

Indication	DA	FS
Read End	E	1
Write End, or Carrier Return End	E	2

At interrupt time, the program may test on errors and on the type of termination. For example, if an operation was ended by off-line switching, the Cancel indicator is on. A check on the residual count

positions 162 and 163 can show how much of the allotted field length has been used. The device-end interrupt remains pending when the channel mask bit in the PSW is off.

Inquiry Interrupt

This interrupt is identified by device address 'E' and function specification 3. An inquiry interrupt is requested when the operator depresses the Request key. To satisfy the operator's request, the program eventually issues a Read instruction.

However, when the Request key is depressed, the request is only stored; the associated interrupt can occur only when the request is enabled. The request can be enabled either before or any time after operation of the Request key. Inquiry requests can disturb operations in progress and are, therefore, subject to program control via Enable Request and Disable Request instructions. For example, an already enabled request can be disabled, provided that the desired inquiry interrupt has not yet occurred. Nevertheless, the request still remains pending until it is enabled, and the inquiry interrupt can then occur.

Because all interrupts can be disabled via channel mask zero, a special Test instruction (Test Inquiry Request) is provided by which 2152 inquiry requests can be satisifed without interrupt. However, even in these cases a request is recognized only when it has been enabled.

PRINCIPLES OF OPERATION

All 2152 operations are performed in time-sharing mode with other input/output and processing operations.

READ OPERATION

A read operation is executed only when the 2152 is ready. If the device is ready, a Read instruction can be accepted, whereupon the 2152 becomes busy, the keyboard Proceed light comes on, and the keyboard is unlocked. The operator may now key in data at any speed because there are not time-out conditions to be met. The carrier proceeds from left to right, one step at a time (with each key operation), until it reaches the right-hand end position. At this position, the keyboard locks,

except for the space bar. No space character, however, can enter main storage as long as the carrier is positioned at the right-hand end.

To start a new print line, the operator must depress the keyboard Return key. This action causes the carrier to return to the left-hand end, whereupon the form is advanced one step. A lever at the platen allows the operator to select the vertical size of this step, with the standard platen, the line feed ratio can thus be adjusted to either three or six lines per inch.

Note: During carrier return, the keyboard is unlocked; if data keys are operated, data enters main storage and is also printed. Data-key operation during carrier return should be avoided.

The Return key can be operated repeatedly to advance the paper the required number of lines. The operation of the space bar causes one blank character to be transferred for each space-bar operation. The operation of function keys, such as Shift (to upper case), EOT or Cancel, initiates the desired action but no function characters are transferred to main storage.

If the field length is used up during a read operation, the keyboard is locked so that the read-in area cannot be overrun. However, the Proceed light remains on until the operation ends. The message is ended when either EOT or CAN is depreseed; this action can occur at any time during the read operation. EOT is used when the operator declares the message to be valid; the Cancel key is used when the message must be altered (that is, declared invalid).

Note: If the operator accidentally depresses the space bar together with a data key, the keyboard locks and the check light comes on; if this has happened, neither the space nor the character has entered main storage. This situation is termed 'operator error'. A message, during which an operator error occurs, must be terminated via the Cancel key; EOT has no function in this case. The operator error is thus indicated to the program by the testable Cancel indicator.

The Cancel indicator is also set when a read operation is terminated by the switching of the 2152 to off-line mode. The cancelled message is not destroyed; the program may allow for correction by repeating the read operation. If a message is ended by either EOT, Cancel, or off-line switching, the carrier returns automatically to the left-hand margin and line feed occurs. If either a P1 or P2 check occurs or if the end-of-forms contact operates during a read operation, the operation is not disturbed but can continue to the end. However, the 2152 becomes not ready thereafter, and a new Read, Write, or Carrier Return/Line Feed instruction cannot be executed unless the ready status is restored. At the end of a Read instruction, device end interrupt is requested regardless of the method of termination, and a residual length count is stored in positions 162, 163.

The issue of a Read instruction has no influence on the inquiry interrupt bit, which is reset only by execution of the interrupt. The operator can, therefore, rely on the Request light being on until interrupt occurs. This arrangement is necessary because the program may volunteer a Read instruction (for example, in order to allow the operator to enter the current date) immediately after the Request key has been operated. However, the operator may have pressed the Request key for a different purpose and, therefore, his request is not satisfied by the inquiry interrupt or by a program test on the inquiry request.

WRITE OPERATION

Two different write operations can be performed by the stored program; write with carrier return/line feed, and write without carrier return/line feed. In both operations, the keyboard of the 2152 remains locked throughout, and the carrier returns automatically when the right-hand end position is reached. The operations differ only in the action of the carrier when the respective operations terminate. When the write-with-carrier-return operation ends, the carrier returns from the position reached to the left-hand end. When the write-without-carrier-return operation ends, the carrier is left at the position adjacent to the last character that was printed; consequently, any subsequent operation (read or write) begins at this position, with no space between the adjacent texts.

Write operations cannot be stopped by the depression of EOT, or CAN, by P1 or P2 checks, or by and end-of-forms condition. However, if a P1 or P2 check or an end-of-forms condition arises, the 2152 becomes not ready and any new Read, Write, or Carrier Return/Line Feed instruction cannot be executed unless the ready status is restored. Only two keys are operational during write operations; REQ and ON LINE/OFF LINE. Requests are stored during write operations and are executed later (see "Inquiry Interrupt" under "2152 Interrupts".) If the 2152 is switched to off-line mode during a write operation, the operation ends, the Cancel indicator is set, and the carrier returns even if the instruction specified no carrier return. At the end of a write operation, device end interrupt is requested regardless of the method of termination. A residual field length count is stored in positions 162, 163.

CONTROL OPERATIONS

Carrier Return/Line Feed

The Carrier Return/Line Feed instruction is executed only when the 2152 is ready. This instruction causes the carrier to return to the left-hand margin and the form to advance to the next print line; if the carrier is already positioned at this margin, the only action is line feed. The instruction can be issued repeatedly to move the paper a corresponding number of lines. The 2152 is busy during carrier return until the device end interrupt is requested, which occurs when the carrier has reached the left-hand margin.

Enable Request

Requests are disabled automatically after system power on and system reset. If a request is entered via the Request key, the request is stored and remains pending until it is enabled by the Enable instruction. When a pending request is enabled, it will request an inquiry interrupt. A request that has been entered via the Request key can be extinguished by switching the 2152 to off-line however, the Enable instruction remains effective until either Disable Request is issued or the system reset key is operated.

Disable Request

Inquiry interrupts may be inconvenient during certain program runs. In order to suppress these interrupts without suppressing all others, the 2152 request can be disabled, either before or after the Request key has been operated.

If Disable Request is issued before the Request key is depressed, a subsequent Request key operation will nevertheless cause a request to be stored, however, no interrupt action will occur until this stored request is enabled.

If Disable Request is issued after a request has been entered via the request key but before the corresponding inquiry interrupt has been executed, then this inquiry interrupt cannot occur until the request has been enabled. Once a request has been entered, it remains stored unless the 2152 is switched to off-line mode.

TIOB OPERATIONS

Test Inquiry Request

The inquiry request can be tested directly to allow the program to branch when interrupts are disabled via channel mask zero. However, the test can find enabled requests only; the branch then occurs and the request is reset. A test on a request that was entered but not enabled cannot cause a branch.

Test Printer Busy

The printer busy condition can be tested to prevent initiation of an operation that cannot be performed successfully. The 2152 becomes busy when it has accepted a Read, Write, or Carrier Return/Line Feed instruction. The busy condition ends when device end interrupt is requested, which occurs at the end of any of these operations. The busy condition is not reset by the test. The program branches when the 2152 is found to be busy; the program continues with the next sequential instruction when the device is found to be not busy.

Test Cancel

The Cancel indicator is set whenever the keyboard Canel key is depressed during read operations. (The Cancel key is ineffective during write or carrier-return operations.) The indicator is also set whenever the 2152 is switched off-line. The Cancel indicator can be tested to find out whether the previous operation should be repeated. If the indicator is on, the test causes the program to branch and the indicator to be reset. However, the test is not mandatory because the indicator is reset automatically with the next Read, Write, or Carrier Return/Line Feed instruction that is issued after the indicator is set.

Test P1 Check

The P1 check indicates that the attachment to which the 2152 transmits (or from which it receives data) contains an even bit configuration. This parity error can occur during a read or write operation, however,

the operation is not stopped but can continue to completion. The P1 check sets the 2152 not-ready indicator light on the console comes on if a new XIO instruction is issued. The test on the P1 check causes the program to branch, and both lights go out. A new Read, Write, or Carrier Return/Line Feed instruction is rejected with condition code 11 (not operational) as long as the P1 check is present. During operator error, the P1 check is suppressed.

Test On P2 Check

The P2 check indicates that the mechanism of the 2152 has operated a wrong combination of contacts, possibly during a read or write operation. The operation is not, however, stopped but can continue to completion. The P2 check sets the 2152 not-ready status and turns on the 2152 check light. The 2152 indicator light on the console comes on if a new XIO instruction is issued. A new Read, Write, or Carrier Return/Line Feed instruction is rejected with condition code 11 (not operational) as long as the P2 check is present. The test on the P2 check causes the program to branch and the check light, as well as the indicator light, is turned off. The P2 check is operational at all times and is not suppressed when an operator error occurs.

Test Any Check

The Any Check indicator is a summary indication for any of the following indicators: P1 check, P2 check, and Cancel. The program branches when the Any check indicator is on, but the indicator is not reset; it is reset only when all indicators that have caused it are reset.

PRINTER CHARACTER SET

The 2152 can print all characters shown within the heavy frames in Figure 2-31. Bit configurations that pertain to the areas outside these frames are valid EBCDIC configurations but, if used, cause unspecified characters to be printed. Unspecified characters do not cause a check. The printable character set corresponds to the keyboard nomenclature and to the print element layout (Figure 2-32). One blank character is transferred to main storage for each operation of the space bar, except when the carrier is at the right-hand end of its travel; with the carrier in this position, the space bar loses its function (and the rest of the keyboard is mechanically blocked). If

the program intends to produce spaces, it must issue a Write instruction and provide blank characters in those positions where a space is to appear.

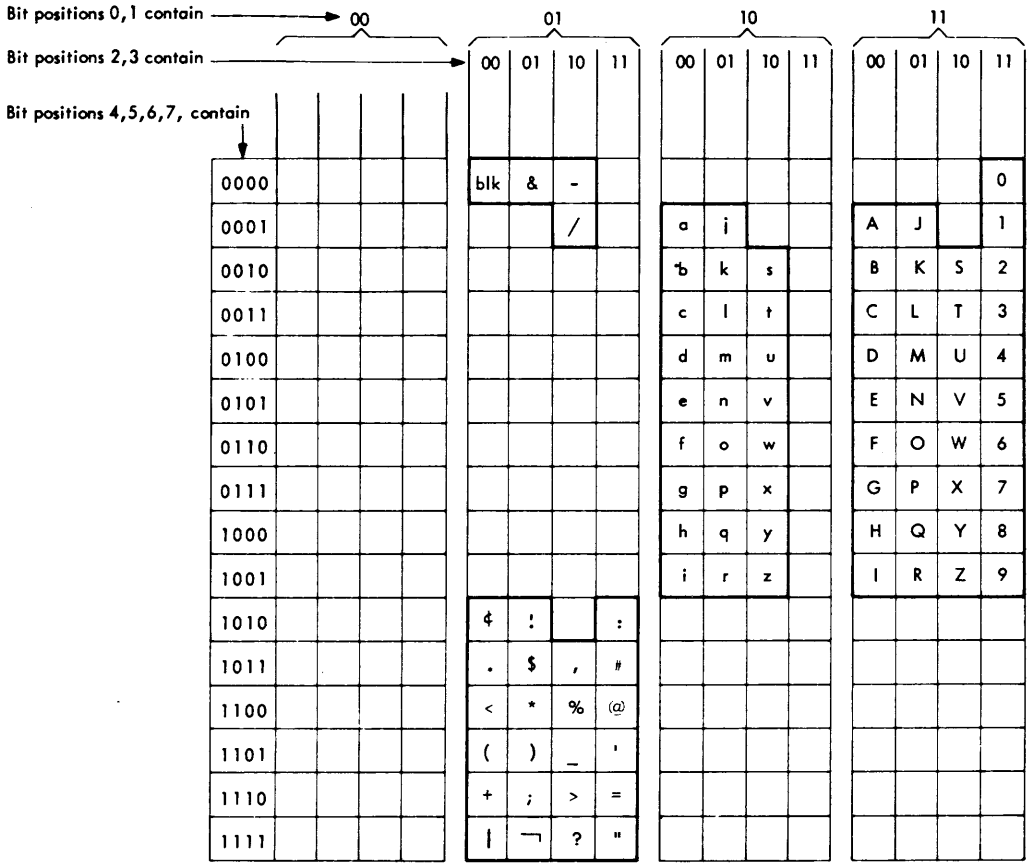


Figure 2-31. 2152 Character Set to EBCDIC

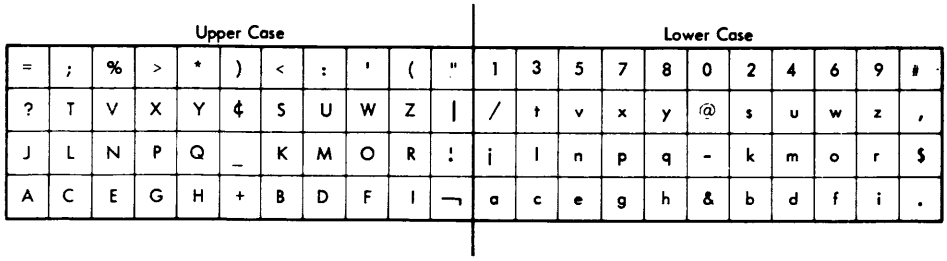


Figure 2-32. Print Element Layout (Type 963)

KEYS AND LIGHTS

Refer to Figure 2-29 for the location of the various keys and lights.

ON LINE/OFF LINE KEY

The toggle-type On Line/Off Line key must be switched to ON LINE if the 2152 is to operate under stored program control. Switching from ON LINE to OFF LINE (or vice versa) causes carrier return with line feed. In off-line mode, the 2152 is not ready and all attachment circuits are in a reset state. Switching to off-line cancels any operation in progress at that time. The key also resets a stored request but, if the request has already caused the inquiry interrupt bit to be stored, an inquiry interrupt will occur eventually. The key has no effect on the enable/disable state of requests.

DATA KEYS

The term 'data keys' refers to all keys by which data is entered. The data keys can be operated only during off-line mode and during on-line mode when a read instruction has started; otherwise, the keyboard is locked.

Note: Although no special skill is required to operate the device, the operator should avoid these errors:

1. Operation of data keys together with the space bar (this causes operator error with keyboard lock).

2. Simultaneous operation of data keys and the Return key (characters may appear in already printed lines on paper but will be stored in adjacent main storage positions).

3. Simultaneous operation of data keys and Shift key (the character on paper will not necessarily correspond to the one in main storage; it may be either the upper or lower case character).

FUNCTION KEYS

Request Key

The Request key must be depressed to notify the program that the operator intends to enter data into main storage. The operation of the Request key causes a request to be stored. This stored request remains ineffective until it is enabled. An enabled request can subsequently initiate inquiry interrupt or it can be tested directly. The request will ultimately cause the issuing of a Read instruction that unlocks the keyboard. A stored request is extinguished only by off-line switching. The Request key is effective only during on-line mode.

EOT Key

The EOT key is used to terminate a read operation when the operator plans no alternations to the message just entered. The key is effective only during read operations when no operator error is present.

Cancel Key

The Cancel key is used to terminate a read operation during which wrong or unsatisfactory data has been entered. The key must be used for termination when the keyboard has locked because of an operator error (simultaneous activation of space bar and data key). Operation of the Cancel key does not destroy the data in main storage. The Cancel key is effective only during read operations.

INDICATOR LIGHTS

Request Light

The Request light comes on when the Request key is operated. The light remains on until either an inquiry interrupt occurs or until the request is tested by the Test Inquiry Request instruction. The light goes

out if the 2152 is switched to off-line mode. Subsequent on-line switching will neither restore the request nor turn on the light.

Proceed Light

The Proceed light comes on whenever the program issues a Read instruction and this instruction is accepted by the 2152. The light goes out only when the operation is terminated via EOT, Cancel or off-line switching.

Check Light

The Check light comes on when either a P1 check, a P2 check, or an operation error occurs. If the light cam on because of a P1 or a P2 check, it goes out when the respective checks are tested. If the Check light was caused by an operator error, it goes off after the Cancel key has been operated.

Attention Light

The Attention light is installed in the attention field of the console. The light comes on whenever an XIO instruction that addresses the 2152 is issued and the device is not ready. The 2152 is not ready when the cables are not attached, the power supply has failed (no 48-bolt dc potential), a P1 or P2 check is present, the 2152 is in off-line mode, or the end-of-forms contact has operated. The light goes out when these conditions are corrected.

PAPER HANDLING

The 2152 prints on continuous forms of 13-5/8 in. (34,6 cm) width. These forms are transported by a pin-feed platen. The horizontal distance from pin to pin is 13-1/8 in. (33,3 cm); the vertical distance from pin to pin is 9/16 in. (1,4 cm). The line feed mechanism moves the paper in increments of either 3 or 6 lines per inch (2,54 cm) as determined by the line adjustment lever at the platen. The maximum print line width is 124 ±1 characters for read operations, therefore the keyboard usually locks when position 124 is reached; however, if the keyboard locks at position 123, then the space bar can still be

operated and thus two space characters can be entered. (Beyond position 125, no addtional space characters can be entered, even though the space bar remains unlocked.) For write operations, the maximum print line width is 124 characters. The margins are fixed in the extreme left-hand and right-hand positions. One original and up to four copies can be printed. A paper guide is provided as standard equipment.

END-OF-FORMS CONTACT

The end-of-forms contact monitors the unprinted paper stock. The contact closes at a point that allows any read or write operation currently in progress to be completed in such a manner that all information is still printed on paper, even if the maximum field length (511 bytes) is used. The operation of the contact does not disturb current operations but the 2152 becomes not ready thereafter.

RIBBON

A single-color ribbon, enclosed in a cartridge, is supplied. Ribbon cartridges can be obtained in a variety of colors through any IBM branch office.

FEATURES

A pin-feed platen with a pin-to-pin width of 13-1/8 in. (333,4mm) is standard. Platens are available for feeding either 6 or 8 lines per inch.

A friction-feed platen (without pins) is optional. A forms stand stacker is available for fanfold paper.

IBM 1442 CARD PUNCH MODEL 5 (OPTIONAL)

INTRODUCTION

The IBM 1442 Card Punch Model 5 (Figure 2-33) provides punched-card output for the 2922 Programmable Terminal. Cards are fed from a 1200-card-capacity hopper, punched serially, and stacked in a 1300-card capacity stacker. Three card-feed cycles move a card from the hopper, through the machine, and into the stacker (Figure 2-34).

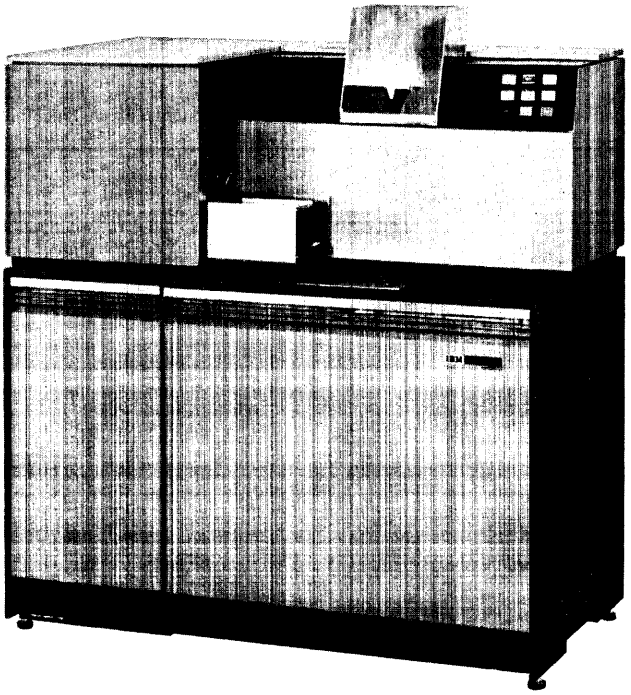


Figure 2-33. IBM 1442 Card Punch Model 5

Cards are punched serially, one column at a time, beginning in column one. Card throughput depends on the number of columns being punched per card and upon the time of receipt of the punch instruction from the control unit. The rated speed of punching is 160 columns per second. The maximum card throughput when punching ten columns is 265 cards per minute. The maximum throughput when punching all 80 columns is 91 cards per minute. Blank columns may be interspersed with the punched columns, but they require the same amount of time.

Punching errors are detected. Each time a hole is punched, a check pulse is developed and compared with a similar pulse from the system.

Dissimilar pulses turn on the punch check light on the 1442-5, and also turn on the punch error indicator latch to allow for program interrogation.

Note: If the punch error indicator is program-tested immediately following punching, the punch light is turned off before it can fully turn on.

The control unit processes other program instructions at the same time it is transferring data to or from an I/O device. For example, when all 80 columns of a card are being punched, the control unit is interlocked with the 1442-5 for approximately 5.5 milliseconds. The remainder of the 1442-5 punching operation is available for other programmable terminal operations. If fewer than 80 columns are being punched, the interlock time is reduced by .067 milliseconds for each column not punched or spaced over.

The time required to punch and feed a card is calculated by the formula T = 6.25 times the number of the last column punched plus 163 milliseconds. If the Punch and Feed instruction is not received by

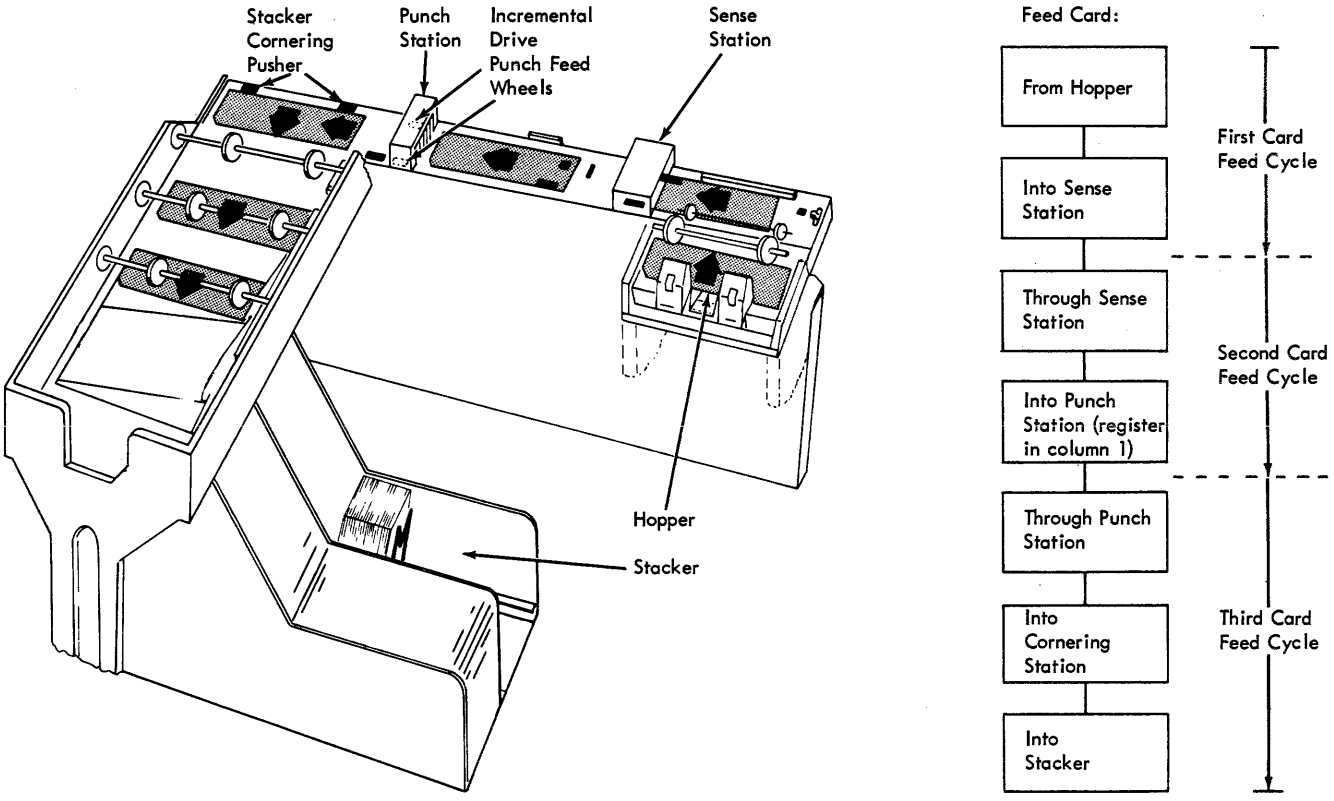


Figure 2-34. Card Feed Path

the end of the 163-millisecond feed cycle, the card punching rate (cards/minute) is reduced accordingly. Figure 2-35 shows the relationship of the number of card columns punched to the maximum number of cards punched per minute.

Last Card Column Punched	Total Punch Cycle Time (milliseconds)	CPM
80	663	91
75	632	95
70	601	100
65	569	105
60	538	112
55	507	118
50	476	126
45	444	135
40	413	145
35	382	157
30	351	171
25	319	188
20	288	208
15	257	233
10	226	265
5	194	309
1	169	355

Figure 2-35. Punch Cycle Times and Throughput Rates

PROGRAMMED OPERATIONS

CONDITION CODES

Three condition codes (Working, Available, and Not-Operational) indicate the status of the 1442-5. The 1442-5 is in a working status during a punch and feed operation; in an available status when ready to receive and execute an instruction; and in a not-operational status when a punch check has occurred and the punch indicator has not been reset, or when the ready light is off for one of the reasons listed under "Check Light."

Data transfer is initiated only when the 1442-5 is in an available condition. When the 1442-5 is in a working or not-operational status, a Punch and Feed instruction results in setting the condition code only, with no action taken in the 1442-5. The control unit proceeds to the next instruction.

TRANSFER INSTRUCTION (XIO)

Function	Op Code	DA	FS
Punch and Feed	D0	3	6

The Punch and Feed transfer command initiates the movement of data from the control unit main storage to the 1442-5. The direct or effective address (derived from the B1-D1 fields of the instruction) specifies the leftmost byte of the output data field in main storage. The length of the output field is derived from the B2-D2 fields of the instruction. The field length specified is restricted to binary values equal to or less than 80.

The punch operation begins with the card registered at column one in the punch station. The data in main storage must be arranged in the exact sequence in which it is to be punched.

When the 1442-5 completes the punch operation, it ejects the card just punched and automatically feeds another card into the punch station. The 1442-5 indicates the end of the punch operation by interrupting the stored program with the channel-end indication.

To obtain maximum card throughput in the 1442-5, the Punch and Feed command must be received prior to the end of the 163-millisecond feed cycle (Figure 2-36).

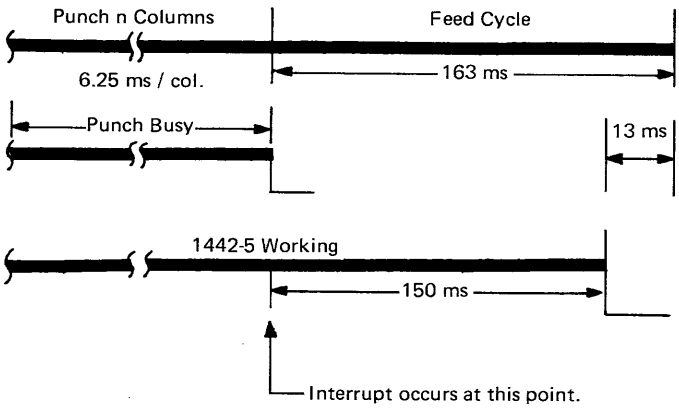


Figure 2-36. Punch and Feed Timings

If the 1442-5 is addressed by a Punch and Feed command when in a not-operational condition, the Punch indicator on the control unit console turns on, the condition code is set to Not-Operational, and the command is terminated.

The Test I/O and Branch instructions direct the control unit to interrogate the 1442-5 for a particular condition (indicator).

Function	Op Code	DA	FS
Test Punch Busy	9A	3	2
Test Punch Error	9A	3	3
Test Feed Error	9A	3	5

Test Punch Busy

The Test Punch Busy instruction tests the punch Busy indicator. When the 1442-5 is in the available condition, this indicator turns on as soon as the control unit sends a Punch and Feed instruction. The Punch Busy indicator turns off when the data transfer portion of the punch operation is completed (last column punched and checked).

If the Punch Busy indicator is on when tested, the program branches to the address specified in the instruction. The channel-end signal is sent to the control unit at the time the Punch Busy indicator is turned off.

Test Punch Error

The Test Punch Error instruction tests the Punch Check indicator. This indicator turns on whenever an error in punching is detected, as further indicated by the punch-check light and by the punch indicator (Attention) on the control unit console.

If the Punch Check indicator is on when tested by the program, the program branches to the address specified in the instruction. The Punch Check indicator is turned off by:

1. The Test Punch Error instruction, or
2. Pressing the I/O Check Reset key on the control unit console,
3. Pressing the System Reset key on the control unit console,
4. Pressing the NPRO key on the 1442-5,
5. Pressing the (program) Load key on the control unit console.

Test Feed Error

The Test Feed Error instruction tests the Feed Check indicator. The Feed Check indicator turns on if a card-feeding malfunction occurs (card jam, failure to feed, etc.) anywhere in the card feed or transport area.

If the Feed Check indicator is on when tested, the program branches to the address specified in the instruction. If a feed cycle is in progress when this instruction is encountered in the program, the program sequence is delayed until the feed cycle is completed, at which time the test is performed.

To avoid delaying the program, the Test Feed Error instruction should be given at the end of the time that the 1442-5 is in a working condition (Figure 2-36). The working condition drops at 150 milliseconds of the 163-millisecond feed cycle.

The condition code is set to 00, 01, or 11 when an XIO instruction is completed to indicate that the 1442-5 card punch is available, working, or not-operational.

The data transfer as specified in an XIO instruction for the 1442-5 is initiated only when the 1442-5 is in the available state.

The 1442 is available when the Ready indicator is on, the Punch Check indicator is not on, and a punch and feed operation is not in progress.

The 1442-5 is working or busy during the execution of a previous 1442-5 punch operation.

The 1442-5 is not-operational when the Ready indicator is off or when the Punch Check or Feed Check indicators are on.

Control I/O Instructions (CIO)

Control I/O instructions to the 1442-5 are treated as No-Op instructions.

Program Timing Considerations

Two basic timings associated with 1442-5 operations must be considered for maximum card throughput and for making full use of the processing time available between card cycles:

1. The 1442-5 is in a working condition for 150 milliseconds following the channel-end interruption (given at the end of the data transfer operation for card punching). If the Test Feed Error instruction is directed to the 1442-5 during this time, the program is delayed until the working condition drops.

2. Thirteen milliseconds elapses between the dropping of the working condition and the end of the feed cycle. A new Punch and Feed instruction must be directed to the 1442-5 prior to the end of the feed cycle if maximum card throughput is to be maintained.

OPERATOR CONTROLS

Figure 2-37 shows the keys and lights for operating the 1442-5.

KEYS

Start Key

This key places the 1442-5 in ready status on initial run-in, provided the following conditions exist:

1. Power on,
2. Card path empty,
3. Cards in hopper,
4. No error lights on.

For initial run-in, pressing the Start key causes two card-feed cycles to register the first card at column one at the punch station. Pressing the Start key after the Stop key has been pressed returns the 1442-5 to ready status again.

Stop Key

This key stops the 1442-5 at the end of a mechanical operation and turns off the ready light. Punch instructions are not accepted until after the Start key is pressed to restore the 1442-5 to ready status.

NPRO Key

This key (Non-Process Runout) runs cards out of the 1442-5 without processing them and resets certain check conditions. When using the NPRO key to run cards out of the machine, these conditions must be satisfied:

1. Hopper empty
2. Any card jams must be cleared.

Three card-feed cycles are sufficient to run all cards out of the machine.

This key also resets any of the Feed Check lights and the Feed Check indicator after the cause of the check has been removed. This key also resets the Punch Check light and indicator (see "Operating Procedures").

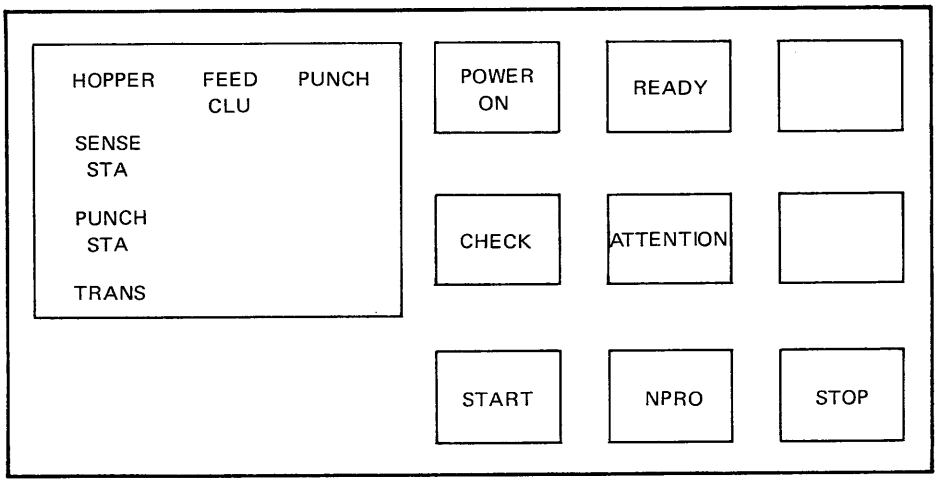


Figure 2-37. Operator Panel

LIGHTS

Power On

This light indicates that ac and dc power is supplied to the 1442-5 circuits.

Ready

This light indicates that the 1442-5 is ready to accept instructions from the control unit. The light is on whenever these conditions are met:

1. Power on
2. Card registered at punch station
3. Cards in hopper
4. Attention light off
5. Hopper, sense station, punch station, transport, and feed clutch lights off
6. Machine not stopped with the Stop key.

The Ready light is turned off by an NPRO sequence or by the operation that feeds the last card from the hopper. If the Ready light is off, the Punch (Attention) indicator on the control unit console turns on when the next Punch and Feed instruction is sent to the 1442-5.

Attention

This light indicates that the chip box is full or not in the machine, that the stacker is full, or that the top cover is open. When the Attention light is on, the Ready light is off.

Check

This light is a general indication that an error condition exists, as further indicated by the following lights on the back-lighted panel. The first five conditions also cause the ready light to turn off.

1. HOPPER--indicates that a card was not fed from the hopper.
2. SENSE STA--indicates a sense station jam.
3. PUNCH STA--indicates a punch station jam.
4. TRANS--indicates a jam in the stacker transport.
5. FEED CLU--indicates that the clutch failed to latch, causing an extra feed cycle.
6. PUNCH--indicates that the punch check pulse did not agree with the punch data pulse. When the light is on, the Punch (Attention) indicator on the control unit console is also on. The Punch light is turned off when the main program tests the Punch Error indicator or when the operator presses I/O Check Reset, System Reset, or Load on the control unit console, or by an NPRO sequence on the 1442-5.

Note: Whenever any one of these lights is on, the Punch indicator on the control unit console turns on immediately. The first five lights indicate a card-feed error condition and can be turned off by an NPRO sequence only.

OPERATING PROCEDURES

MACHINE SETUP

When power is turned on in the programmable terminal control unit, a hopper-check light condition results automatically in the 1442-5 forcing the operator to use the NPRO key to reset the light condition. This ensures that the card path is clear of cards from a previous operation. The procedure for setting up the 1442-5 for operation is:

1. Remove any cards from the hopper.
2. Press the NPRO key and hold it down for a few cycles.
3. Place cards in hopper face down, 9-edge first.
4. Press the Start key. This feeds two cards from the hopper, registers the first card on column 1 in the punch station, and turns on the Ready light.

The Ready light along with the auxiliary Punch On-Line light, located on the 2922-1 console, indicates that the punch is ready for operation. The Punch On-Line light may not be on immediately but comes on when the punch is ready and the BSCA is communicating with a down-line system.

PUNCH OPERATION

Punching may be performed while the BSCA is busy and for a period of 28 seconds after the communications line is disconnected. Any 1442 check, stop condition, or 2922-1 terminal stop condition while punching inhibits the timeout until the unusual condition is cleared. Data to be punched does not have to come from the transmission line. However, any programs requiring punch output must establish communications via the BSCA.

When the punch is ready, it is disabled (as indicated by the Punch On-Line light being off) 23 seconds after the communication line has been disconnected. Any alteration of storage either manually or by IPL further prevents punching. Communication must be established again to continue punch operations.

INDICATOR LIGHT PROCEDURES

Figure 2-38 is a guide whenever card jams or punch errors occur in the 1442-5. The correction procedures in this chart restore the 1442-5 to ready status.

1. Light and 2. Program Indication	Cause	Correction Procedure
1. Hopper--Check 2. Feed Check	Failure to feed card from hopper	1. Lift cards from hopper 2. Press NPRO key for several cycles 3. Remove cards from bottom of deck and place deck back in hopper 4. Remove blank card from stacker 5. Press start key
1. Sense Sta-Check 2. Feed Check	Jam at sense station	Fellow procedure under *Card Jam Removal*
1. Punch Sta-Check 2. Feed Check	Jam at punch station	Follow procedure under *Card Jam Removal*
1. Trans-Check 2. Feed Check	Jam in stacker transport area	Follow procedure under *Card Jam Removal*
1. Feed Clu-Check 2. Feed Check	Feed clutch failed to latch, causing an extra feed cycle	1. Lift cards from hopper 2. Press NPRO key for several cycles 3. Remove blank card from stacker 4. Place card deck back in hopper 5. Press start key
1. Punch-Check 2. Punch-Check	Error detected in punching	Refer to the Operator Procedure Guide for the program being processed
1. Attention 2. Not Operational Condition Code	Chip box full or not in the machine, stacker full, or top cover open	Correct the cause and press the start key
1. Attention-Check 2. Not-Operational Condition Code	Cover open during punching* *If punch-check light is on, treat as punch-check light condition	1. Lift cards from hopper 2. Press NPRO key for several cycles 3. Remove blank card from stacker 4. Place card deck back into hopper 5. Press the start key

Figure 2-38. Indicator Light Procedures

CARD JAM REMOVAL

The Check light, along with a Sense Station, Punch Station, or Transport light, indicates a card jam condition. If the card is jammed at the punch station, repunch the card. Do not use a card saw to remove a jam in the punch station. The punch mechanism could be damaged.

Use the following procedure for removing jammed cards. If the jam is not in the punch station, only the first five steps are necessary.

1. Remove the cards from the hopper.
2. Raise the two top covers and locate the card jam. If the jam is not in the punch station, raise the appropriate plastic card guide and remove the jammed card(s).

If the jam is in the punch station, skip steps 3 through 5 and proceed with steps 6 through 13.

3. Press the NPRO key to run out any cards in the card feed or transport areas.
4. Replace the cards in the hopper.
5. Press the Start key. This again places the 1442-5 in ready status.
6. Loosen the two twist-type fasteners (Figure 2-39) on back cover and lower the cover.
7. With the power on, rotate the punch unit handwheel at least one-half revolution.
8. Press down on the spoon-shaped lever (Figure 2-39) and pull out the card. If card remains jammed, loosen the lower punch guide holding screw and remove the lower punch guide. Do not use a card saw.
9. Replace the lower punch guide if it was removed.
10. Close the covers. Make sure that the cover interlocks are operated.
11. Press the NPRO key to reset the error condition.
12. Place one card in the hopper and feed it through with the Start key and the NPRO key to ensure that the card path is clear.
13. Place cards in the hopper and press the Start key. The 1442-5 is now ready for continued operation.

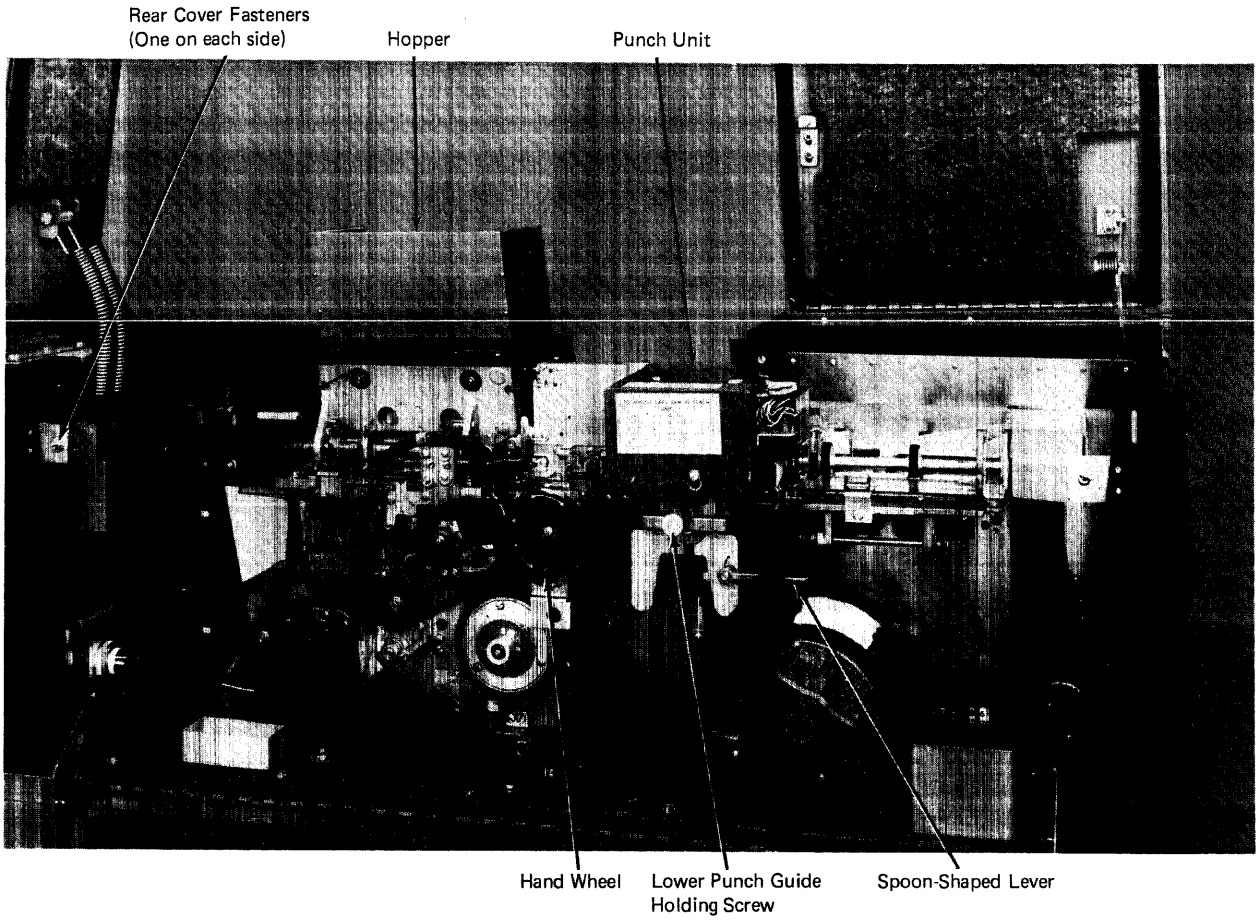


Figure 2-39. 1442-5, Rear View

BINARY SYNCHRONOUS COMMUNICATIONS ADAPTER

INTRODUCTION

Telecommunication, the ability to exchange up-to-date information between remote locations, is a valuable asset in the fast-paced modern business world of industry and commerce. Therefore, powerful tools, such as electronic data processing systems, should not be limited to strictly local input and output. Nowadays, as business activities expand and companies establish branches farther afield, the rapid exchange and processing of vital data is of ever-increasing importance. Thus, more and more emphasis is being placed on teleprocessing, which offers interesting prospects to all users of data processing systems.

The Binary Synchronous Communications Adapter (BSCA) for the 2922 Programmable Terminal is a teleprocessing device which is virtually independent of such restricting factors as the necessity for code translation (due to a limited character set), complex synchronizing procedures, slow transmission speeds, etc.

The BSCA operates in time-sharing mode with other input/output and processing operations (at speeds up to 7200 bits per second).

The BSCA meets the requirements of high-speed communication systems by offering the following:

1. Transmission of the entire Extended Binary-Coded-Decimal Interchange Code.
2. Transmission of the entire American National Standard Code for Information Interchange.
3. Transmission of any binary data (entire programs in any language, packed decimal data, random data, etc).
4. Transmission speeds of 2000, 2400, 4800 or 7200 bits per second.
5. Block-dividing of messages to adapt to optimal conditions on telephone lines of a given quality.
6. Powerful error-checking methods.
7. Optional features that allow a variety of different operating modes.

The 2922-1 equipped with the BSCA can communicate with the following:

1. A System/360, all models, equipped with an IBM 2701 or 2703 Transmission Control and appropriate binary synchronous communication features, or

2. A System/370, all models, equipped with a 2701 or 2703 Transmission Control that has the appropriate binary synchronous communications features, or

3. Another 2922 Programmable Terminal.

The BSCA operates over two-wire or four-wire telephone lines that may be either privately-owned, leased, or switched (that is, part of a common carrier dial-up network). The transmission technique is half-duplex (that is, in one direction only at a time). The adapter can communicate in point-to-point fashion. In point-to-point operation, data exchange is between two stations.

The BSCA is fully program-controlled by 13 basic instructions (see Figure 3-2). To the 2922-1, the BSCA is another I/O device. However, considerable flexibility is introduced into the program by the binary synchronous philosophy. This philosophy features precisely-defined control character sequences that "frame" each message at its beginning and end. Such a message, with its control sequences at both ends, is moved byte-by-byte from main storage in the control unit to a buffer from where it is transmitted bit-by-bit (serially) to the remote CPU. Here, the stream of bits enters a similar buffer where the data is re-assembled into characters and moved eventually into main storage. The control characters trigger certain actions in both stations so that each message actually controls its own mode of transmission. Thus, the main controlling elements are the control character sequences which enable each message function as a self-controlling entity.

Transmission Codes

When ordering a 2922 terminal, the customer must specify either the ASCII or the EBCDIC as his line code. The adapter accommodates only one line code.

American National Standard Code for Information Interchange (ASCII): This code consists of seven data bits and one parity bit (Figure 3-1). Thus, an eight-bit character is formed which must have odd parity. The parity bit is supplied automatically by the BSCA as needed and, therefore, the parity bit position must always be zero in all ASCII

characters that are to be transmitted from main storage. The automatically-supplied parity bit is transmitted over the line; in the receiving BSCA, however, the bit is removed automatically and entered into main storage as a zero bit. The ASCII character is transmitted low-order bit (bit 1) first, bit 2 next and so on, ending with the parity bit. The ASCII table in Appendix L shows all bit configurations; the characters within thick lines are control characters, while those in the shaded areas cause line turnaround (change of transmission direction) whenever they are preceded by the control character DLE. This operation is explained later under "Control Sequences (Basic BSCA)".

Extended Binary Coded Decimal Interchange Code (EBCDIC): This code consists of eight data bits (see Figure 3-1). EBCDIC characters are transmitted low-order bit (bit 7) first, bit 6 next and so on, ending with bit 0. The EBCDIC table in Appendix L shows all bit configurations; the thick lines enclose the control characters, while the shaded areas contain characters that cause line turnaround (change of transmission direction) when they are preceded by the control character DLE. See "Control Sequences (Basic BSCA)".

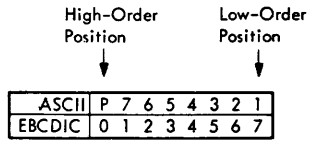


Figure 3-1. Sequential Numbering of Bit Positions in EBCDIC and ASCII

Interfaces

Interface Complying with EIA Requirement RS 232B: Two interfaces, the data set interface and the data station interface, are available for connecting the BSCA to U.S. modems.

The data set interface complies with the Electronic Industry of America (EIA) requirement RS232B; this interface allows appropriate modems to operate on voice-grade telephone lines at speeds ranging from 2000 to 7200 bits per second.

Modems

The modem converts dc signal output from the BSCA into modulated carrier signals for transmission over telephone lines.

The BSCA does not impose on the customer the use of any particular modem, but he must consider certain aspects when planning his BSCA installation. Modems that do not provide bit timing generally present few problems. Modems that are equipped with a clock and use relative phase angle modulation (4-phase modulation) offer the advantage of operating on narrow-band communication channels; in transparent mode, however, these modems are sensitive to certain steady bit patterns. In particular, modems that transmit the serial BSCA bits in pairs of bits (dibits) can lose synchronization when a sequence of more than 40 consecutive one-zero (or zero-one) patterns is transmitted. Likewise, high speed data stations cannot transmit a series of 500 (or more) zero bits unless the stations are equipped with a universal scrambler.

For guidance, refer to Systems Reference Library, General Information--Binary Synchronous Communications, Order No. CA27-3004.

DESCRIPTION (BSCA)

The information in this section pertains to the functional characteristics of a BSCA.

INSTRUCTION SET AND CONTROL CHARACTERS (SUMMARY)

The instruction set for the BSCA is shown in Figure 3-2, while EBCDIC and ASCII control character codes are given in Figures 3-3 and 3-4 respectively.

Type	Format	Op Code	DA	FS	Function
XIO	SS	D0	5	0	Transmit and Receive
XIO	SS	D0	5	1	Receive Initial
XIO	SS	D0	5	4	Receive Only
XIO	SS	D0	5	8	Transmit Only
CIO	SI	9B	5	0	Disable ITB
CIO	SI	9B	5	1	Enable ITB
CIO	SI	9B	5	2	Enable BSCA
CIO	SI	9B	5	3	Disable BSCA
CIO	SI	9B	5	6	Store Current Address
CIO	SI	9B	5	7	Store Sense Information
CIO	SI	9B	5	8	Store ITB Address
TIOB	SI	9A	5	0	Test and Branch on Any Indicator Set
TIOB	SI	9A	5	8	Test and Branch on Busy

Figure 3-2. BSCA Instruction Set

Bit Position 0 1 2 3 4 5 6 7	Mnemonic	Function
0 0 0 0 0 0 0 1	SOH	Start of Heading
0 0 0 0 0 0 1 0	STX	Start of Text
0 0 0 0 0 0 1 1	ETX	End of Text
0 0 1 0 0 1 1 0	ETB	End of Transmittal Block
0 0 1 1 0 1 1 1	EOT	End of Transmission
0 1 1 *	Column 3	Change of Direction Modifier
0 0 1 1 1 1 0 1	NAK	Negative Acknowledgement
0 0 1 0 1 1 0 1	ENQ	Enquiry
0 0 0 1 0 0 0 0	DLE	Data Link Escape
0 0 1 1 0 0 1 0	SYN	Synchronization Character
0 0 0 1 1 1 1 1	ITB	End of Intermediate Transmission Block

*'Do not care' positions
These bits are used at the discretion of the programmer

Figure 3-3. EBCDIC Control Character Codes

Bit Position P 7 6 5 4 3 2 1	Mnemonic	Function
P 0 0 0 0 0 0 1	SOH	Start of Heading
P 0 0 0 0 0 1 0	STX	Start of Text
P 0 0 0 0 1 0 0	EOT	End of Transmission
P 0 0 0 0 0 1 1	ETX	End of Text
P 0 0 1 0 1 1 1	ETB	End of Transmittal Block
P 0 1 1 [*]	Column 3	Change of Direction Modifier
P 0 0 1 0 1 0 1	NAK	Negative Acknowledgement
P 0 0 0 0 1 0 1	ENQ	Enquiry
P 0 0 1 0 1 1 0	SYN	Synchronization Character
P 0 0 1 0 0 0 0	DLE	Data Link Escape
P 0 0 1 1 1 1 1	ITB	End of Intermediate Transmission Block

*'Do not care' positions
These bits are used at the discretion of the programmer

Figure 3-4. ASCII Control Character Codes

OPERATING PRINCIPLE

Operations in the BSCA are started by initialization, that is, by bidding for the line. The initialization assigns the right to begin transmission to one or the other station. If the BSCA gets the line, it must first receive the bid from the other station before it can start its transmission. The actual data exchange is accomplished with Transmit and Receive instructions that are issued in both stations. Before each XIO instruction (transmit/Receive) is executed, the condition code is set to indicate to the program whether the instruction can be executed or not. After each XIO instruction is executed, an interrupt is requested to provide a means for checking the results. If an error occurs, details of the cause for rejection (in the form of sense information) are available at interrupt time.

Initialization At Transmitting Station

The actual wire connection between two stations is either permanent (on private and leased lines) or it must be established by a dialing procedure (in switched networks).

After connection is established, an initialization procedure must be used. For reasons that are explained later in "XIO Instructions"

under "Transmit and Receive", the initialization is accomplished by two different methods that ensure the proper rhythm of data exchange between the two stations. The method used depends on the intentions of the user of a particular station. If he intends first to send data to a remote station, he must use one method; if he intends first to receive data from the remote station, he must use the other method. Thus, initialization can be likened to starting on 'one foot or the other', one station using one method and the other station using the opposite method; neither method represents an advantage or a disadvantage because, after initialization, each station transmits in turn.

Before any other action, the BSCA is enabled by issuing an Enable BSCA instruction. This instruction produces a positive level on the signal line 'data terminal ready' that runs from the BSCA to the modem or data set; this positive level is needed for the modem to become operational. No other preparation is needed if the telephone line is a private or leased line not using a telephone set; for preparation on switched lines, however, the operator must dial the other station, listen for the answer tone, and manually switch the modem to data mode immediately after the answer tone cuts off.

The modem then switches itself to the telephone line and returns the confirmation signal 'data set ready' to the BSCA. Until this signal is received, the BSCA program loops on condition code 11 (binary).

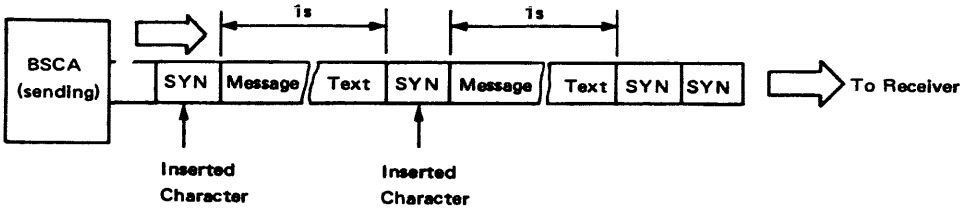
The transmitting BSCA bids for the telephone line by issuing a Transmit and Receive instruction. The instruction reads out the actual bid from the terminal main storage and transmits it to the remote station. In the simplest form, the bid consists of an Enquiry (ENQ) character that is interpreted by the recipient as "Who are you?" A more sophisticated bid consists of some identification followed by the ENQ character which is interpreted as "I am xxx, who are you?"

The BSCA reacts to the Transmit and Receive instruction that is issued to send the bid by setting a condition code to inform the program how it may proceed:

Condition Code	Function
0 (binary 00)	BSCA is 'available' and has accepted the instruction.
1 (binary 01)	BSCA is 'working' (it is still processing a previously-issued XIO instruction).
3 (binary 11)	Manual intervention is required because the modem or data set is not operational.

If the BSCA was not enabled when the Transmit and Receive instruction was issued, condition code 3 (binary 11) is set to indicate

that the BSCA is not operational. In this example, it is assumed that the BSCA has accepted the instruction; therefore, the BSCA activates the modem interface line 'request to send'. This causes the modem to set its carrier on the line and subsequently to respond with 'clear to send'. At this point, transmission over the telephone line begins with two automatically-supplied synchronization (SYN) characters that provide synchronization with the other station, the actual bid follows next, and an automatically-supplied pad character (sent behind the ENQ character in the bid) ends the transmission.



At this point, however, the instruction has not ended because the ENQ character causes the BSCA to turnaround, and to 'listen' for a response from the remote station. Normally, the Transmit and Receive instruction ends only when a reply is received.

However, the BSCA spends only a limited amount of time on awaiting a reply. As soon as turnaround occurs, a receive timeout delay of 3-seconds duration starts running. This timeout delay ensures that the BSCA cannot be hung-up indefinitely and unnoticed. The timeout delay is reset by every none-synchronizing character that is received immediately after a SYN character. Since the BSCA automatically inserts a SYN character after a certain time interval during transmit operations, The 3-second timeout delay (started in the receiving BSCA) normally cannot elapse when SYN characters are detected in the data stream. It is only if either all SYN characters, or all data, or nothing is received that a timeout occurs to end the instruction. The SYN characters are not entered into main storage.

The end of a Transmit and Receive instruction is indicated by a request for an interrupt. As explained later, tests performed in the interrupt routine reveal the ending status of the instruction. With the acceptance of the Transmit and Receive instruction, the BSCA becomes busy; the busy condition ends when interrupt is requested.

Initialization At Receiver (Private or Leased Lines)

The receiving station prepares its BSCA to receive a bid from the transmitting station by issuing a Receive Initial instruction. Prior to this instruction, the BSCA must have been enabled, otherwise the Receive Initial instruction is rejected, whereby condition code 3 (binary 11) is set to indicate that the BSCA is not operational. The BSCA reacts to the Receive Initial instruction by setting a condition code to inform the program how it may proceed.

Note: If the modem has power off, the condition code for a Receive Initial instruction is set to 00 (available) and, therefore, modem power off is not noticed immediately.

In this description, it is assumed that the Enable BSCA instruction was given and that the Receive Initial instruction was accepted. The BSCA does not immediately become busy; the busy condition begins only with the arrival of data from the other station. The 3-second timeout delay is started when synchronization (character phase) is established between two stations. To avoid a permanent hang-up of the BSCA, a timeout occurs under any of the following conditions:

1. If no data is received.
2. If a constant stream of SYN characters arrives.
3. If data is received that does not contain a character (such as ENQ) to cause turnaround.

The end of a Receive Initial instruction is indicated by a request for an interrupt. This interrupt request is made under any of the following conditions:

1. When a bid has been received and stored in main storage.
2. When a timeout occurs.
3. When an interfering burst operation ends.

Initialization At Receiver (Switched Lines)

On switched telephone lines a manual answering procedure.

Manual Answering Procedure

An incoming call is denoted when the bell in the telephone set rings. The operator picks up the handset to connect with the other operator, then starts the BSCA program. This program must issue first an Enable BSCA instruction, then a Receive Initial instruction. Once the BSCA program has started, the operator manually switches the modem to answer status and, if so desired, places the handset on the cradle.

With the Receive Initial instruction issued, the BSCA is ready to receive a bid as soon as the modem has sent out its 3-second answer tone. Some modems generate this answer tone and subsequently indicate 'data set ready' to the BSCA; other modems do not generate an answer tone. However, either type of modem can be used, provided the BSCA is equipped with the correct modem interface feature.

For modems not capable of sending the answer tone, the BSCA generates the 3-second tone (due to the interface feature) and, subsequently, is ready to receive. On switched telephone lines, the 3-second timeout is started as soon as the answer tone has been sent, and the BSCA now begins to search for synchronization. An interrupt will then indicate that either a timeout occurred or a bid was received.

Interrupt

The interrupt indicates that an XIO instruction has ended. The BSCA interrupt is identified by the device address 5 and the function specification 6 that are stored in the old program status word. Since it is not known how a particular instruction has ended, the program must test certain indicators at interrupt time to find out whether the instruction has ended in normal or in abnormal fashion. The testing is simplified by use of the TIOB On Any Indicator Set instruction. If no indicator is set, the previous instruction has ended without errors, which means that either a bid, or a positive or negative acknowledgement, or a regular message, has been received. If one (or more) indicators are on, it is necessary to retrieve them from the unit in which they were generated and to store them into a main storage location where they can be inspected by the program. The Store Sense Information instruction is used for this task.

If an instruction has ended with timeout, the timeout flag in the sense information would be on. If a Receive Initial instruction has been interfered with by a burst operation that coincided with an incoming bid, the overrun flag in the sense information would be on. For details of all error situations, see "Sense Information".

If the error testing at interrupt time reveals that the initialization was successful, the data exchange may begin with the issue of a Transmit and Receive instruction.

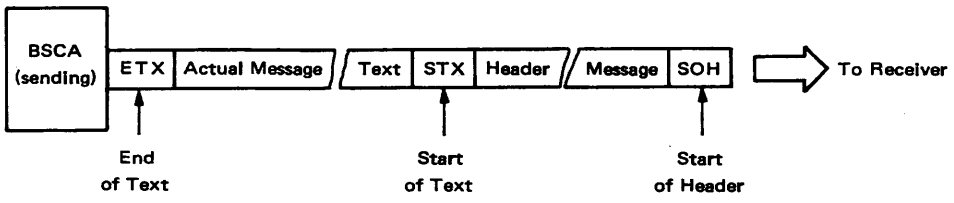
The individual transmit and receive portions in corresponding instructions of both stations mutually overlap. This overlapping creates the proper rhythm of data exchange which is shown later in Figure 3-10

Contention

On privately-owned or leased lines, it is possible for two stations to vie for the line at the same time. Therefore, priority is assigned to one station. Where contention exists, the assigned station (defined as the primary station) gains control of the line. The other station (defined as the secondary station) relinquished its bid for the line and becomes a slave whenever contention situations arise.

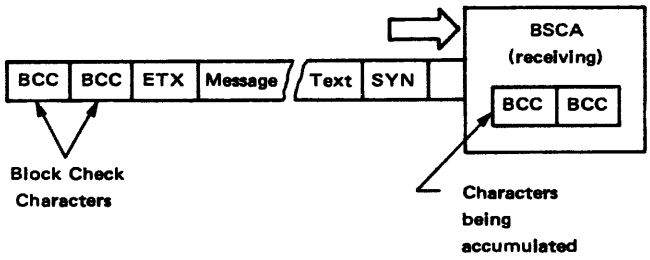
Priority can be allotted by different methods; for example, the program of the primary station may ignore inquiries from the secondary station and react only to positive acknowledgements. Thus, the secondary station is eventually forced to 'listen' to the primary station.

In switched networks, the distinction between primary and secondary stations is not made; normally, contention is not possible because the two stations are only connected when transmission is desired, and the calling stations has priority. However, if one station replies to a receive bid with an acknowledgement which indicates that reception cannot continue, contention is possible on switched lines too, because both, stations remain connected after such a reply. For details, see "Control Sequences".



Basic Message Structure

INTERMEDIATE BLOCK CHECKING MODE



Basic Checking Scheme

Operation in Intermediate Block Checking mode allows the programmer to divide a message into small blocks, so that each block may be checked separately. Turnaround (with acknowledgement of each intermediate block) is not needed and, therefore, overall throughput is increased.

The length of the intermediate blocks can be freely chosen. A block-divided message can be processed by the receiver while transmission is still in progress because each block (that is received free of errors) is made available by means of an intermediate interrupt. This interrupt is requested only in the receiving BSCA and it is identified by the funciton specification 2; it is not effective, however, during high-speed operation. The final interrupt (function specification 6) occurs only at the end of the entire message, that is, when turnaround occurs; this final interrupt resets any pending intermediate interrupt.

Note: If the channel mask bit in the PSW is temporarily disabled (zero) while Intermediate Block (ITB) messages are being received, intermediate interrupts are lost. However, the final interrupt remains pending and will occur when the channel mask is set to one.

The ITB mode is automatically disabled after power on or system reset. If this mode is to be used, the Enable ITB instruction must be issued. The mode is operational until either Disable ITB or system reset is performed. The Enable/Disable instructions are executed immediately, provided that the BSCA is not busy. In the busy condition, the instructions are held in the busy loop and are executed as soon as the BSCA becomes available.

When the BSCA is programmed by means of the Input/Output Control System (BSCA IOCS), Enable ITB and Disable ITB instructions need not be specially issued, as these instructions are already included in the IOCS routines. Conversely, where BSCA IOCS routines are not used, ITB mode and non-ITB mode can only be set by incorporating Enable/Disable ITB instructions in the user program.

The message is divided into blocks by the ITB character as shown in Figure 3-5. The ITB characters are recognized as normal data with no other function when the ITB mode is not enabled.

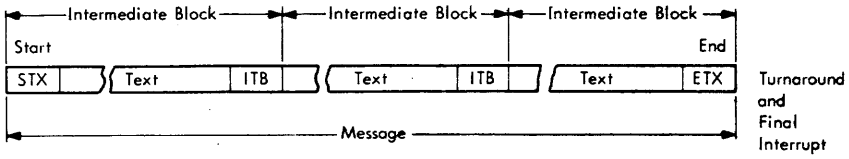


Figure 3-5. ITB Message

When the intermediate interrupt occurs in the receiving BSCA, the last-received block can be used directly without testing for any errors. The error testing is superfluous because the intermediate interrupt is requested only when The received block is error-free; in addition, the TIOB on Any Indicator Set instruction cannot be used because the BSCA remains busy for the duration of the entire message and, therefore, the test instruction would be locked in the I/O-busy loop. If a block is faulty, no intermediate interrupt occurs for this and all following blocks. At final interrupt time (end of message), error testing can be performed.

During ITB operations, the receiving BSCA stores automatically the address of each ITB character. This address is updated by every block that is received free of errors, and thus, it shows where the ITB character of the last-received error-free block is located in the main storage. If an error occurs, the ITB address is not updated but remains unaltered so that the address of the last 'good' block can be made available at final interrupt time. The address is made availalbe by issuing the Store ITB Address instruction. The instruction is executed immediately (even though the BSCA is busy) and moves the ITB address to the main-storage position addressed by B1-D1.

The block in which an error is detected is stored into main storage as it it were a good block, but any further blocks that belong to the message are not stored, even if they are received free of errors. However, the last character of the message is stored into the byte next to the last byte of the faulty block.

All blocks that belong to a message must be transmitted with one Transmit and Receive instruction. Transmitting a single block defeats the purpose of block checking and, if a single block does not end with the ETX character, a timeout occurs because there is no turnaround after ITB.

Intermediate block checking is inhibited by the Disable ITB instruction.

Intermediate block checking can be used irrespective of the operating speed of the BSCA.

INSTRUCTIONS (BSCA)

The information in this section pertains to the functional characteristics of a BSCA.

XIO INSTRUCTIONS

Transfer I/O instructions occupy six bytes of main storage and consist of the operation code, the device address, the function specification, the address of the first (leftmost) byte of the main-storage input or output field, and the field length. The B1-D1 field can be used as the direct address of the input/output field, or an effective address may be generated by adding the contents of a general register (defined by B1) to the displacement (D1). The address thus derived must not exceed the storage capacity or violate the protected area, and must not be equal to zero as this causes a specification error. The field length is similarly derived from the B2-D2 field of the instruction and represents the true field length. The field length is variable from 1 to 4,095 bytes.

All XIO instructions set the condition code prior to execution and request an interrupt after execution.

Receive Initial

The Receive Initial instruction prepares the BSCA to receive an initial sequence (a bid) from the other station. When this instruction is issued, the BSCA remains in a non-busy state until synchronization is achieved (on leased lines) or until a bid arrives (on switched lines). However, the BSCA cannot be considered as fully available at this time, becuase the issue of a further XIO instruction would cause a condition code of 01 (binary) to be set.

When synchronization is established, or when the bid arrives, the BSCA becomes busy and the bid is stored into the main-storage field defined by the B1-D1 address. An interrupt is requested and the busy condition ends when the bid is stored or when a timeout occurs due to a faulty message (all SYN characters, or all data, or nothing) being received.

In leased- or private-line operations, a timeout during Receive Initial can occur only because the 3-second timeout delay is started when synchronization is achieved, as opposed to switched-line operations where the timeout delay starts when the answer tone is transmitted.

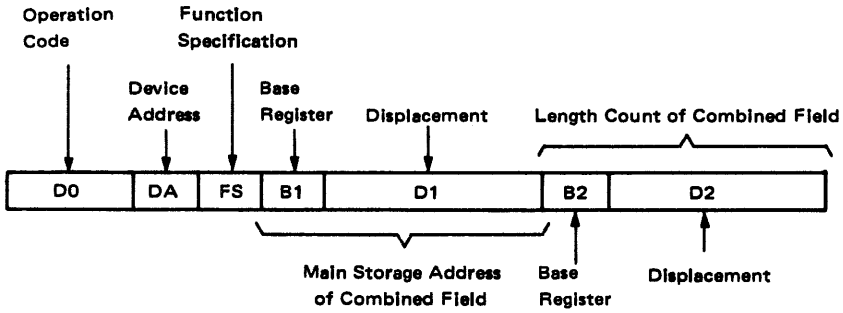
A receive initial operation can be cancelled by the Disable BSCA instruction, provided that this instruction is issued before an initial sequence is received.

Wait State

The Receive Initial instruction can be used to place the terminal in a wait state, during which the control unit is kept in a loop but is prepared to receive and, subsequently, to process an incoming call. As a prerequisite, the toggle switch on the BSCA console (see Figure 82) must be set to WAIT STATE; this can be done any time before or during processing operations of any kind, because the control unit enters the wait state only when a Receive Initial instruction is issued. The REC INITIAL OR ADPREP light on the BSCA console comes on when the wait state begins. When a call comes in (on switched lines), or when synchronization is achieved (on leased lines), the 3-second timeout is started to prevent hangup. The wait state allows data exchange with an unattended station.

Transmit And Receive

In the half-duplex mode of data exchange that is used by the BSCA, transmission occurs in one direction only at a time. Therefore, at any given moment, one station transmits and the other station receives. The station which has the right to transmit a message at a given instant is defined as the master station. The station which is receiving a message from a master, and which is obliged to transmit an appropriate reply, is defined as the slave station. The expressions master and slave define only temporal states of the interacting stations, the roles of which can change many times during the course of a transmission.

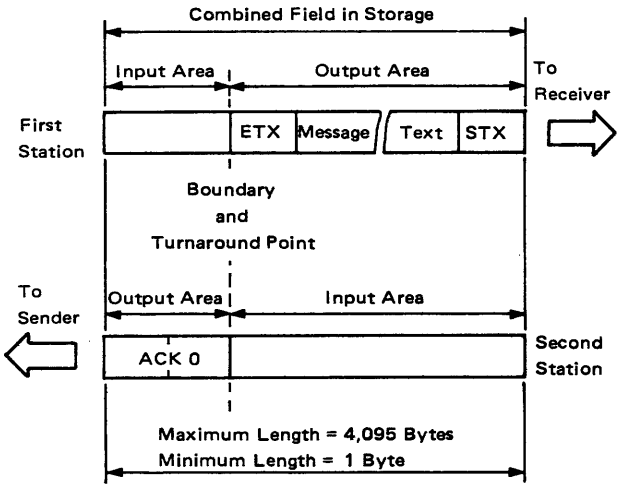


Structure of a Data Transfer Instruction

The Transmit and Receive instruction serves a dual purpose as the name indicates, that is, it causes the BSCA to transmit and subsequently to turn around to receive. This dual function is needed because of the extremely fast response coming from the other station. The B1-D1 field of the Transmit and Receive instruction defines the leftmost byte of a main-storage field that represents an output area as well as an input area, while the field length (defined by B2-D2) pertains to the total length of this combined output/input field. The length is variable from 1 to 4,095 bytes, any portion of which may be chosen as the output field, with the remaining portion representing the input field.

The master station assigns to the output portion of the combined field the message to be transmitted and reserves the input portion for the reply from the slave. The reply is usually a standard acknowledgement. See "Control Sequences (Basic BSCA)".

The slave station uses the Transmit and Receive instruction in a similar way, but for a different purpose; it assigns its reply to the output area of the combined field and reserves the input area for the next message it expects to receive.



Mutual Message Arrangement

Any message that is to be transmitted or received must be framed by control characters on both ends. These control characters activate certain functions in the BSCA when the message is read out from the main storage of the master and, subsequently, they activate corresponding functions in the slave upon arrival. See "Control Characters (Basic BSCA)". Figure 3-6 shows a combined field as it appears in the main storage of the master and Figure 3-7 shows the corresponding field as it is used by the slave.

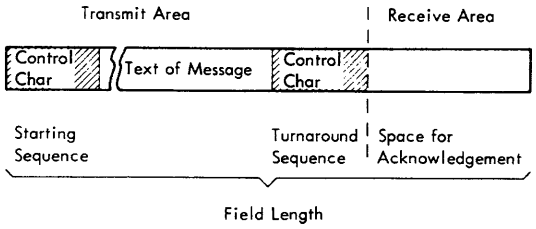


Figure 3-6. Transmit and Receive Area Used by Master Station

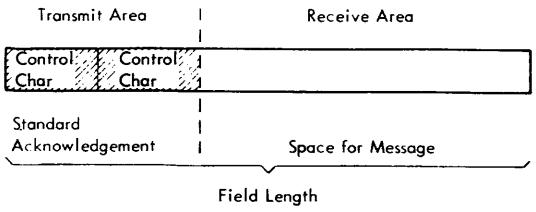


Figure 3-7. Transmit and Receive Area Used by Slave Station

When a Transmit and Receive instruction is issued, the output portion (transmit area) of the combined field is always dealt with first. For the master, this means that the message is sent out bit-by-bit until the control character at the end of the message is detected. This control character triggers the turnaround function in the master. The turnaround character thus marks the boundary between output and input areas. With the detection of the turnaround character, the 3-second timeout is started to prevent a hangup situation in the event that nothing is received. The BSCA becomes busy when the Transmit and Receive instruction is accepted and this condition lasts until interrupt occurs. Interrupt is normally requested only when the reply has been received and stored into the input portion (receive area) of the combined field. As explained later, interrupt is also requested when certain abnormal conditions are detected.

A Transmit and Receive instruction is processed in exactly the same way in the slave station as in the master station. The only difference is in the logical purpose for which the instruction is used; the slave station sends an acknowledgement first after which it turns around to receive the next message. Again, an interrupt is requested when this new message has been stored or when the timeout delay has elapsed if nothing was received.

Limited Conversational Mode

From the foregoing explanation, it may seem that the original transmitter has an advantage over the receiver, in that the transmitter may send messages of any length while the receiver is limited to sending standard acknowledgements. However, this is not necessarily so; the receiver is allowed to send a message of any length in reply, provided it has received the previous message from the transmitter without errors. This type of operation is termed limited conversational or 'tete-a-tete' mode to distinguish it from the method of replying with standard acknowledgements.

Operation in limited conversational mode permits the slave (that has received from the master an error-free message ending in End of Text (ETX) to reply with a message instead of with a standard acknowledgement. However, the original master must then reply with an acknowledgement, and not with a message, before the transmission can continue. Thus, in limited conversational mode, a conversational reply cannot be made to a conversational reply. The status of the two stations involved in the transmission therefore alternates between master and slave as shown in Figure 3-8.

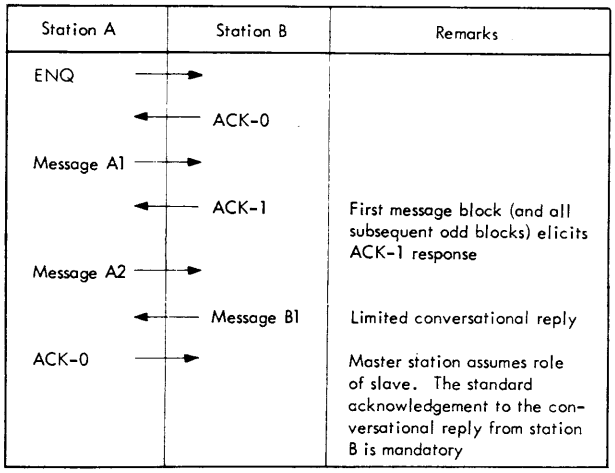


Figure 3-8. Operation in Limited Conversational Mode

Limited conversational mode speeds up overall throughput because a conversational reply, which replaces the actual acknowledgement, implies that the previous reception was error-free. When a previous reception was faulty, however, the slave must send a negative acknowledgement to the master. Figure 3-9 shows the main-storage fields for a Transmit and Receive instruction used in limited conversational mode.

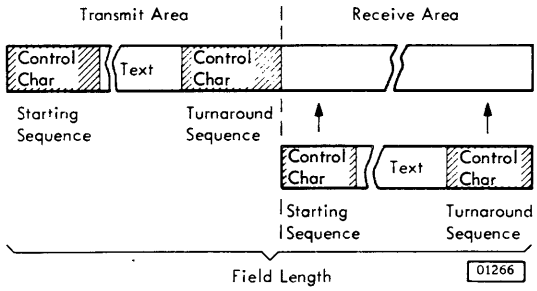


Figure 3-9. Transmit and Receive Area Used in Limited Conversational Mode

Since Transmit and Receive instructions are used by both stations, a conflict must be avoided; this is achieved by using two different initialization procedures. The master starts with a Transmit and Receive instruction while the slave begins with a Receive Initial instruction. Thus, an offset in the rhythm of data exchange is produced to ensure that the transmit portion of a given instruction is always faced with a receive portion at the other station (Figure 3-10). The Transmit and Receive instructions used in both stations are not

necessarily mirror images of each other. However, two types of errors (short record, and storage wraparound) must be avoided.

The short record error arises when the message that is to be transmitted, or the message that is being received, is longer that the field length specified in the Transmit and Receive instruction. When a short record is detected, an interrupt request is generated immediately. The storage wraparound error arises when the storage capacity is exceeded during a transmit or receive operation. For this error, the interrupt request is also generated immediately. Both error cases are registered in the sense information, which is available at interrupt time.

Receive Only

The Receive Only instruction sets the BSCA to receive mode and the 3-second timeout is started. The instruction is used when a previously-issued Transmit and Receive instruction has ended wth timeout and the receiving BSCA assumes that the transmitter is only delayed temporarily, but will start sending again eventually. Thus, the Receive Only instruction serves as a timeout extender. When the instruction ends with timeout again, nothing is received. The Receive Only instruction can be issued as often as desired. The BSCA becomes busy when it has accepted this instruction, and the busy condition lasts until an interrupt occurs.

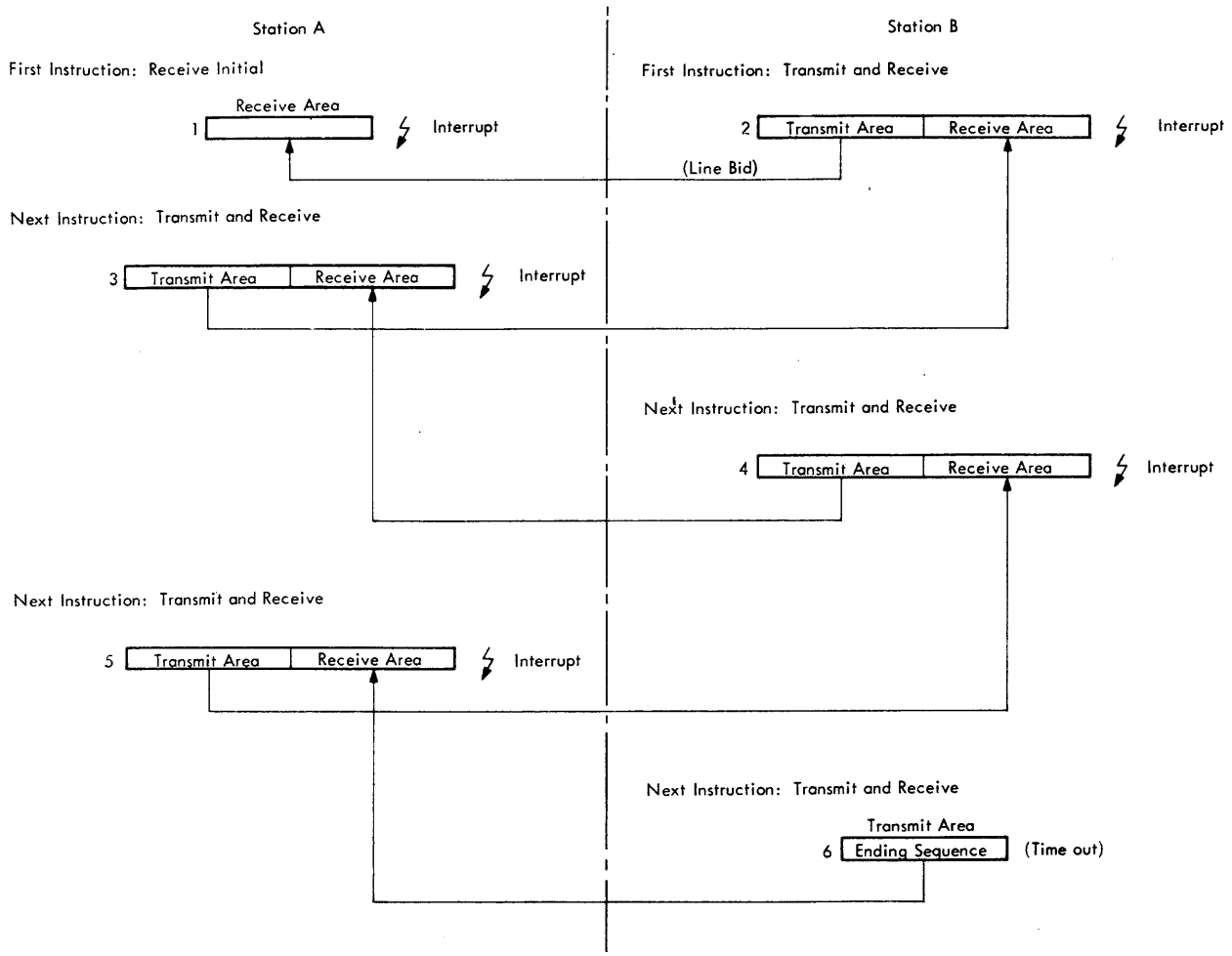


Figure 3-10. Data Exchange via Transmit and Receive

Transmit Only

The Transmit Only instruction is used for sending a message to which a reply is either not expected or not desirable. For example, when it is necessary to send a negative acknowledgement because operations are held up by certain circumstances, it may be advantageous to use a Transmit Only instruction to avoid receiving any further data from the other station. The BSCA becomes busy when it has accepted this instruction, and the busy condition ends when interrupt is requested.

CIO INSTRUCTIONS

Control I/O instructions occupy four bytes of main storage and consist of the operation code, the device address, the function specification, and the B1-D1 field; this field further specifies the control function that is to be performed in the BSCA according to the rules for direct or effective addressing. These rules are given previously in this manual in "Control Unit" under "Addressing". When a Control I/O instruction is issued, it is executed immediately and no condition code is set. However, if the BSCA is working (that is, processing a previously-issued XIO instruction), the Control I/O instruction enters the I/O-busy loop and is executed when the BSCA becomes available.

Enable BSCA

The Enable BSCA control instruction has a general "turn on" function, and is a prerequisite for all BSCA operations. When the instruction is issued, the control signal line 'data terminal ready' that runs from the BSCA to the modem or data set becomes positive; this positive level makes the modem operational. The Enable BSCA instruction must be issued before the first Transmit and Receive, Receive Only, or Receive Initial instruction, or otherwise a program error stop occurs when any of these instructions are issued.

Enable ITB

The Enable ITB instruction sets the BSCA to the intermediate block check mode. An Enable instruction is executed immediately provided that the BSCA is not busy; otherwise, the instruction is held in the busy loop and is executed as soon as the BSCA becomes available.

In ITB Mode, the receiving station recognizes and decodes the ITB characters as they occur in the incoming data stream. The ITB character allows the message to be divided into blocks, each block being checked separately. If found correct, the block is made available immediately for further processing at the receiver by means of an intermediate interrupt. Thus, throughput is increased because a block-divided message can be processed by the receiver while transmission is still is progress.

ITB mode remains operational until either Disable ITB or system reset is performed.

Disable BSCA

The Disable BSCA control instruction, when issued, makes the 'data terminal ready' signal line negative. The instruction can be used to withdraw from an intended data exchange that was started, provided it is issued prior to the reception of the bid or acknowledgement from the other station. If Disable BSCA is issued while a transmission is in progress, the current operation continues to completion and all operations then cease because the BSCA enters a reset state.

Disable ITB

The Disable ITB instruction is used to disable the intermediate block checking function of the BSCA. A Disable instruction is executed immediately provided that the BSCA is not busy; otherwise, the instruction is held in the busy loop and is executed as soon as the BSCA becomes available.

After the Disable ITB instruction has been issued, the receiving station does not recognize the control function of the ITB character, which, if received, is treated as normal data. In this case, no intermediate block checking is carried out and no intermediate interrupts are requested. Received data is checked and interrupt is requested only when the complete message has been received.

Disable ITB remains operational until an Enable ITB instruction is executed.

Store Current Address

The current address is the address of the last-received byte, updated by plus one. Thus, it defines the beginning of the next combined input/output field in main storage or the end of the previous message. The Store Current Address instruction moves the current address to the storage location addressed by the B1-D1 field of the instruction. By means of this instruction, search operations can be avoided and, therefore, inspection of the last-received message is simplified.

Store ITB Address

In the intermediate block checking mode, the receiving station automatically stores the address of the last ITB character and updates it for each error-free block received. At final interrupt time, this address, therefore, shows where the last-received correct block is positioned in main storage. The purpose of the Store ITB Address instruction is to make the address of this block available for inspection by the program.

The instruction, which is executed immediately (even if the BSCA is busy), moves the ITB address to the mainstorage location addressed by the B1-D1 fields.

Store Sense Information

At interrupt time, information on the ending status of the previously-executed XIO instruction is available. When a TIOB on Any Indicator Set instruction shows that the previous XIO insttuction was ended by an abnormal condition, the Store Sense Information instruction can be used to retrieve the error indications from the inaccessible auxiliary storage and to store these indications into that byte of main storage that is addressed by the B1-D1 field of the Store Sense Information instruction. The following conditions can thus be stored:

1. EOT received but no XIO issued.
2. Transmit/receive mode error.
3. Short record.
4. Storage wraparound.
5. Overrun.
6. Timeout.
7. Cyclic, longitudinal, or vertical redundancy check (CRC/LRC/VRC).
8. Parity check.

For explanations of these checks, see "Sense Information".

TIOB INSTRUCTIONS

Test I/O Branch instructions occupy four bytes of main storage and consist of the operation code, the device address, the function specification, and a B1-D1 field from which the branch address can be derived according to the rules for direct or effective addressing. The TIOB instructions test direct or effective addressing. The TIOB instructions test for a particular condition in the BSCA and if the condition is present, the program branches to the address defined by B1-D1. If the particular condition is not present, the program continues with the next sequential instruction address located in the program status word. When TIOB instructions are issued, no condition code is set.

TIOB On Any Indicator Set

The TIOB on Any Indicator Set instruction can be used to find out whether an instruction was ended by an abnormal condition. If any of the check conditions that are contained in the sense information (time-out, overrun, etc.) are present, the program branches. In this manner, the program can determine whether it is necessary to store the sense information. The instruction is executed immediately, except when the BSCA is processing a previously-issued XIO instruction. In that case, the TIOB on Any Indicator Set instruction enters the I/O-busy loop and is executed as soon as the BSCA becomes available.

TIOB On Busy

The TIOB on Busy instruction is used to find out whether the BSCA is busy (that is, processing a previously-issued XIO instruction); if so, the program branches. The BSCA becomes busy when it has ccepted an XIO instruction. The busy condition ends when an interrupt occurs. Thus, the BSCA-busy condition is identical with the BSCA-working condition, except for the following difference in the use that is made of both indications:

1. The condition code for 'working' is set, after the XIO instruction has been attempted, to inform the program why the attempt was unsuccessful.

2. The TIOB on Busy instruction can be performed to find out whether an XIO instruction would be successful if it were issued.

The TIOB on Busy instruction is always executed immediately, regardless of the BSCA condition.

CONTROL CHARACTERS (BSCA)

The control characters are used either singly or in groups of two consecutive characters (thus forming a control sequence) to cause two actions in the BSCA; setting the BSCA to text mode, and setting the BSCA to control mode. Data can be exchanged in either mode; however, text mode is reserved strictly for text transmission, while control mode is used for preparatory and administrative purposes. For example, when the BSCA has called another station or when it has been addressed by another station, it is in control mode.

In control mode, all control characters are effective so that, if used, they activate certain functions (such as the setting of text mode) in both stations.

In text mode, control characters must not be used because they make the message invalid. Only specific control characters (and/or sequences) cause the BSCA to return to control mode when they are used at the end of the text. The return to control mode is always accompanied by turnaround so that the other station has an opportunity to reply.

The actions of the individual control characters are described in the following paragraphs, and the bit configurations of the control characters are listed in Figures 3-3 and 3-4.

Start of Heading (SOH)

The SOH character sets the BSCA to text mode and indicates that the data that follows is an identification of the message that will follow eventually. The heading can be a name, a text, or a number. The SOH character cannot be used in the message.

Start of Text (STX)

The STX character sets the BSCA to text mode and indicates that the data that follows is the actual text of the message. STX cannot be used in the text.

End of Text (ETX)

The ETX character causes the BSCA to leave text mode and to enter control mode. This mode change causes turnaround and both stations remain connected. ETX is commonly used to mark the boundary between the core storage transmit and receive areas.

End of Transmittal Block (ETB)

The ETB character is an alternate for ETX, ETB can be used to divide a message into blocks of optimal length for a telephone line of a given quality. The function of ETB is identical to that of ETX; ETB causes the BSCA to leave text mode but it indicates to the program tht the next transmission will be the logical continuation of the one that has just ended.

End of Transmission (EOT)

The EOT character is used by the transmitting station to tell the recipient that the current transmission will not be continued. EOT can only be used in control mode; the character causes turnaround so that the receiver has an opportunity to start its own transmission. Both stations remain connected after EOT is sent. EOT must not be used in text mode.

Negative Acknowledgement (NAK)

The NAK character is used as a reply to tell the transmitter that the received message cannot be accepted because it was faulty. NAK can only be used in control mode; the character causes turnaround and both stations remain connected. The NAK reply is likely to be received from a station to which text containing control characters was sent in text mode.

Enquiry (ENQ)

The ENQ character can be used in text mode as well as in control mode. ENQ causes turnaround and both stations remain connected. In control mode, ENQ is used as the initial sequence or as request for information on the status of the other station. The program should, therefore, react to a received ENQ by sending information that indicates whether transmission can continue or not. When ENQ appears in the data stream during text mode, both stations return to control mode and turnaround occurs. ENQ can be used by the transmitter to abort the current transmission.

Data Link Escape (DLE)

The DLE character is used in control sequences only; it cannot stand alone and must not be used in text mode. For details, see "Control Sequences (Basic BSCA)".

Synchronization Character (SYN)

The SYN character is automatically inserted into the data stream at the proper time to establish and maintain synchronization. The program has no influence on this SYN insertion but can use SYN in text mode or in control mode as a filler character, provided that such fillers are sent for less than 3 seconds. If SYN is sent for more than 3 seconds, a timeout occurs. SYN characters do not enter main storage.

Control Sequences (BSCA)

Control sequences are formed by two consecutive control characters with no data between them. The sequences are: standard affirmative replies, turnarounds, and ending sequences. Control sequences can be used in control mode only, they must not be used in text mode.

Standard Affirmative Replies

In normal data exchange, it is customary to acknowledge each received message with an affirmative reply. If the reply cannot be positive because the received message was faulty, a NAK character must be used as reply.

Odd-numbered messages (first, third, fifth, and so on) are acknowledged by ACK-1 and even-numbered messages are acknowledged by ACK-0. The abbreviation ACK stands for Acknowledgement, but the actual acknowledgement is realized by the DLE character as the leader, followed by specific bit combinations as the trailer. A third positive acknowledgement, which has a special meaning is Wait Before Transmit-Positive Acknowledgement (WACK), which indicates to the transmitting station that the receiver has a temporary not-ready condition.

Figure 3-11 shows the three standard affirmative replies in ASCII and EBCDIC. All standard replies cause turnaround. Standard affirmative replies, however, need not always be used particularly when the BSCA is operating in limited conversational mode. In this case, another message can be used as the positive reply, provided the previous transmission was received correctly and was not itself a conversational reply. However, when the previous transmission received was in error, the NAK character must be used as a negative reply. Standard affirmative replies and NAK can only be used outside of text mode.

Function	Mnemonic	Code Representation			
		ASCII		EBCDIC	
		Sequence	Hex Notation	Sequence	Hex Notation
Even Acknowledge	ACK-0	DLE 0	10 30	DLE 70	10 70
Odd Acknowledge	ACK-1	DLE 1	10 31	DLE /	10 61
Wait Before Transmit -Positive Acknowledge	WACK	DLE ;	10 3B	DLE ,	10 6B

*Hexadecimal notation (no graphic assignment)

Figure 3-11. Standard Affirmative Reply Sequences

Start of Block Sequences

To protect against the possible loss or duplication of data blocks when operating in limited conversational mode, two sequences can be used as a programming convention to identify each message. These sequences can either precede the heading or they can be used in the

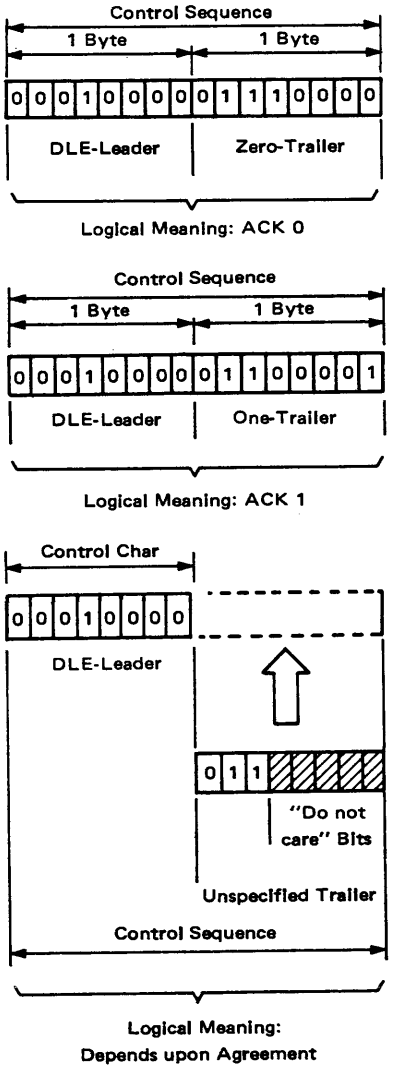
heading. They represent the functions Start of Block (SOB)-0 and SOB-1 (Figure 3-12). Both sequences activate no function in the BSCA, but concern only the program.

Function	Mnemonic	Code Representation	
		ASCII	EBCDIC
Start of Block (Even)	SOB-0	DLE /20/	DLE /40/
Start of Block (Odd)	SOB-1	DLE /21/	DLE /51/

Note: The trailers are shown in hexadecimal notation

Figure 3-12. Start of Block Sequences

Turnaround Sequences



Basic Types of Turnaround Sequence

Turnaround sequences are used to effect turnaround when the BSCA is in control mode. These sequences must not be used in text mode. The sequences consist of the DLE character followed by any of the 'Column 3' characters. Column-3 characters are located in the third column of the ASCII table in Appendix L, and are shown also in the shaded areas of the EBCDIC table in the same appendix. The Column-3 characters alone have no effect.

The character sequence DLE/Column 3 can be used as an acknowledgement to which a special meaning may be assigned by agreement. In addition, the low-order bits of these characters are 'do not care' bits, that is, the bits can be used by the programmer, provided that

the total character retains odd parity in ASCII and in EBCDIC. If the total character is turned to even parity (by insertion of bits into the 'do not care' area), it loses its control function.

Special Turnaround Sequences

Temporary Text Delay (STX/ENQ)

The Temporary Text Delay (TTD) sequence is transmitted by the master station if it is remporarily unable to continue transmission of the next block. TTD is transmitted two seconds from the receipt of the last acknowledgement from the slave station if the next data block is not ready for transmission by that time. The normal response of the slave to TTD is NAK. After receipt of the NAK response, the master station may repeat the sequence at 2-second intervals if data is still not ready for transmission. Either the master or slave station may terminate by transmitting EOT after a predetermined number of tries. The TTD sequence can be used in text mode and in control mode.

Reverse Interrupt (DLE/RVI)

The RVI character is a reply which indicates that the previous message was received error free; however, RVI requests that the data transfer is reversed so that the receiving station is given an opportunity to send an important message. The recipient of an RVI must then provide storage space into which the message from the other station may enter. RVI is commonly used in non-conversational data exchange where one station primarily receives messages and refrains from transmitting data other than acknowledgements. When the slave sends RVI, the master answers with ACK, whereupon the master-slave relationship is reversed. The RVI sequence can only be used outside of text mode.

ENDING SEQUENCES

Mandatory Disconnect (DLE/EOT)

The mandatory disconnect sequenct (DISC) is used on switched networks only. When sent or received, it causes both stations to terminate and disconnect the call. The sequence is not valid if it is received or transmitted in text mode.

Abort Procedure

It is possible that a station is forced to stop transmission owing to a not-ready condition in one of its I/O devices. Such halts, which can be caused, for example, by card jams, a forms check, or an empty hopper, interfere with BSCA operations. This gives rise to a so-called abort situation, in which the station with the not-ready condition informs the other station that transmission must be terminated.

The master station normally aborts by initially sending the TTD sequence, STX/ENQ. On receipt of a NAK response from the slave station, the master station then terminates by transmitting EOT or, on switched networks, the mandatory disconnect sequence, DLE/EOT. An abort initiated by the master station is defined as a forward-acting abort.

It is possible, however, that the abort situation arises in the slave station. In this instance, the slave must send EOT (or DLE/EOT on switched lines) after it has received the ending character (ETB or ETX) of the current message. When initiated by the slave station, an abort is defined as reverse-acting.

After an abort, the transmission is terminated by issue of a halt instruction.

Initialization Sequence

The initialization sequence is used to bid for the line and is actually an indirect control sequence since it only initiates data exchange. The sequence consists either of an ENQ character alone which is interpreted as "Who are you?", or of some identification followed by NEQ which is interpreted as "I am xxx, who are you?" The sending of an initialization sequence does not set the BSCA to text mode.

Full Transparency Feature

The Full Transparency feature allows the BSCA to transmit (free of restriction) any kind of binary data. The feature is required for the transmission of packed decimal data, entire programs, random data, and so on. Transmission of this binary data is accomplished by making all control characters available for use as data; thus, a message becomes transparent.

The feature is activated by sending or receiving the starting sequence DLE/STX. This sequence places the BSCA into data mode and

into transparent mode at the same time. When transparent mode is set, the following characteristic changes occur:

1. The BSCA recognizes neither individual control characters nor control sequences as anything but data with no other associated function.

2. All inserted SYN characters are automatically preceded by a DLE character.

3. A second DLE is automatically attached to every 'data-DLE' to mark the data-DLE as such. This second DLE, and the inserted DLE/SYNs, are automatically deleted upon reception and do not enter main storage. (Therefore, the program must not insert SYN characters as 'fillers' as these will enter main storage.)

To leave transparent mode, the following ending sequences are available: DLE/ETX, DLE/ETB, DLE/ITB, and DLE/ENQ. These sequences are recognized as escape signals only when they coincide with two specific values in the field length counter of the transmitting station. DLE must coincide with the counter value 11 (decimal) and the associated ETX, ETB, ITB, or ENQ must coincide with the counter value 10 (decimal). As Figure 3-13 shows, a transparent message may be of any length up to 4,095 bytes; however, due to the fixed location at which the escape sequence must appear, a reply of not greater than nine-bytes length is required. Longer replies, such as used in limited conversational mode, are not permited, but a conversational reply limited to nine bytes in length is allowed.

All escape sequences (except DLE/ITB) that are used to leave transparent mode cause the BSCA to return to control mode. In addition, all of them (except DLE/ITB) cause turnaround. DLE/ITB is used to end a transparent block when intermediate block checking is enabled together with the full transparency feature. When transparent intermediate blocks are to be transmitted, there must be a Transmit and Receive instruction for each transparent block.

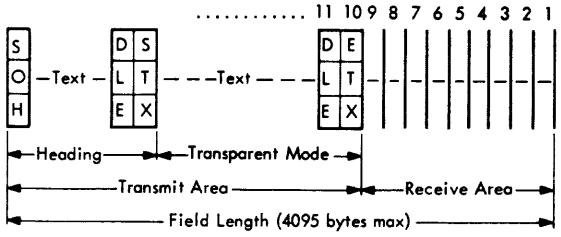


Figure 3-13. Transparent Message

Each transparent intermediate block is started with DLE/STX and ended with DLE/ITB. After each transparent intermediate block, the

transmitting BSCA sends SYN characters for 3 seconds to retain synchronization. To avoid timeout, the transmitter must send the next block before 3 seconds have elapsed.

The sequence DLE/SOH does not set transpatent mode; it is accepted strictly as SOH. The text of the heading must not end with DLE when the adjacent text is to be sent in transparent mode. If the heading ends with DLE, the transparency start signal is changed into DLE/DLE/STX; the BSCA expects STX or ITB directly after DLE (since it does not know where the heading ends) and, therefore, DLE/DLE/STX is not recognized.

The sequence DLE/EOT is not a transparent sequence. DLE/EOT is the disconnect sequence for switched networks.

SENSE INFORMATION

The sense information is a collection of check bits which indicates the type of error that occurred. These bits are available at final interrupt time and must be moved from auxiliary storage (where they are generated) to a main-storage position where they can be examined by the program. The Store Sense Information instruction is designed explicitly for the retrieval of this information. After the execution of this instruction, or after acceptance of the next XIO instruction, the sense information is reset (see BSCA Sense Byte Table in Appendix L.)

CHECK BITS

The individual bits are described in the following text.

Bit 0--Transmit/Receive Mode Error

Bit 0 indicates whether the error occurred during transmission or during reception. When this bit is present (binary 1), the error occurred during receive operations. When it is not present (binary 0), the error occurred in transmit operations.

Bit 1--EOT Received

Bit 1 indicates that an EOT has previously been received but could not be stored because no corresponding receive-type XIO instruction was issued in time. The bit is then set by the next transmit- or receive-type XIO instruction (except ADPREP) and reset by subsequent XIO instructions.

Bit 2--CRC/LRC/VRC Check

Bit 2 is set when either the Cyclic Redundancy Checking (CRC) circuits or the Longitudinal Redundancy Checking (LRC) circuits have detected an error. Both checking circuits can be installed in the BSCA. The checking circuits are activated in both stations with each starting sequence (SOH, STX) and they accumulate a block check character by way of arithmetic operations. After each ending sequence (ETX, ETB, or ITB), the transmitter sends its Block Check Character (BCC) to the receiver where it is compared with the BCC generated by the receiver. Thus, only the receiver detects CRC/LRC errors. These errors are caused by noise on the telephone line. (The efficiency of a noisy telephone line can be greatly increased when messages are divided into blocks that reduce the error probability by way of smaller individual lengths.)

When ASCII is used as the basic code and the full transparency feature is not installed, bit 2, which is set in the receiving BSCA only, indicates a vertical or longitudinal redundancy error (invalid ASCII character) detected during receive operations.

Bit 3--Timeout

Bit 3 is turned on in the receiving BSCA only when the BSCA is in receive mode and if either:

1. The BSCA does not receive a SYN character, interspersed at the proper time interval.

2. The BSCA does not receive a turnaround sequence before 3 seconds have elapsed.

3. The BSCA receives constant SYN characters for 3 seconds.

Bit 4--Short Record

Bit 4 indicates that the field length specification of a Transmit and Receive instruction is smaller than the size of the combined output/input field in main storage that is addressed by the instruction. During processing of an instruction with insufficient field length, the field length counter in the BSCA runs down to zero prior to the detection of the turnaround character (ETX or ETB).

The short record bit is also turned on under the following circumstances:

1. When the field length specification is correct but the turnaround character is either not detected or missing.

2. If the escape sequence in transparent data does not coincide with field length counter values 11 (decimal) and 10 (decimal).

3. If a dial digit is greater than 9.

Bit 5--Storage Wraparound

Bit 5 is turned on when the combined output/input field (addressed by a Transmit and Receive the upper limit of the main storage of the control unit. The upper limit is 8,191, beyond this limit, storage wraparound occurs.

Bit 6--Parity Check

Bit 6 is turned on when a parity error is detected within the data path of the BSCA. This parity error requires the attention of the customer engineer.

Bit 7--Overrun

The overrun bit is set in the event that a BSCA malfunction prevents the microprogram from servicing BSCA requests; this fault must be corrected by a customer engineer.

BSCA REJECT CONDITIONS

To avoid program error stops during BSCA operation, the following basic requirements must be met:

1. The BSCA must be enabled (to supply the 'data terminal ready' signal for the modem).

2. The modem must have power on and must not be in a test state (so that it can supply the 'data set ready' signal for the BSCA).

3. Instructions must not be issued for optional features which are not installed.

The BSCA instructions must be located on even boundaries in main storage, and addresses must neither violate the protected area nor exceed storage capacity. Field length specifications in excess of 4,095 bytes can be stated; however, only the 12 low-order bits of the field length are used.

Figure 3-14 shows the reject conditions of the BSCA.

Affected Instructions	Cause of Rejection	Result
Receive Initial	1. On leased or private lines: Modem power off 2. On switched networks: Line connection not yet established	Condition code 00 (available)
Transmit and Receive, Receive, or Transmit, Address Prepare	Modem not operational (Data Set Ready off)	Condition code 3 (not operational)
All XIO instructions	BSCA not enabled (Data Terminal Ready off)	Condition code 3
	BSCA busy with processing of previous XIO instruction	Condition code 1 (working). See Note
Receive Initial, Transmit and Receive, Receive, or Transmit	EOT/DLE-EOT sequence received when BSCA not busy	Condition code 3 (not operational) and sense bit 1 are set
Transmit and Receive, or Transmit	Data Set Ready and character phase in receive mode	Enter I/O-busy loop, wait for character phase off, then execute instruction
Receive, or Receive Initial		Check-condition indicator on, sense bits 4 and 0 are set, and the instruction is executed
All CIO and TIOB instructions, except Test on Busy and Store ITB Address	BSCA busy with processing of previous XIO instruction	Enter I/O-busy loop, wait for end of busy condition, then execute instruction

Figure 3-14. BSCA Reject Conditions

ERROR CHECKING METHODS

Three error-checking methods are used in the BSCA: CRC, LRC, and VRC. The methods are used in the following circumstances:

Circumstances	Checking Methods
When EBCDIC is used as the transmission code.	CRC
When ASCII is used as the transmission code and the full transparency feature is not installed in the BSCA.	LRC and VRC
When ASCII is used as the transmission code and the full transparency feature is installed in the BSCA.	CRC

PAD CHECKING

To ensure that the last bits of a message are correctly transmitted, a pad character is sent immediately following each ending code or sequence. In the case of EOT, NAK, ENQ, and DLE/Column 3, the trailing pad character is automatically checked to ensure that the first four bits are 1-bits. If so, the ending code is accepted as correct and turnaround takes place. However, if the trailing pad does not present four initial 1-bits, the BSCA ignores the ending code and continues to transfer data into core storage.

BSCA CONSOLE

The BSCA console (Figure 3-15) is located adjacent to the control unit console and contains all indicator lights that are required to show the current state of the BSCA.

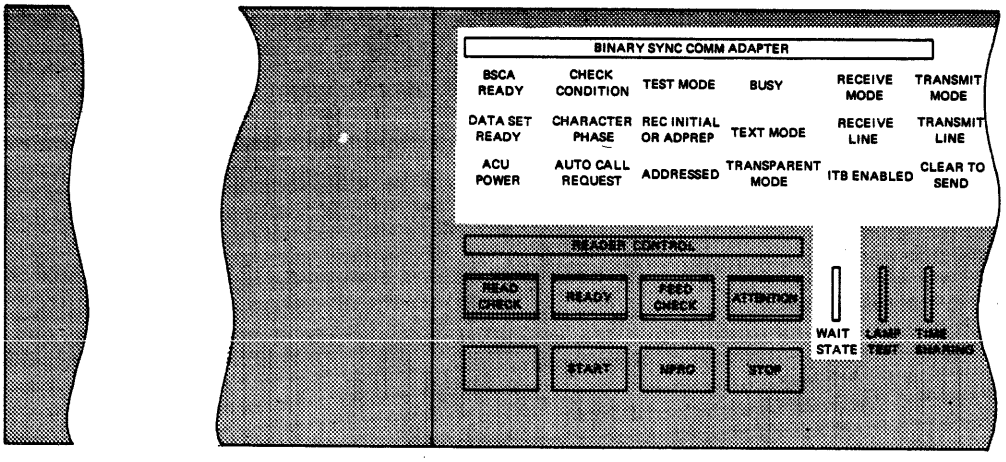


Figure 3-15. BSCA Console

INDICATOR LIGHTS

The following indicator lights are provided on the BSCA console.

BSCA Ready

Indicates that the BSCA has been enabled.

Data Set Ready

Indicates that the modem has line connection and is prepared to operate.

Check Condition

Indicates that any of the testable BSCA indicators has been set by an abnormal condition.

Character Phase

Indicates that the BSCA has achieved synchronization with another station.

Test Mode

Indicates that the test switch located in the interface plug is in the test position, or that the CE test switch is on. In test mode, the BSCA operates in a closed loop with the control unit simulating a remote station.

Rec Initial or ADPREP

Indicates that the BSCA is awaiting an initial sequence.

Busy

Indicates that the BSCA is processing an XIO instruction.

Text Mode

Indicates that a control character (or sequence) has set the BSCA to text mode.

Transparent Mode

Indicates that the BSCA has been set to transparent mode.

Receive Mode

Indicates that the BSCA is prepared to (or has turned around to) listen.

Receive Line

Indicates that incoming information is on the line.

ITB Enabled

Indicates that the BSCA has been set to the intermediate block checking mode.

Transmit Mode

Indicates that the BSCA is prepared to (or has turned around to) transmit.

Transmit Line

Indicates that outgoing data is on the line.

Clear to Send

Indicates that the modem has responded to a request to send, via its ready-for-sending interface line. This light must remain on during transmission; if it goes out, a modem or line failure is indicated.

ACU Power, Auto Call Request, Addressed

Not used by the programmable terminal.

WAIT STATE SWITCH

The Wait State toggle switch on the BSCA console can be flipped to WAIT STATE at any time to prepare the control unit for entering this state. The control unit enters wait state when either a Receive Initial or an Address Prepare instruction has been accepted; this is indicated by REC INITIAL OR ADPREP. The wait state is characterized by the control unit being held in a loop with power on. In wait state, no I/O or processing operation can be started unless the Wait State switch is moved to the central (that is, normal) position. However, the control unit is ready to receive a call, and starts when this comes in.

BSCA LOOP TEST

The first step in tracing teleprocessing errors is to find out which part of the data link is at fault. The defect may be at the transmitting station, at the receiving station, or in the line.

A convenient method of narrowing down the possibilities is to test the BSCA first by running the BSCA loop test; this enables the operator to establish whether his BSCA is causing trouble. If the operator's BSCA is faulty, an IBM service representative If the operator's BSCA is not at fault, the trouble may be in the line, in the modem, or in the remote data terminal. In this instance, the operator res onsible for the remote data terminal should be asked to perform the loop test at the remote terminal. I the BSCA and the remote data terminal operate successfully, the engineer responsible for the modem and/or the line should be called.

The BSCA loop test is carried out by a program that is loaded into core storage from a card deck. Only the BSCA circuits are tested, not the modem or the line. The card deck is available from IBM service representatives.

OPERATING PROCEDURES

To operate the loop test, proceed as follows:

1. Set loop test switch at cable connector of data set to TEST; test mode indicator on BSCA panel lights. This shows that the BSCA is set to operate in a closed loop, with the control unit simulating a remote station.

2. Set Timeout Inhibit switch on CE console to up position. (The CE console is located behind a cover at the extreme left of the console. The Timeout Inhibit switch is a toggle switch in the lower right-hand corner of this console.)

3. Set Test switch on CE console to up position. (The test switch is located immediately to the left of the Timeout Inhibit switch.) The test mode indicator remains lit.

4. Set data switches on console to 00, and set address/register data switches to 009C.

5. Turn off (central position) Wait State switch on BSCA console.

6. Turn on (down position) Time-sharing Switch on console.

7. Place card deck in hopper of primary load device and make device ready.

8. Make printer ready.

9. Press Program Load key on console.

10. Carry out actions according to Figure 3-16

Note: In general, the test program is activated in stages by successive operation of the Start key. At the end of each stage, the operator reads the console display and performs the appropriate action as shown in Figure 3-16. After the last stage, the operator is informed of the result of the test by the printout "NO ERRORS DETECTED' or 'ERRORS DETECTED IN ROUTINE', followed by the numbers of the incorrectly functioning routines.

E-S-T-R Display	Meaning or Action Required
0F01	Press Start key
0F03	Card deck may be incomplete. Run test with backup deck. If halt also occurs with backup deck, call for service
0F04	Printer error
0F05	Reader error
0AA1	1. Set Timeout Inhibit switch to down position 2. Press Start key
0AA2	1. Set Timeout Inhibit switch to up position 2. Press Start key
0BB1	1. Set Test switch to down position 2. Press Start key
0BB2	Press Start key
0999	End of test 1. Check printout 2. Set Loop Test switch at cable connector of data set to OPERATE 3. Set Timeout Inhibit switch and Test switches to down position 4. Close CE panel

Note: These stops may not all come up during the test, but this does not mean that the test is working incorrectly

Figure 3-16. Programmed Stops in the BSCA Loop Test

INTERFACES

The term "interface" pertains to the line of demarcation between the BSCA and a connected external equipment. The interface is located at a male plug on the cable between the BSCA and the equipment. Because telephone line facilities differ in design and quality and are often supervised or operated by government authorities, various rules and standards are applicable, depending on the country in which the BSCA is to operate.

To cater for these variations, the conversion of the bipolar BSCA signals into frequency-encoded signals suitable for the particular telephone network is made by an external equipment known as a data set, a data station, a line adapter or a modem. (In this publication, the term "modem" is used for these equipments). Modems are likewise subjected to various rules and standards which the BSCA interface is designed to meet.

The BSCA can operate at transmission speeds of from 2,000 bits per second (bps) to 7,200 bps.

The BSCA interface complies with the Electronic Industry of America (EIA) standard RS 232B.

The BSCA interface is described in the following text so that modem designers can assess the minimum effort required to obtain compatibility. Signal descriptions include recommendations giving modem requirements.

BSCA INTERFACE

Figure 3-17 shows the plug pin allocations, and the circuit names of the interface.

FUNCTIONAL HIGHLIGHTS

- The transmission speed is determined by the modem, within a range of 2,000 to 7,200 bps. (Designer's choice).
- The modulation method is by designer's choice.
- Privately-owned or leased two-wire or four-wire lines may be used.
- Switched two-wire lines may be used in all cases where telephone companies do not supply the modem.
- The connection between stations is point-to-point only.

SIGNAL DESCRIPTION

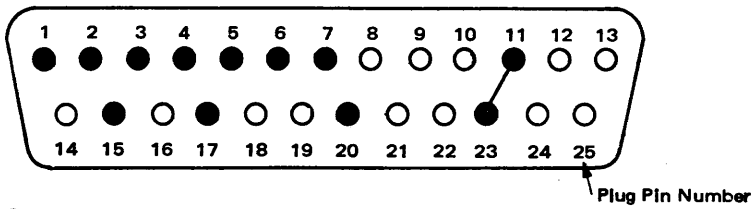
Protective Ground or Earth

'Protective ground or earth' is a conductor that is bonded to the terminal control unit's frame. In the modem, this conductor should be bonded to the modem frame and to all other metallic structures where dangerous voltages may occur (power or line transformers, shields, etc).

Recommendation: 'Protective ground or earth' should normally have no connection with signal ground; provision should be made, however, for such a connection in case local safety regulations or signal interference need it. 'Protective ground or earth' may be omitted from modems that are fully enclosed in an insulated housing.

If the conductor is used, it should be connected to the same ground to which the terminal control unit is connected. Undergrounded neutral lines must not be used as protective ground.

BSCA-to-Modem Plug



● Pins used
○ Pins not used

Interface Lines

Plug Pin No.	EIA Ident	EIA Circuit Name	CCITT Ident	CCITT Circuit Name	Direction
1	AA	Protective Ground	101	Protective Ground or Earth	–
2	BA	Transmitted Data	103	Transmitted Data	To Modem
3	BB	Received Data	104	Received Data	From Modem
4	CA	Request to Send	105	Request to Send	To Modem
5	CB	Clear to Send	106	Ready for Sending	From Modem
6	CC	Data Set Ready	107	Data Set Ready	From Modem
7	AB	Signal Ground	102	Signal Ground or Common Return	–
11	CH	Data Signaling Rate Selector	111	Data Signaling Rate Selector	To Modem
15	DB	Transmitter Signal Element Timing	114	Transmitter Signal Element Timing	From Modem
17	DD	Receiver Signal Element Timing	115	Receiver Signal Element Timing	From Modem
20	CD	Data Terminal Ready	108	Data Terminal Ready	To Modem
23	CH	Data Signaling Rate Selector	111	Data Signaling Rate Selector	To Modem

Figure 3-17. BSCA Interface

Signal Ground or Common Return

'Signal ground or common return' is a conductor that establishes the common reference potential for all BSCA-interchange circuits except 'protective ground or earth'. A connection between 'signal ground' and 'protective ground' cannot be made in the BSCA. The modem must use 'signal ground' as the common reference potential for all its interchange circuits to form a common return for both units (BSCA and modem), because both are fed by their own power supplies.

Data Terminal Ready (Direction: To Modem)

The 'data terminal ready' interchange circuit is at up level (in reference to 'signal ground') when the BSCA is operational. (The terms "up level" and "down level" are defined later in this manual n "Low-Speed Interface Drivers and Terminators"). The on-indication (up level) of the circuit is given when the 'enable BSCA' program instruction is issued. The off-indication (down level) is given under any of the following conditions:

1. 'Disable BSCA' instruction issued.
2. System reset occurs.
3. Control character sequence DLE-EOT either sent or received.
4. Interface test switch set to TEST.

Recommendation: If the modem is designed for privately-owned or leased line application, the use of 'data terminal ready' may be omitted because such lines are permanent connections which require no switching.

If the modem is designed for switched line application, 'data terminal ready' is the signal which the BSCA uses to control the disconnection of the line. 'Data terminal ready' should therefore be used as the prerequisite signal without which the modem would not be capable of seizing the telephone line. The signal should not, however, initiate the switching action itself; the line transfer (from telephone set to modem) should be initiated by a switch or contact. The switching action must be initiated at the calling station before the BSCA of the called station enters the 3-second receive timeout period. The modem should, however, release the line to the telephone set whenever 'data terminal ready' has down level.

Data Set Ready (Direction: From Modem)

The BSCA interprets a down level or a floating condition on the 'data set ready' interchange circuit as an indication that the modem is not operational. Any BSCA data-transfer instruction (such as 'transmit only', 'receive only', or 'transmit and receive') is rejected by the BSCA when 'data set ready' is off. Only the 'receive initial' instruction is executed independently of this signal.

Recommendation: If the modem is designed for privately-owned or leased line application, 'data set ready' should be at up level as long as the modem has power on. The signal should be at down level only if the modem has a reset condition or is in a special test mode (if applicable).

If the modem is designed for switched line application, 'data set ready' must be at down level until the line connection and the identification routines (if any) have been completed successfully. Because the BSCA controls the disconnection of the line (on switched line networks), 'data set ready' should be at down level after the line has been released to the telephone set (due to 'data terminaly ready' off). The down level on 'data set ready' is then taken as a confirmation of the line disconnect action.

Request to Send (Direction: To Modem)

The BSCA uses the 'request to send' interchange circuit to control the transmit status of the modem. 'Request to send' is at up level when one of the following instructions has been issued by the program:

1. Transmit only.
2. Transmit and receive.
3. Enable BSCA (provided a four-wire connection is available).

'Request to send' changes to down level whenever any turnaround characters or sequences (such as ETX, ETB, ACK, NAK) are found during transmission and the block check character and trailing pad character have been sent. 'Request to send' is also down when either the 'receive only' or 'receive initial' instruction has been issued.

Recommendation: If the modem is designed for half-duplex operation over two-wire lines, the modem should be in the receive status whenever 'request to send' is down. The modem should clamp its receiver whenever the signal has up level; a logical interlock against control errors is not required because the BSCA is responsible for line discipline.

If the modem is designed for dual simplex operation on four-wire lines or for full-duplex operation on two-wire lines, no clamping is required. The modem may then omit the use of 'request to send' and may assume the circuit to be on permanently.

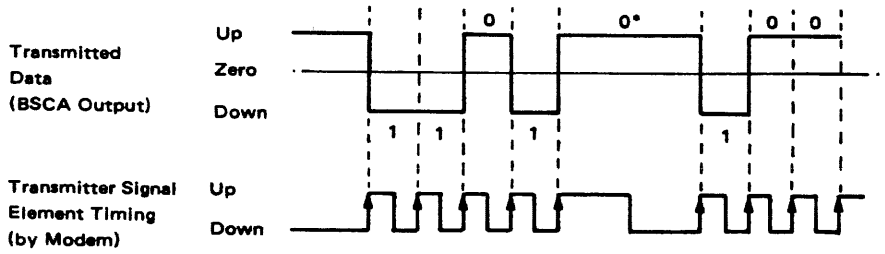
Ready for Sending, and Clear to Send. (Direction: From Modem)

The BSCA interprets an up level on the 'ready for sending' ('clear to send') interchange circuit as a confirmation to a 'request to send' signal that has been received previously. If 'ready for sending' is not up, the BSCA cannot begin to transmit.

Recommendation: The modem should indicate 'ready for sending' after an appropriate delay, during which the modem receiver is clamped and its transmitter is made operational (clamp removed, carrier on, etc). To ensure a reasonable turnaround delay, the elapsed time between the reception of 'request to send' and the indication of 'ready for sending' should be no longer than 80 milliseconds(ms). The shortest delay time should be determined by the time required to safely detect a received line signal in the modem of the remote station. The 'ready for sending' delay should always be longer than the on-delay of the carrier detector of the remote modem.

The 'ready for sending' circuit should have down level whenever 'request to send' or 'data terminal ready' is down or the connection is disrupted.

If the modem operates on two channels (four-wire lines) and keeps a steady carrier on the transmission line, a switching delay is not needed and 'ready for sending' may be on at all times that the line connection exists. This recommendation applies also to modems designed for duplex transmission.



* BSCA holds bit level until timing proceeds

Figure 3-18. Principle of Transmitted Data Timing

Transmitter Signal Element Timing. (Direction: From Modem)

The BSCA requires a bit timing strobe that gates out each individual bit. A transition from off (down level) to on (up level) on the 'transmitter signal element timing' interchange circuit (Figure 3-18) causes the BSCA to present either an up level (for a logical zero-bit) or a down level (for a logical one-bit) on the 'transmitted data' interchange circuit. The BSCA holds each bit-level until the next transition from down to up level occurs on the 'transmitter signal element timing' circuit.

Recommendation: The repetition rate of the timing strobes (that is, the transmission speed) is at the discretion of the modem designer and may be from 2,000 bps to (and including) 7,200 bps.

The up level of each strobe should be mtaintained for at least 200 nanoseconds (ns), and the down level should be maintained for a least 500 ns. At the beginning of any transmission, the BSCA sends a leading pad character that consists of eight one-bits (Figure 3-19). The leading pad character is then followed by two SYN characters which provide a total of eight level transitions for the purpose of establishing character synchronization for the receiving BSCA.

Transmitted Data. (Direction: To Modem)

The BSCA uses the 'transmitted data' interchange circuit to deliver its binary-encoded messages for transmission over the telephone line. An up level denotes a binary zero, and a down level denotes a binary one. Data is delivered to the modem at a rate of one bit per strobe, the strobe being generated by the modem.

Recommendation: The method of modulation is by designer's choice. Theoretically, any proven method (such as coherent or noncoherent frequency modulation, 2-phase or 4-phase modulation, pulse code modulation) may be used, provided that the following errors cannot occur:

1. Loss of synchronization when continuous logical zero-or logical one-bits are transmitted.

2. Loss of synchronization when alternating zero/one- or one/zero-bits are transmitted.

3. Bit inversion caused by line noise when continuous zero-or one-bits are transmitted.

If such errors are inherent in the modulation method chosen, they should be compensated for by appropriate means. The overall signal distortion rate should be less than 15%.

Should the BSCA fail to place a defined level on the 'transmitted data' interchange circuit (disrupted connection) while 'request to send' is up, the modem should operate as if logical one-bits are to be transmitted.

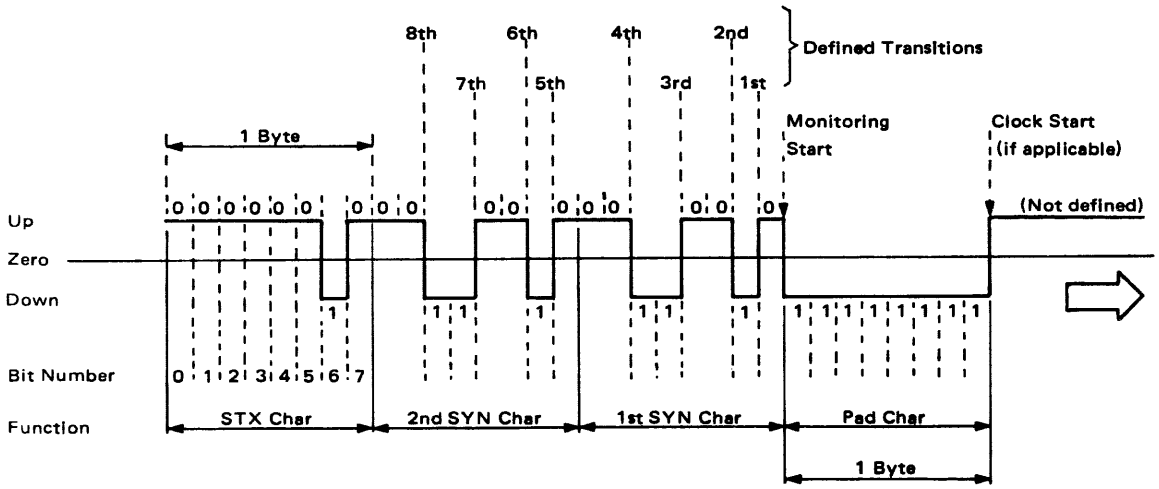
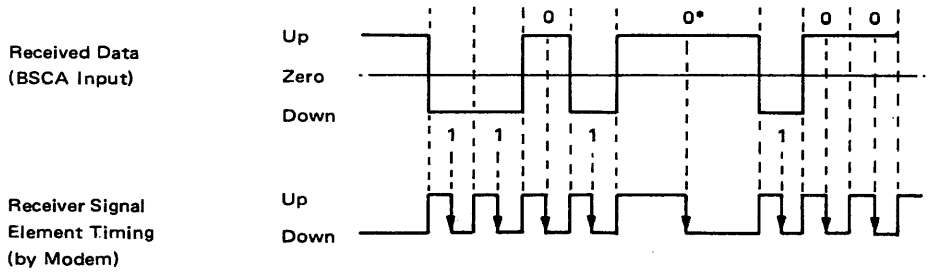


Figure 3-19. Synchronization Information Ahead of Message

Receiver Signal Element Timing. (Direction: From Modem)

The BSCA derives bit timing from the 'receiver signal element timing' interchange circuit. A transition from on (up level) to off (down level) is taken as the nominal centerpoint of a bit time, ind pendent of transmission speed (Figure 3-20). The BSCA uses the down transition to generate a strobe that gates the data offered on the 'received data' interchange circuit into the BSCA shift register.



*BSCA holds bit level until timing proceeds

Figure 3-20. Principle of Received Data Timing

Recommendation: The transition from on to off should be centered, up to ± 1/32 of a bit time, at any transmission speed. The synchronization between the modem transmit clock and the modem receive clock is by designer's choice. The two leading SYN characters which the BSCA provides for each message, as well as the inserted SYN characters, may be used for clock starting and clock correction. The clock should, however, remain stable for at least 1 second without synchronization information because the BSCA may send all zero-bits or all one-bits before the next SYN character is sent. Any additional synchronization information that the modem may introduce must neither alter the message nor appear on the 'received data' interchange circuit of the recipient station.

Strobe suppression during transmit operations is not required because the BSCA controls strobe acceptance.

Received Data. (Direction: From Modem)

The BSCA interprets an up level on the 'received data' interchange circuit as a logical zero-bit, and a down level as a logical one-bit. Data thus presented enters the terminal control unit only when one of the following instructions has been issued:

1. Receive initial.

2. Receive only.

3. Transmit and receive (after turnaround has occurred).

If non-valid data or nothing is received for 3 seconds, the BSCA ceases to accept data. Data is valid when at least one SYN character

is received per elapsed second or a turnaround character is received before 3 seconds have elapsed; however, a steady stream of SYN characters would make the data invalid.

Recommendation: The modem should present a down level on the 'received data' interchange circuit whenever no data is received while line connection is established. If the modem operates in half-duplex mode over two-wire lines, 'received data' should present a down level whenever 'request to send' is activated by the BSCA.

A transition on the 'received data' circuit must not occur simultaneously with a down transition on the 'receiver signal element timing' circuit.

Data Signaling Rate Selector. (Direction: to Modem)

The BSCA does not exert program-control over the 'data signaling rate selector' interchange circuit. This circuit is provided for modems which feature two selectable transmission speeds. With a jumper installed in the BSCA, a permanent up level is provided in the circuit; without this jumper, the circuit has a permanent down level.

Recommendation: Use of the 'data signaling rate selector' circuit is up to the designer. If the circuit is not used, the modem should be insensitive to either level that may be on the circuit.

INTERFACE DRIVERS AND TERMINATORS

BSCA INTERFACE DRIVERS AND TERMINATORS

The drivers and terminators used in the version 1 low-speed interface are described in the following text.

BSCA Driver

The BSCA driver (Figure 3-21) drives a 3-kiloohm load to signal ground (common return) at ± 5.0 volts (V), with the load being at the end of a 50-foot (15,24 meter) (maximum) multiwire cable. The cable harness is covered by a shield that represents the protective-ground conductor.

The driver circuit consists of an AND switch at the input and a transistor that translates the input levels to EIA-or CCITT-compatible levels without signal inversion.

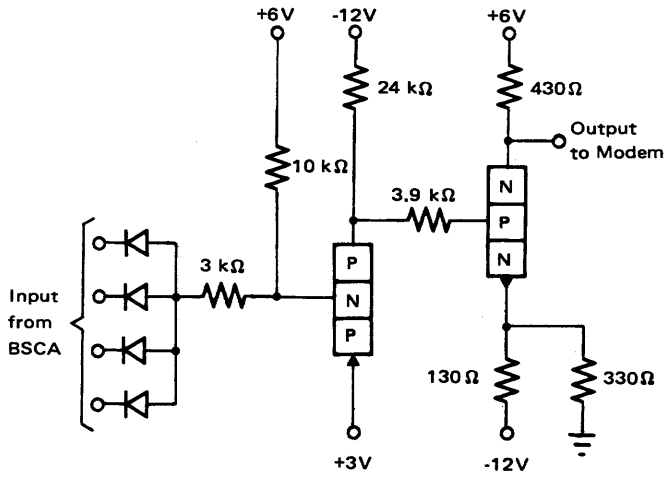


Figure 3-21. BSCA Driver Circuit

Characteristics: The characteristics of the driver are shown in Figure 3-22.

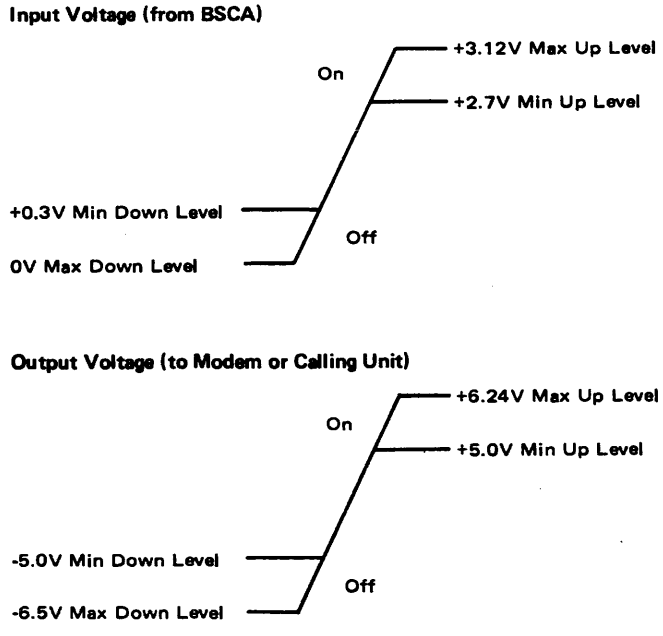


Figure 3-22. BSCA Driver Characteristics

Switching Delays: The following delays were measured with the circuit driving a 50-foot cable, with 2-volt maximum ground shift:

Turn-on delay = 1.5 microseconds (us) maximum.

Turn-on (fall) transition = 1.3 us maximum.

Turn-off delay = 3.4 us maximum.

Turn-off (rise) transition = 4.0 us maximum.

Note: The circuit is not designed to drive pure inductive loads (such as relays) directly. If relays are to be operated by an interchange circuit, a relay driver should be used as the terminator. The modem circuit should be sensitive to voltage only; signal overshoot or undershoot should not affect the modem.

BSCA Terminator

The BSCA terminator circuit (Figure 3-23) consists of an NPN transistor that inverts an EIA or CCITT signal applied between input and signal ground or common return.

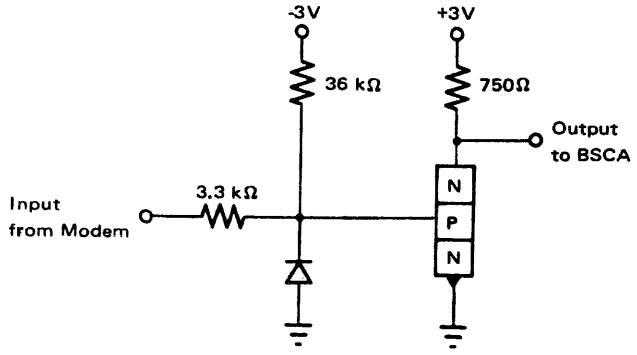


Figure 3-23. BSCA Terminator Circuit

Characteristics: The characteristics of the terminator are shown in Figure 3-24

Current Requirements: These are as follows:

0.84 milliampere (mA) minimum, 1.0 mA maximum at ± 3.0V.

5.0 mA minimum, 6.3 mA maximum at ± 25.0V.

Transitions: The following transitions were measured from 10 to 90% of rise or fall:

Turn-on (fall) transition = 500 ns maximum.

Turn-off (rise) transition = 200 ns maximum.

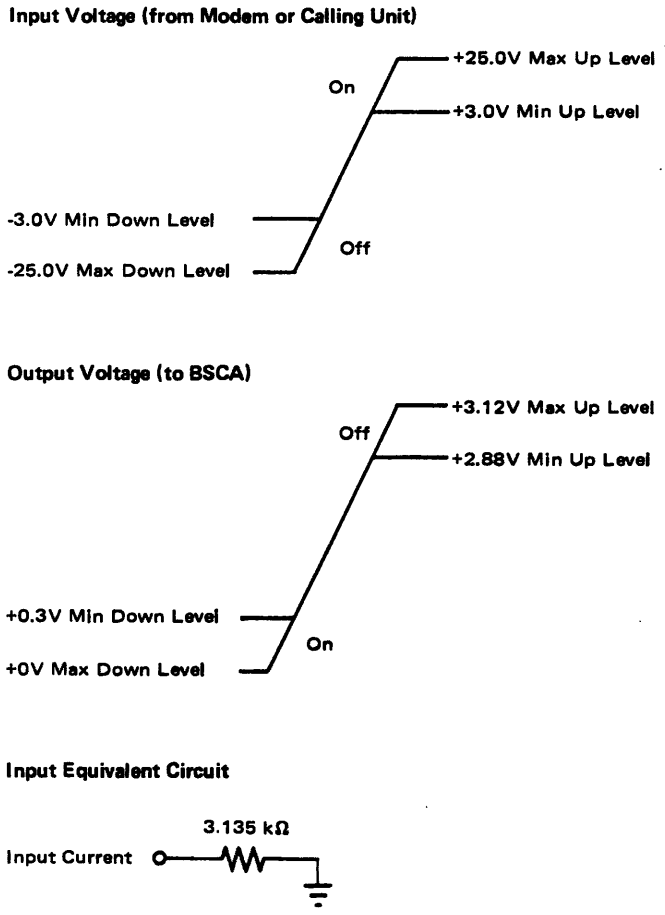


Figure 3-24. BSCA Terminator Characteristics

PLUGS AND RECEPTACLES

BSCA INTERFACE

The cable from the BSCA to the modem or the automatic calling unit is fitted with a 25-pin male plug, Cannon or Cinch type, part DB 19604-432 (IBM part 5302662). Figure 3-25 shows the dimensions of the plug.

The modem or the automatic calling unit should be fitted with a female receptacle, Cannon or Cinch type, part DB 19604-433 or equivalent.

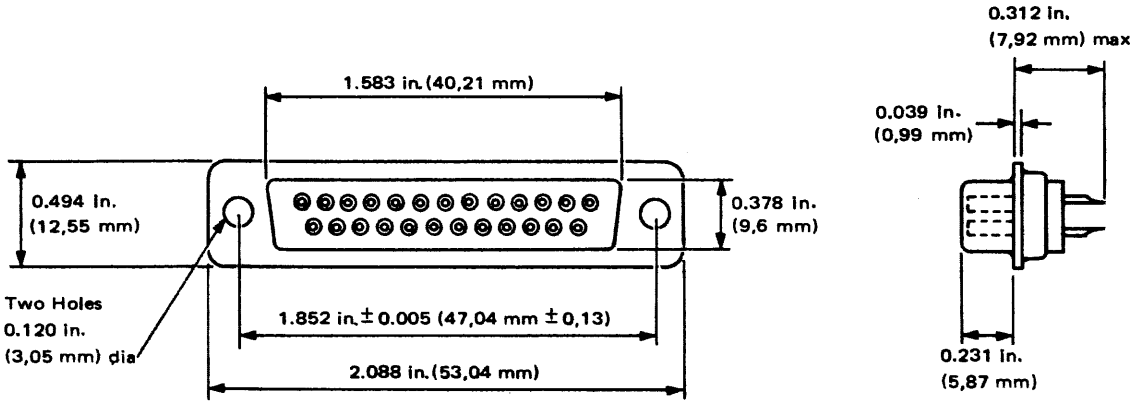


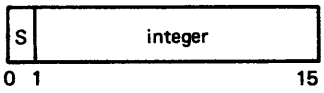
Figure 3-25. BSCA Interface Plug

APPENDIX A. FORMATS

DATA FORMATS

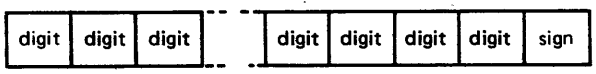
Binary Number

Halfword Binary Number



Decimal Numbers

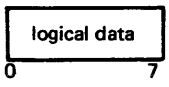
Packed Decimal Number



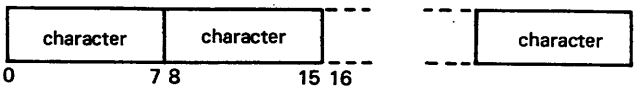
Zoned Decimal Number

Logical Information

Fixed-Length Logical Information

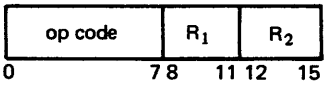


Variable-Length Logical Information

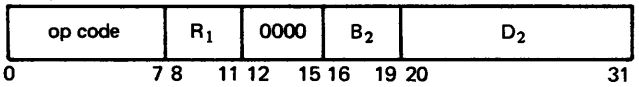


INSTRUCTIONS BY FORMAT TYPE

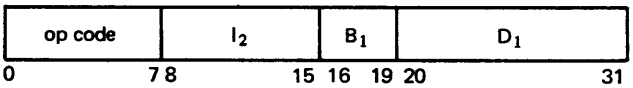
RR Format



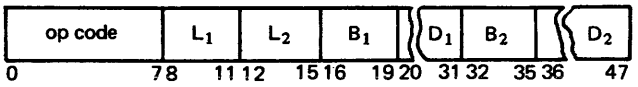
RX Format



SI Format

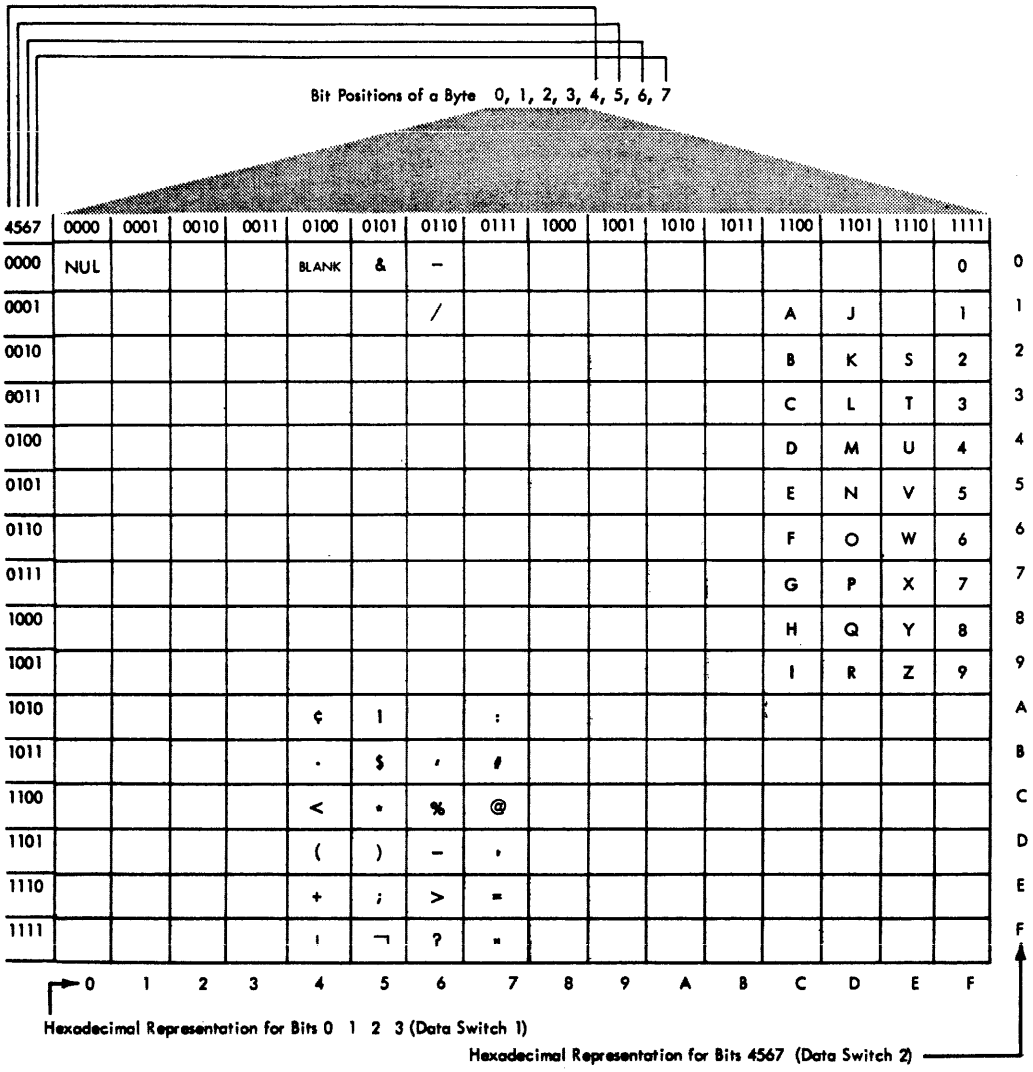


SS Format



APPENDIX B. EBCDIC AND ASCII CHARTS

Extended Binary-Coded-Decimal Interchange Code (EBCDIC)

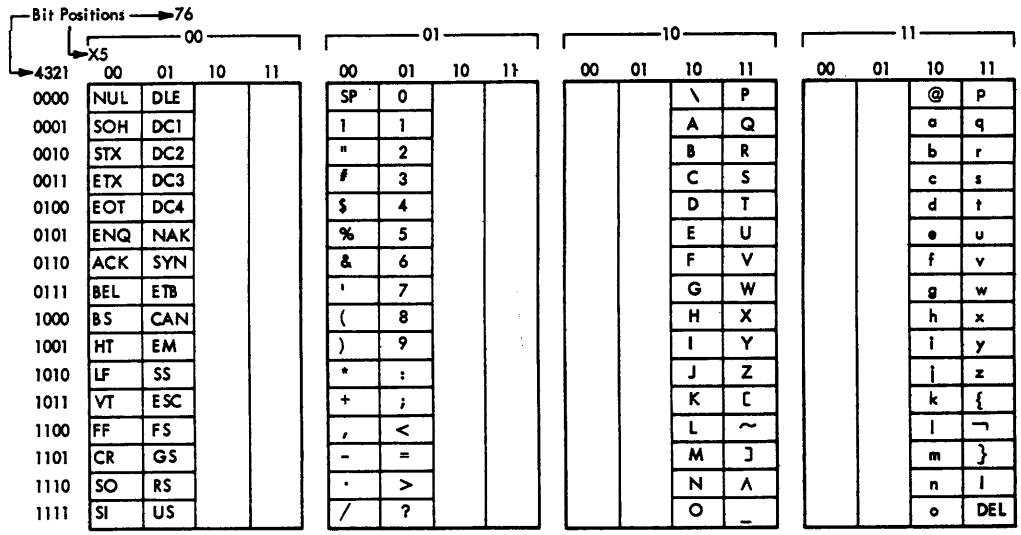


4567	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
* 0000	NUL				BLANK	&	-									0	0
0001							/						A	J		1	1
0010													B	K	S	2	2
0011													C	L	T	3	3
0100													D	M	U	4	4
0101													E	N	V	5	5
0110													F	O	W	6	6
0111													G	P	X	7	7
1000													H	Q	Y	8	8
1001													I	R	Z	9	9
1010					¢	!		:									A
1011					.	$	,	#									B
1100					<	*	%	@									C
1101					(	)	_	'									D
1110					+	;	>	=									E
1111					¦	¬	?	"									F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Hexadecimal Representation for Bits 0 1 2 3 (Data Switch 1)

Hexadecimal Representation for Bits 4567 (Data Switch 2)

American National Standard Code for Information Interchange (ASCII)
Extended to Eight Bits



Bit Positions → 76 / X5 / 4321	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NUL	DLE			SP	0					\	P			@	P
0001	SOH	DC1			!	1					A	Q			a	q
0010	STX	DC2			"	2					B	R			b	r
0011	ETX	DC3			#	3					C	S			c	s
0100	EOT	DC4			$	4					D	T			d	t
0101	ENQ	NAK			%	5					E	U			e	u
0110	ACK	SYN			&	6					F	V			f	v
0111	BEL	ETB			'	7					G	W			g	w
1000	BS	CAN			(	8					H	X			h	x
1001	HT	EM			)	9					I	Y			i	y
1010	LF	SS			*	:					J	Z			j	z
1011	VT	ESC			+	;					K	[			k	{
1100	FF	FS			,	<					L	~			l	¬
1101	CR	GS			-	=					M	]			m	}
1110	SO	RS			.	>					N	^			n	~
1111	SI	US			/	?					O	_			o	DEL

APPENDIX C. POWERS-OF-TWO TABLE

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

APPENDIX D. BINARY AND HEXADECIMAL NUMBER NOTATION

BINARY NUMBER NOTATION

A binary number system, such as is used in the Programmable Terminal, uses a base of two. The concept of using a base of two can be compared with the base of ten (decimal) number system:

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

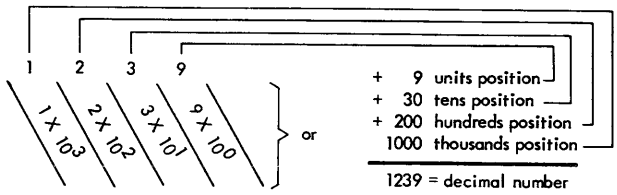


Figure D-1. Example of a Decimal Number

As shown in Figure D-1, the decimal number system allows counting to ten in each position from units to tens to hundreds to thousands, and so on. The binary system (Figure D-2) allows counting to two in each position. Register displays in the control unit are in binary form; a bit light on is a 'one'; a bit light off is a 'zero'.

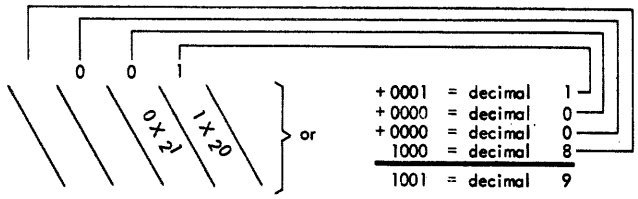


Figure D-2. Example of a Binary Number

HEXADECIMAL NUMBER SYSTEM

It has been noted that binary numbers require about three times as many positions as decimal numbers to express the equivalent number. This is not much of a problem to the control unit, however, in talking and writing or in communicating with the control unit, these binary numbers are bulky. A long string of 1's and 0's cannot be effectively transmitted from one individual to another. Some shorthand method is necessary.

The hexadecimal number system fills this need. Because of the simple relationship of hexadecimal to binary, numbers can be converted from one system to another by inspection. The base or radix of the hexadecimal system is 16. This means there are 16 symbols; 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F. the letters A,B,C,D,E, and F represent the 10-base system values of 10,11,12,13,14, and 15 respectively.

Four binary positions are equivalent to one hexadecimal position. The following table shows the compariable values of the three number systems:

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

At this point, all 16 symbols have been used, and a carry to the next higher position of the number is necessary.

16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14
21	0001 0101	15

and so on.

Remember that as far as the internal circuitry of the control unit is concerned, it only understands binary. But an operator can look at a series of lights on the console showing binary 1's and 0's, for example 0001 1110 0001 0011, and say that the lights represent the hexadecimal value 1E13 which is easier to state than the string of 1's and 0's.

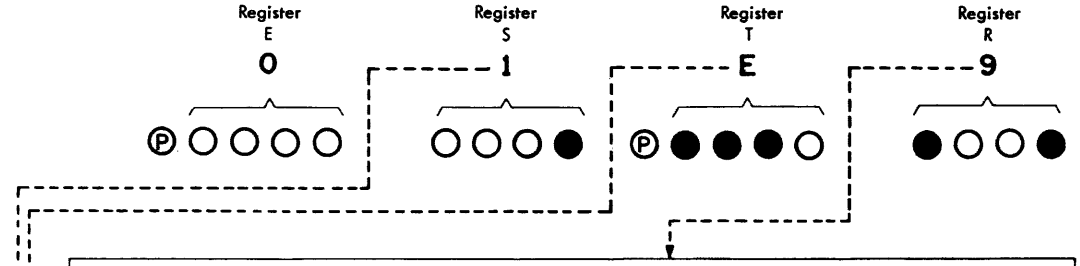
APPENDIX E. HEXADECIMAL-DECIMAL CONVERSION TABLES

The tables in this appendix provide for the direct conversion of decimal and hexadecimal number in these ranges:

Hexadecimal	Decimal
000 to FFF	000 to 4095

For numbers outside the range of the table, add the following values to the table figures:

Hexadecimal	*Decimal*
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 _	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01 _	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02 _	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03 _	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04 _	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05 _	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06 _	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07 _	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08 _	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09 _	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A _	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B _	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C _	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D _	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E _	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F _	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255
10 _	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11 _	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12 _	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13 _	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14 _	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15 _	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16 _	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17 _	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18 _	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19 _	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A _	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B _	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C _	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D _	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E _	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F _	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20_	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21_	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22_	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23_	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24_	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25_	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26_	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27_	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28_	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29_	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A_	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B_	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C_	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D_	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E_	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F_	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30_	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31_	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32_	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33_	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34_	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35_	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36_	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37_	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38_	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39_	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A_	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B_	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C_	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D_	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E_	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F_	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
40_	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41_	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42_	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43_	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44_	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45_	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46_	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47_	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48_	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49_	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A_	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B_	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C_	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D_	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E_	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F_	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50_	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51_	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52_	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53_	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54_	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55_	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56_	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57_	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58_	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59_	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A_	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B_	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C_	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D_	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E_	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F_	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

	0	I	2	3	4	5	6	7	8	9	A	B	C	D	E	F
60_	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61_	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62_	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63_	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64_	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65_	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66_	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67_	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68_	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69_	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A_	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B_	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C_	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D_	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E_	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F_	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
70_	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71_	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72_	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73_	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74_	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75_	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76_	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77_	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78_	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79_	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A_	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B_	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C_	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D_	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E_	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F_	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80_	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81_	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82_	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83_	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84_	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85_	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86_	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87_	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88_	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89_	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A_	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B_	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C_	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D_	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E_	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F_	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90_	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91_	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92_	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93_	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94_	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95_	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96_	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97_	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98_	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99_	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A_	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B_	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C_	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D_	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E_	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F_	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0 _	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1 _	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2 _	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3 _	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4 _	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5 _	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6 _	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7 _	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8 _	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9 _	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA _	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB _	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC _	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD _	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE _	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF _	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0 _	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1 _	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2 _	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3 _	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4 _	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5 _	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6 _	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7 _	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8 _	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9 _	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA _	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB _	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC _	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD _	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE _	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF _	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0 _	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1 _	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2 _	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3 _	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4 _	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5 _	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6 _	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7 _	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8 _	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9 _	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA _	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB _	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC _	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD _	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE _	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF _	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D0 _	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1 _	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2 _	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3 _	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4 _	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5 _	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6 _	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7 _	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8 _	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9 _	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA _	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB _	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC _	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD _	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE _	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF _	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E0_	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1_	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2_	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3_	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4_	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5_	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6_	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7_	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8_	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9_	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA_	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB_	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC_	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED_	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE_	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF_	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0_	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1_	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2_	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3_	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4_	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5_	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6_	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7_	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8_	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9_	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA_	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB_	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC_	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD_	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE_	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF_	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

APPENDIX F. OPERATION CODES

Operation Codes					
Name	Mnemonic	Op Code		Class	Format
		Hexadecimal	Binary		
Branch on Condition	BCR	07	0000 0111	Branching	RR
Branch and Store	BASR	0D	0000 1101	Branching	RR
Add	AR	1A	0001 1010	Binary	RR
Subtract	SR	1B	0001 1011	Binary	RR
Store Halfword	STH	40	0100 0000	Binary	RX
Branch on Condition	BC	47	0100 0111	Branching	RX
Load Halfword	LH	48	0100 1000	Binary	RX
Compare Halfword	CH	49	0100 1001	Binary	RX
Add Halfword	AH	4A	0100 1010	Binary	RX
Subtract Halfword	SH	4B	0100 1011	Binary	RX
Branch and Store	BAS	4D	0100 1101	Branching	RX
Test under Mask	TM	91	1001 0001	Logical	SI
Move	MVI	92	1001 0010	Logical	SI
Set PSW	SPSW	81	1000 0001	Branching	SI
And	NI	94	1001 0100	Logical	SI
Compare	CLI	95	1001 0101	Logical	SI
Or	OI	96	1001 0110	Logical	SI
Halt & Proceed	HPR	99	1001 1001	Logical	SI
Test I/O and Branch	TIOB	9A	1001 1010	Input/Output	IO
Control I/O	CIO	9B	1001 1011	Input/Output	IO
Transfer I/O	XIO	D0	1101 0000	Input/Output	IO
Move Numerical	MVN	D1	1101 0001	Logical	SS
Move Characters	MVC	D2	1101 0010	Logical	SS
Move Zone	MVZ	D3	1101 0011	Logical	SS
Compare	CLC	D5	1101 0101	Logical	SS
Edit	ED	DE	1101 1110	Logical	SS
Move with Offset	MVO	F1	1111 0001	Decimal	SS
Pack	PACK	F2	1111 0010	Decimal	SS
Unpack	UNPK	F3	1111 0011	Decimal	SS
Zero and Add	ZAP	F8	1111 1000	Decimal	SS
Compare Decimal	CP	F9	1111 1001	Decimal	SS
Add Decimal	AP	FA	1111 1010	Decimal	SS
Subtract Decimal	SP	FB	1111 1011	Decimal	SS
Multiply Decimal	MP	FC	1111 1100	Decimal	SS
Divide Decimal	DP	FD	1111 1101	Decimal	SS
Translate	TR	DC	1101 1100	Logical	SS

APPENDIX G. CONDITION CODE

Instruction	Condition Code Setting			
	00	01	10	11
Add	zero	<zero	>zero	--
Subtract	zero	<zero	>zero	--
Compare Halfword	equal	low	high	--
Add Halfword	zero	<zero	>zero	--
Subtract Halfword	zero	<zero	>zero	--
Zero & Add	zero	<zero	>zero	--
Compare Decimal	equal	low	high	--
Add Decimal	zero	<zero	>zero	overflow
Subtract Decimal	zero	<zero	>zero	overflow
Test under Mask	zero	mixed	--	all one
And	zero	not zero	--	--
Compare Logical	equal	low	high	--
Or	zero	not zero	--	--
Edit	zero	<zero	>zero	--
Transfer I/O*	available	working	channel busy	not operational
Control I/O**	available	working	--	not operational

* The Transfer I/O and the corresponding condition codes do not pertain to the Input/Output Channel, Native Tape Attachment, and Storage Control feature

** 2922-2 and 2152 Control Carriage Instruction only

APPENDIX H. TERMINAL TIMINGS

Arithmetic and Logical Operations

Processing Operation	Format	Mnemonic	Timing (average in microseconds with time-sharing switch ON)
Branch on Condition	RR	BCR	78
Branch and Store	RR	BASR	94
Add	RR	AR	174
Subtract	RR	SR	183
Store Halfword	RX	STH	117
Branch on Condition	RX	BC	110
Load Halfword	RX	LH	124
Compare Halfword	RX	CH	223
Add Halfword	RX	AH	209
Subtract Halfword	RX	SH	216
Branch and Store	RX	BAS	124
Test Under Mask	SI	TM	132
Move	SI	MVI	104
Set PSW	SI	SPSW	149
AND	SI	NI	123
Compare Logical	SI	CLI	122
OR	SI	OI	130
Halt and Proceed	SI	HPR	105
Move Numerical	SS	MVN	$137 + 16N$
Move Characters	SS	MVC	$137 + 16N$
Move Zone	SS	MVZ	$137 + 16N$
Compare Logical	SS	CLC	$126 + 24B$
Edit	SS	ED	$165 + 26N_1$
Move with Offset	SS	MVO	$170 + 10.2N_1 + 7.8N_2$ (if $N_1 > N_2$) $170 + 18.0N_1$ (if $N_1 \leq N_2$)
Translate	SS	TR	$148 + 53N$
Pack	SS	PACK	* $182 + 27N_2(\frac{N_2 + P_3}{2} - 1) + 16.8P_1 + 9.6N_1 - 4.8N_2$ * Use 182, if $2N_1 - (N_2 + 1)$ is > 0 Use 161, if $2N_1 - (N_2 + 1)$ is < 0
Unpack	SS	UNPK	$178 + 9.6N_1 + 8.4N_2 + P_4(5 + 4.4N_1 - 8.4N_2)$
Compare Decimal	SS	CP	$312 + 13.2N_2 + (N_2 - 1)(69 + 15P_8) + (N_1 - N_2)(38 + 9P_8) + 12P_6$
Add Decimal	SS	AP	$312 + 13.2N_2 + (N_2 - 1)(69 + 15P_5) + (N_1 - N_2)(38 + 9P_5) + 12P_6 + P_7(15 + 57N_1)$
Subtract Decimal	SS	SP	(The formula is the same for both instructions)
Multiply Decimal	SS	MP	$215 + 20N_2 + (N_1 - N_2)(70 + 9.6N_2) + \Sigma Opl_{dig}(39 + 9.6N_1 + 44N_2)$
Divide Decimal	SS	DP	$325 + 250N_2 + (N_1 - N_2)(150 + 250N_2) + \Sigma Q(100 + 50N_2)$
Zero and Add	SS	ZAP	$210 + 10.2N_1 + 13.2N_2$

Notes:
1. With time-sharing switch OFF, add 22 usec to each instruction.
2. When effective address generation (Indexing) is used, add 56 usec for each generated address.

Symbols used in the timing formulas:

N	= Number of bytes in field (defined by coupled field length)
N_1	= Number of bytes in the first operand
N_2	= Number of bytes in the second operand
B	= Total number of processed bytes in the first operand field
ΣQ	= Sum of quotient digits
ΣOpl_{dig}	= Sum of operand 1 digits

The following constants must be placed into the respective formulas in the positions indicated by P_1 through P_8:

P_1 = 1 if $2N_1 - (N_2 + 1)$ is odd 0 if $2N_1 - (N_2 + 1)$ is even	P_4 = 1 if $N_1 + 1 - 2N_2$ is <0 0 if $N_1 + 1 - 2N_2$ is ≥ 0	P_6 = 1 if $L_1 > L_2$ 0 if $L_1 = L_2$
P_2 = 1 if $N_1 = N_2 = 1$ 0 if other	P_5 = For Add: 1 if the signs of oprnd 1 and oprnd 2 differ 0 if the signs of both oprnds are alike For Subtract: 1 if the signs of both oprnds are alike 0 if the signs of oprnd 1 and 2 differ	P_7 = 1 if Recomplement occurs 0 if No Recomplement occurs
P_3 = 1 if N_2 is odd 0 if N_2 is even		P_8 = 1 if the signs of oprnd 1 and oprnd 2 are alike 0 if the signs of the oprnds differ

I/O Device	Input/Output Operation	Mnemonic	Timing (average) in Microseconds	Remarks
2922-3	Read Card	XIO	186 + 65F	F = Field length in XIO
	Test Reader Busy	TIOB	112	
	Test Reader Error	TIOB	112	
2922-3	Print	XIO	2400 + 6.6F	F = Field length in XIO
	Test Printer Busy	TIOB	115	
	Test Printer Error	TIOB	115	
	Test Channel 9 or 12	TIOB	115	
	Control Carriage	CIO	120	
1442	Punch Card	XIO	183 + 65F	F = Field length in XIO
	Test Punch Busy	TIOB	112	
	Test Punch Error	TIOB	112	
	Test Feed Error	TIOB	112	
BSCA	Transmit Only	XIO	194 + 54F	F = Field length in XIO
	Transmit and Receive	XIO	194 + 54F	
	Receive Only	XIO	200 + 53F	
	Receive Initial	XIO	198 + 53F	
	Address Prepare	XIO	192 + 53F	
	Enable/Disable BSCA	CIO	106	
	Enable/Disable ITB	CIO	106	
	Store Current Address	CIO	149	
	Store Sense	CIO	115	
	Store ITB Address	CIO	149	
	Test Any Check Condition	TIOB	106	
	Test Busy	TIOB	103	

APPENDIX I. INSTRUCTION USE EXAMPLES

The following examples illustrate the use of instructions. Note that these examples closely approximate machine language to best illustrate the operation of the system. For clarity, the mnemonic for each operation code is used instead of the actual machine code. In addition, whenever possible, the contents of registers, storage locations, and so on, are given in decimal notation rather than the actual binary formats. When binary formats are used, they are segmented into bytes (eight bits) for ease of visual comparison.

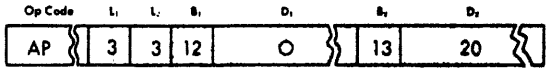
DECIMAL ADD

The signed, packed decimal field at location 500-503 is to be added to the signed, packed decimal field at location 2000-2003.

Assume:

Reg 12	20 00
Reg 13	04 80
Loc 2000-2003 (before)	00 38 46 0-
Loc 500-503	01 12 34 5+

The instruction is:



Loc 2000-2003 (after) 00 73 88 5+
Condition code =2 sum is greater than zero.

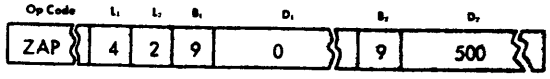
ZERO AND ADD

The signed, packed decimal field at loation 4500-4502 is to be moved to location 4000-4004 with four leading zeros in the result field.

Assume:

```
Reg 9                                          40 00
Loc 4000-4004 (before)              12 34 56 78 90
Loc 4500-4502                          38 46 0-
```

The instruction is:




```
Loc 4000-4004 (after)               00 00 38 46 0-
```
Condition code=1; sum is less than zero.

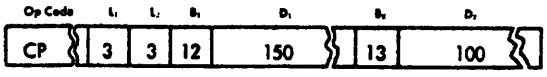
COMPARE DECIMAL

The contents of location 700-703 are to be compared algebraically with the contents of location 500-503.

Assume:

```
Reg 12                                    05 50
Reg 13                                    04 00
Log 700-703                     17 25 35 +6
Loc 500-503                     06 72 14 2+
```

The instruction is:



Condition code=2; first operand is high.

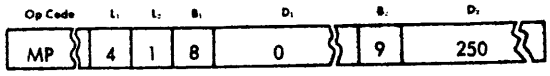
MULTIPLY DECIMAL

The signed, packed decimal field in location 1200-1204 is to be multiplied by the signed, packed decimal field in location 500-501, and the product is to be placed in location 1200-1204.

Assume:

Reg 8		12 00
Reg 9		02 50
Loc 1200-1204 (before)	00 00 38 46	0-
Loc 500-501		32 1-

The instruction is:



Loc 1200-1204 (afteter) 01 23 45 66 0+
Condition code: unchanged.

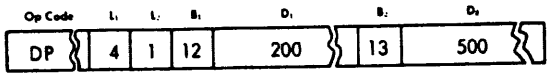
DIVIDE DECIMAL

The signed, packed decimal field at location 2000-2004 is to be divided by the packed decimal field at location 3000-3001:

Assume:

Reg 12		18 00
Reg 13		25 00
Loc 2000-2004 (before)	01 23 45 67	8+
Loc 3000-3001		32 1-

The instruction is:



Loc 2000-2004 (after) 38 46 0-01 8+
where the quotient is 38460--and the remainder is 018+ Condition code: unchanged.

PACK

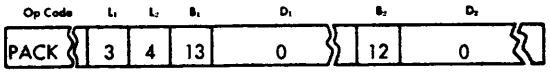
Assume locations 1000-1004 contain the following:

```
    Z1 Z2 Z3 Z4 S5
where Z = four-bit zone code
      S = four-bit sign code
```

The field is to be in packed format with two leading zeros and placed in location 2500-2503.

```
Reg 12                              10 00
Reg 13                              25 00
Loc 1000-1004                 Z1 Z2 Z3 Z4 S5
Loc 2500-2503 (before)        A  B  C  D
```

The instruction is:




```
Loc 2500-2503 (after)         00 12 34 5S
Condition code: unchanged
```

UNPACK

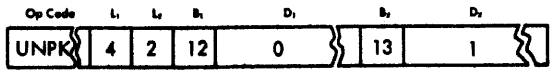
Assume locations 2501-2503 contain the following fields:

12 34 5S

This field is to be put into zoned format and placed in the locations 1000-1004 where: S is a four-bit sign code.

```
Reg 12                              10 00
Reg 13                              25 00
Loc 2501-2503                 12 34 5S
Loc 1000-1004 (before)        A  B  C  D  E
```

The instruction is:



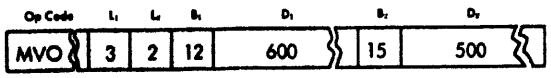
and results in
Loc 1000-1004 (after) Z1 Z2 Z3 Z4 S5
where Z is a four-bit zone code.
Condition code: unchanged.

MOVE WITH OFFSET

The unsigned three-byte field at location 4500-4502 is to be moved to location 5600-5603 and given the sign of the one byte field location at 5603.

```
Reg 12                              50 00
Reg 15                              40 00
Loc 5600-5603 (before)        77 88 99 0+
Loc 4500-4502                 12 34 56
```

The instruction is:



Loc 5600-5603 (after) 01 23 45 6+
Condition code: unchanged.

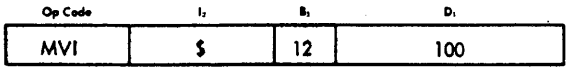
MOVE IMMEDIATE

A dollar sign ($) is to be placed in location 2100, leaving locations 2101-2105 unchanged. Let Z represent a four-bit zone.

Assume:

Reg 12		20 00
Loc 2100-2105 (before)	Z0 Z1 Z2 Z3 Z5 Z0	

The instruction is:



Loc 2100-2105 (after) Z1 Z2 Z3 Z5 Z0
Condition code: unchanged.

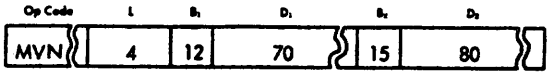
MOVE NUMERIC

Let Z and Y represent four-bit zones. The numberic parts of the eight-bit characters in the field at locations 6070-6074 are to be replacei by the numeric parts of eight-bit characters at locations 8080-8084.

Assume:

Reg 12		60 00
Reg 15		80 00
Loc 6070-6074 (before	Y1 Y2 Y3 Y4 Y5	
Loc 8080-8084	Z3 Z6 Z9 Z7 Z8	

The instruciton is:



Loc 6070-6074 (after) Y3 Y6 Y9 Y7 Y8
Condition code: unchanged.

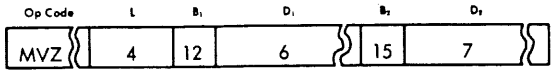
MOVE ZONES

Let Z and Y represent four-bit zones in the eight-bit characters making up the fields at locations 2006-2010 and 3007 3011, respectively. The zones of he field at 2006-2010 are to be replaced by the zones from location 3007-3011.

Assume:

Reg 12			20	00	
Reg 15			30	00	
Loc 2006-2010 (before)	Z1	Z4	Z7	Z8	Z5
Loc 3007-3011	Y8	Y7	Y4	Y6	Y8

The instruction is:



Loc 2006-2010 (after)	Y1	Y4	Y7	Y8	Y5

Condition code: unchanged

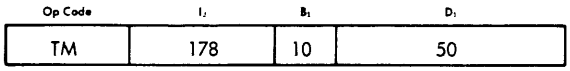
TEST UNDER MASK

Test bit positions 0, 2, 3, and 6 of a given byte in storage to determine if all of these bit positions contain 1s. A Test Under Mask instruciton with a mask of 10110010=17810 is used. The byte to be tested is stored at location 1250 and contains 01101101.

Assume:

Reg 10 12 00

The instruction is:



Mask from TM 10110010
Byte tested 01101101
Selected result 00100000
Condition code = 1; some selected bits are 0, some selected bits are 1 (mixed).

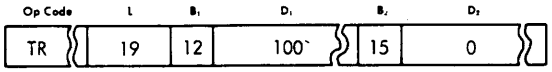
TRANSLATE

Assume a stream of 20 characters comes into location 2100 in EBCDIC. Translate to ASCII-8.

Assume:

Reg 12 20 00
Reg 15 10 00
Loc 2100-2119 (before) JOHN JONES 257 W.95

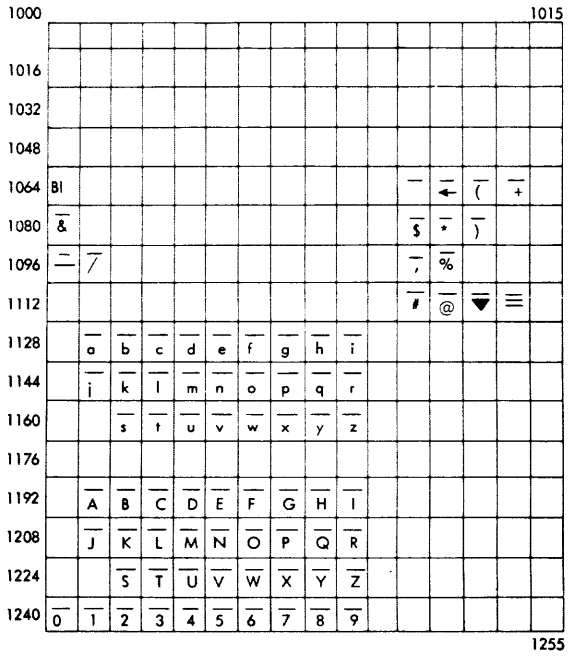
The instruction is:



Loc 2100-2119 (after) JOHN JONES 257 W.95

where the overbar means the same graphic in ASCII-8 Condition code: unchanged.

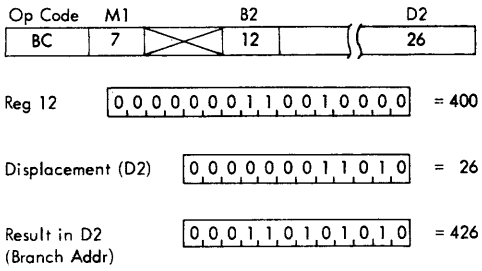
Translate Table



BRANCH-ON CONDITION

Assume a prior operation has been performed which resulted in setting the condition code in the PSW. The program is to branch when the condition code has any value above zero.

To cover all possible condition code settings (only one of four possible settings can be present at a time), the mask is set to 7 (binary 0111). This forces the program to branch when the condition code is either 01, 10, or 11 (binary). If the value in the B2 field of the instruciton is 12 (for example), an effective branch address is generated by adding the contents of the D2 field to the contents of a general register 12. Therefore, if register 12 contains a bit configuration that represents the decimal value of 400 and the D2 field contains, likewise, a value of 26, the resulting branch address is 426. The condition code is not changed.


Op Code
M1
B2
D2
BC
7
12
26
Reg 12
0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0
= 400
Displacement (D2)
0 0 0 0 0 0 0 1 1 0 1 0
= 26
Result in D2
(Branch Addr)
0 0 0 1 1 0 1 0 1 0 1 0
= 426

APPENDIX J. 4-of-8/EBCDIC CONVERSION TABLE

Graphic	4 of 8								EBCDIC							
	N	X	O	R	8	4	2	1	0	1	2	3	4	5	6	7
'	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0	1
7	0	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1
#	0	0	0	1	1	0	1	1	0	1	1	1	1	0	1	1
"	0	0	0	1	1	1	0	1	0	1	1	1	1	1	1	1
=	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0
X	0	0	1	0	0	1	1	1	1	1	1	0	0	1	1	1
,	0	0	1	0	1	0	1	1	0	1	1	0	1	0	1	1
?	0	0	1	0	1	1	0	1	0	1	1	0	1	1	1	1
:	0	0	1	0	1	1	1	0	0	1	1	1	1	0	1	0
+	0	0	1	1	0	1	1	0	0	1	0	0	1	1	1	0
>	0	0	1	1	1	0	1	0	0	1	1	0	1	1	1	0
;	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0
p	0	1	0	0	0	1	1	1	1	1	0	1	0	1	1	1
$	0	1	0	0	1	0	1	1	0	1	0	1	1	0	1	1
¬	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	1
-	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0
(	0	1	0	1	0	1	1	0	0	1	0	0	1	1	0	1
_	0	1	0	1	1	0	1	0	0	1	1	0	1	1	0	1
)	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	1
C	0	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1
E	0	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1
F	0	1	1	0	0	1	1	0	1	1	0	0	0	1	1	0
I	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0	1
¢	0	1	1	0	1	0	1	0	0	1	0	0	1	0	1	0
<	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0	0
A	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1
B	0	1	1	1	0	0	1	0	1	1	0	0	0	0	1	0
D	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0	0
H	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	0
G	1	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1
.	1	0	0	0	1	0	1	1	0	1	0	0	1	0	1	1
|	1	0	0	0	1	1	0	1	0	1	0	0	1	1	1	1

Graphic	4 of 8								EBCDIC							
	N	X	O	R	8	4	2	1	0	1	2	3	4	5	6	7
&	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0
3	1	0	0	1	0	0	1	1	1	1	1	1	0	0	1	1
5	1	0	0	1	0	1	0	1	1	1	1	1	0	1	0	1
6	1	0	0	1	0	1	1	0	1	1	1	1	0	1	1	0
9	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0
(')	1	0	0	1	1	1	0	0	0	1	1	1	1	1	0	0
T	1	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1
V	1	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1
W	1	0	1	0	0	1	1	0	1	1	1	0	0	1	1	0
Z	1	0	1	0	1	0	0	1	1	1	1	0	1	0	0	1
none	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	0
%	1	0	1	0	1	1	0	0	0	1	1	0	1	1	0	0
/	1	0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
S	1	0	1	1	0	0	1	0	1	1	1	0	0	0	1	0
U	1	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0
Y	1	0	1	1	1	0	0	0	1	1	1	0	1	0	0	0
L	1	1	0	0	0	0	1	1	1	1	0	1	0	0	1	1
N	1	1	0	0	0	1	0	1	1	1	0	1	0	1	0	1
O	1	1	0	0	0	1	1	0	1	1	0	1	0	1	1	0
R	1	1	0	0	1	0	0	1	1	1	0	1	1	0	0	1
!	1	1	0	0	1	0	1	0	0	1	0	1	1	0	1	0
*	1	1	0	0	1	1	0	0	0	1	0	1	1	1	0	0
J	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	1
K	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0
M	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0	0
Q	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1
2	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	0
4	1	1	1	0	0	1	0	0	1	1	1	1	0	1	0	0
8	1	1	1	0	1	0	0	0	1	1	1	1	1	0	0	0
blank	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0

Graphic	EBCDIC	4 of 8
	0123 4567	NXOR 8421
blank	0100 0000	1111 0000
¢	0100 1010	0110 1010
.	0100 1011	1000 1011
<	0100 1100	0110 1100
(	0100 1101	0101 0110
+	0100 1110	0011 0110
|	0100 1111	1000 1101
&	0101 0000	1000 1110
!	0101 1010	1100 1010
$	0101 1011	0100 1011
*	0101 1100	1100 1100
)	0101 1101	0101 1100
;	0101 1110	0011 1100
¬	0101 1111	0100 1101
-	0110 0000	0100 1110
/	0110 0001	1011 0001
,	0110 1011	0010 1011
%	0110 1100	1010 1100
_	0110 1101	0101 1010
>	0110 1110	0011 1010
?	0110 1111	0010 1101
:	0111 1010	0010 1110
#	0111 1011	0001 1011
@	0111 1100	1001 1100
'	0111 1101	0000 1111
=	0111 1110	0001 1110
"	0111 1111	0001 1101
A	1100 0001	0111 0001
B	1100 0010	0111 0010
C	1100 0011	0110 0011
D	1100 0100	0111 0100
E	1100 0101	0110 0101

Graphic	EBCDIC	4 of 8
	0123 4567	NXOR 8421
F	1100 0110	0110 0110
G	1100 0111	1000 0111
H	1100 1000	0111 1000
I	1100 1001	0110 1001
J	1101 0001	1101 0001
K	1101 0010	1101 0010
L	1101 0011	1100 0011
M	1101 0100	1101 0100
N	1101 0101	1100 0101
O	1101 0110	1100 0110
P	1101 0111	0100 0111
Q	1101 1000	1101 1000
R	1101 1001	1100 1001
none	1110 0000	1010 1010
S	1110 0010	1011 0010
T	1110 0011	1010 0011
U	1110 0100	1011 0100
V	1110 0101	1010 0101
W	1110 0110	1010 0110
X	1110 0111	0010 0111
Y	1110 1000	1011 1000
Z	1110 1001	1010 1001
0	1111 0000	1001 1010
1	1111 0001	1110 0001
2	1111 0010	1110 0010
3	1111 0011	1001 0011
4	1111 0100	1110 0100
5	1111 0101	1001 0101
6	1111 0110	1001 0110
7	1111 0111	0001 0111
8	1111 1000	1110 1000
9	1111 1001	1001 1001

APPENDIX K. EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE (EBCDIC) CARD CODE TABLE

Bit Positions of a Byte 0, 1, 2, 3, 4, 5, 6, 7

4567	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
0000	TO9 18	TE9 18	EO9 18	TEO9 18	Blank	T &	E -	TEO	TO 18	TE 18	EO 18	TEO 18	TO	EO	O 28	O 0	0
0001	T9 1	E9 1	O9 1	9 1	TO9 1	TE9 1	01 /	TEO9 1	TO 1	TE 1	EO 1	TEO 1	T1 A	E1 J	EO9 1	1 1	1
0010	T9 2	E9 2	O9 2	9 2	TO9 2	TE9 2	EO9 2	TEO9 2	TO 2	TE 2	EO 2	TEO 2	T2 B	E2 K	02 S	2 2	2
0011	T9 3	E9 3	O9 3	9 3	TO9 3	TE9 3	EO9 3	TEO9 3	TO 3	TE 3	EO 3	TEO 3	T3 C	E3 L	03 T	3 3	3
0100	T9 4	E9 4	O9 4	9 4	TO9 4	TE9 4	EO9 4	TEO9 4	TO 4	TE 4	EO 4	TEO 4	T4 D	E4 M	04 U	4 4	4
0101	T9 5	E9 5	O9 5	9 5	TO9 5	TE9 5	EO9 5	TEO9 5	TO 5	TE 5	EO 5	TEO 5	T5 E	E5 N	05 V	5 5	5
0110	T9 6	E9 6	O9 6	9 6	TO9 6	TE9 6	EO9 6	TEO9 6	TO 6	TE 6	EO 6	TEO 6	T6 F	E6 O	06 W	6 6	6
0111	T9 7	E9 7	O9 7	9 7	TO9 7	TE9 7	EO9 7	TEO9 7	TO 7	TE 7	EO 7	TEO 7	T7 G	E7 P	07 X	7 7	7
1000	T9 8	E9 8	O9 8	9 8	TO9 8	TE9 8	EO9 8	TEO9 8	TO 8	TE 8	EO 8	TEO 8	T8 H	E8 Q	08 Y	8 8	8
1001	T9 18	E9 18	O9 18	9 18	T 18	E 18	O 18	18	TO 9	TE 9	EO 9	TEO 9	T9 I	E9 R	09 Z	9 9	9
1010	T9 28	E9 28	O9 28	9 28	T28 ¢	E28 !	TE :	28 :	TO 28	TE 28	EO 28	TEO 28	TO9 28	TE9 28	EO9 28	TEO9 28	A
1011	T9 38	E9 38	O9 38	9 38	T38 .	E38 $	O38 ,	38 #	TO 38	TE 38	EO 38	TEO 38	TO9 38	TE9 38	EO9 38	TEO9 38	B
1100	T9 48	E9 48	O9 48	9 48	T48 <	E48 *	O48 %	48 @	TO 48	TE 48	EO 48	TEO 48	TO9 48	TE9 48	EO9 48	TEO9 48	C
1101	T9 58	E9 58	O9 58	9 58	T58 (	E58)	O58 _	58 '	TO 58	TE 58	EO 58	TEO 58	TO9 58	TE9 58	EO9 58	TEO9 58	D
1110	T9 68	E9 68	O9 68	9 68	T68 +	E68 ;	O68 >	68 =	TO 68	TE 68	EO 68	TEO 68	TO9 68	TE9 68	EO9 68	TEO9 68	E
1111	T9 78	E9 78	O9 78	9 78	T78 |	E78 ¬	O78 ?	78 "	TO 78	TE 78	EO 78	TEO 78	TO9 78	TE9 78	EO9 78	TEO9 78	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Hexadecimal Representation for Bits 0 1 2 3 (Data Switch 1)

Hexadecimal Representation for Bits 4567 (Data Switch 2)

Notes:
The numbers contained in the blocks represent row punches in the card.
"T" indicates a twelve zone punch.
"E" indicates an eleven zone punch.
EBCDIC graphic characters are shown in the shaded areas.

APPENDIX L. BSCA CODE TABLES

ASCII TABLE OF CONTROL CHARACTERS

Row	Column 0	1	2	3	4	5	6	7
Bits → 765								
4321 ↓	000	001	010	011	100	101	110	111
0	0000 NUL	DLE	SP	ACK-0 Trailer				
1	0001 SOH	DC1		ACK-1 Trailer				
2	0010 STX	DC2						
3	0011 ETX	DC3						
4	0100 EOT	DC4						
5	0101 ENQ	NAK						
6	0110 ACK	SYN						
7	0111 BEL	ETB						
8	1000 BS	CAN						
9	1001 HT	EM						
10	1010 LF	SUB						
11	1011 VT	ESC		WACK Trailer				
12	1100 FF	FS		RVI Trailer				
13	1101 CR	GS						
14	1110 SO	RS						
15	1111 SI	US						DEL

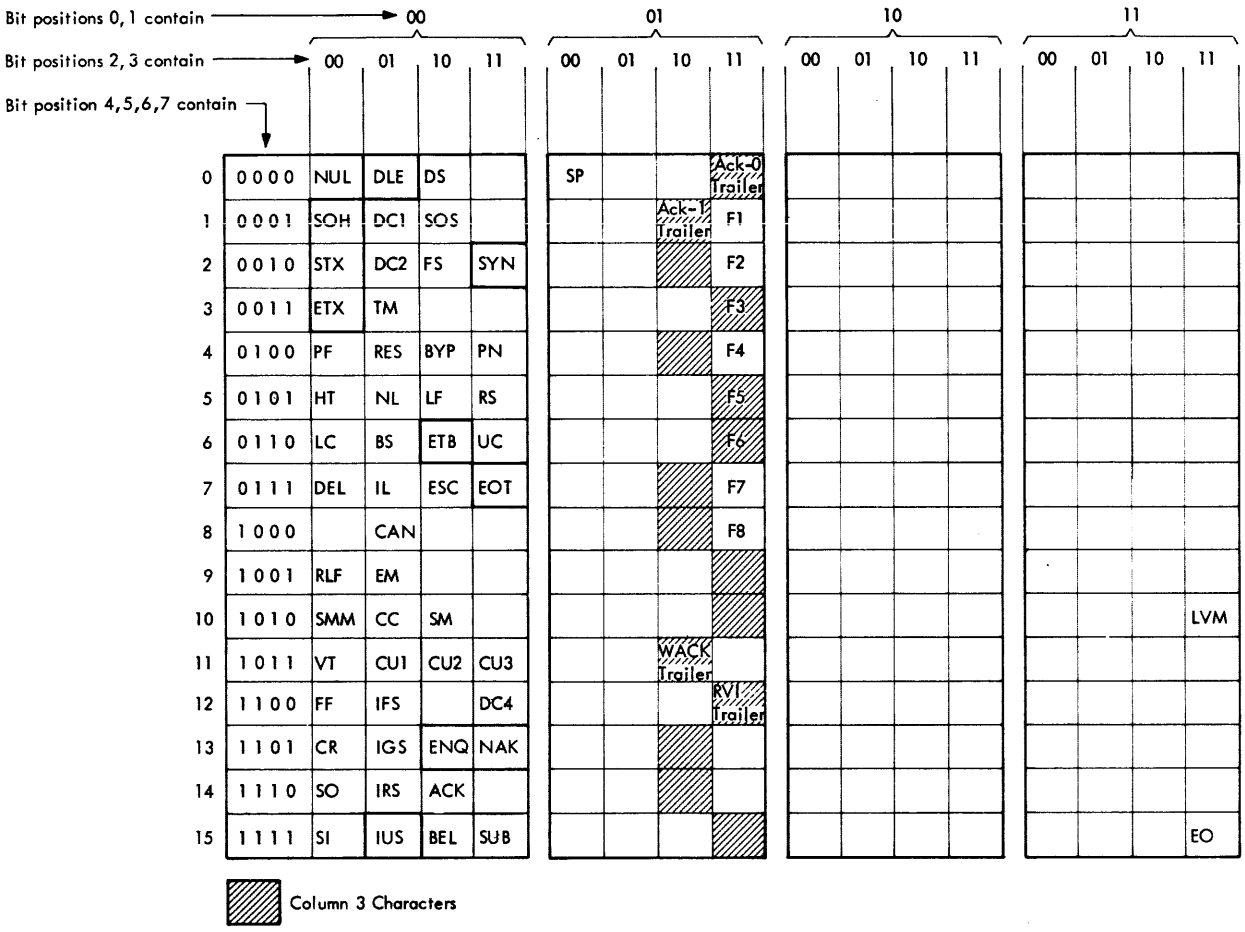
Column 3 Characters

NOTE: The trailer characters have a normal data function when not preceded by DLE

ASCII TABLE OF GRAPHICS

Row	Column 0	1	2	3	4	5	6	7
Bits → 765								
4321 ↓	000	001	010	011	100	101	110	111
0	0000			0	@	P	`	p
1	0001		!	1	A	Q	a	q
2	0010		"	2	B	R	b	r
3	0011		#	3	C	S	c	s
4	0100		$	4	D	T	d	t
5	0101		%	5	E	U	e	u
6	0110		&	6	F	V	f	v
7	0111		'	7	G	W	g	w
8	1000		(	8	H	X	h	x
9	1001		)	9	I	Y	i	y
10	1010		*	:	J	Z	j	z
11	1011		+	;	K	[	k	{
12	1100		,	<	L	\	l	¦
13	1101		-	=	M	]	m	}
14	1110		.	>	N	^	n	~
15	1111		/	?	O	_	o	

EBCDIC TABLE OF CONTROL CHARACTERS



Notes: The trailer characters have a normal data function when not preceded by DLE

EBCDIC TABLE OF GRAPHICS

Bits positions 0,1 contain →		00				01				10				11			
Bit positions 2,3 contain →		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
Bit positions 4,5,6,7 contain ↓																	
0	0000						&	-						{	}	\	0
1	0001							/		a	j	~		A	J		1
2	0010									b	k	s		B	K	S	2
3	0011									c	l	t		C	L	T	3
4	0100									d	m	u		D	M	U	4
5	0101									e	n	v		E	N	V	5
6	0110									f	o	w		F	O	W	6
7	0111									g	p	x		G	P	X	7
8	1000									h	q	y		H	Q	Y	8
9	1001									i	r	z		I	R	Z	9
10	1010					¢	!	¦	:								
11	1011					•	$	,	#								
12	1100					<	*	%	@					⌡		⌐	
13	1101					(	)	_	'								
14	1110					+	;	>	=					⌠			
15	1111					¦	¬	?	"								

Note: The graphics shown are given for reference only; they may vary depending on the printer equipment attached.

BSCA SENSE BYTE TABLE

Bit Position		Valid on XIO Insn during	
		TSM	REC
0	TSM/REC Mode Error		X
1	EOT Received		X
2	CRC/LRC/VRC Check		X
3	Timeout		X
4	Short Record	X	X
5	Storage Wraparound	X	X
6	Parity Check (VRC or BSCA Check)	X	X
7	Overrun	X	X

APPENDIX M. STOP CONDITIONS

TABLE OF NORMAL STOPS

Cause of Stop	Displayed on Console							
	P	I	U	L	E	S	T	R
Stop Key, Address Stop, Instruction Step	0	0	NSI Op Code		NSI Address			
System Reset	0	0	0	0	0	0	0	0
Programmed Halt	0	0	9	9	Halt Identifier			
Register Alter	Number of Selected Register		0	0	Register Data			
Register Display	Number of Selected Register		0	0	Register Data			

TABLE OF PROGRAMMING ERROR STOPS

Cause of Stop	Displayed on Console							
	P	I	U	L	E	S	T	R
Invalid Op Code	0	1	Op Code of Current Instruction		Address of Current Instruction			
Address Error: 1. Address lower than 144. 2. Address exceeds storage. 3. Invalid register number	0 0 0	4 5 5						
Specification Error: 1. Instruction not on halfword boundary. 2. Binary operand not on halfword boundary 3. In AP, SP, CP, and ZAP instructions, L2 is greater than L1. 4. In MP and DP instructions, L2 is greater than 7 or greater than or equal to L1. 5. Bits 12 through 15 of an RX format instruction are not all zero. 6. Field length in I/O instruction is zero or (except for BSCA and SIOC-connected devices) exceeds allowable maximum.	0	6						
Binary Overflow	0	8						
Data Error: 1. A sign or digit code of an operand in ZAP, AP, SP, CP, MP, or DP instruction is incorrect, or operand fields overlap incorrectly. 2. The multiplicand field (first operand) in an MP instruction has insufficient high-order zeros. 3. Invalid digit code in second operand of ED instruction	0	7						
Decimal Divide Check	0	B						

APPENDIX N. GLOSSARY

Acknowledgment: In teleprocessisng, a message from the addressee sent in response to a communication received.

Adapter: A device used to interconnect two machines of different characteristics.

Address: 1. A number which designates a register, a storage location or an input/output device.

2. The part of an instruction which specifies the location of an operand.

ASCII: American National Standard Code for Information Interchange. A code for data transfer adopted by the American Standards Association to achieve compatibility between data devices. ASCII is a seven-bit code, which is also available in an extended eight-bit version then termed ASCII-8.

Available: A condition in which part of a data processing system is operational, does not contain data or error check conditions, and is not busy with a previously initiated operation.

Bid: A procedure which establishes the right to start the first transmission to one or the other teleprocessing terminal of a point-to-point data line.

Binary: A number system whose successive digits are interpreted as coefficients of the successive powers of the base 2.

Binary Coded Decimal: A decimal notation in which the individual decimal digits are each represented by four binary digits having a position value of 8-4-2-1. For example, in binary coded decimal

(8 421 8 421)
notation, the number 23 is represented as 0 010 0 111, whereas
(16 8 421)
in binary notation 23 is reprsented as 1 0 111

Binary Synchronous Communication System: A teleprocessing system in which data is transmitted as a serial stream of binary digits (0-bits and 1-bits) between stations on a communications channel. The stations are locked in step through the recognition of a specified bit pattern, known as the 'sync pattern'.

Bit: The contraction of 'binary digit', the smallest unit of information in the programmable terminal. A bit can have either of the two binary values: zero or one.

Bit stream: A binary signal without regard to grouping by character.

Block (Blocking): 1. To group records for the purpose of conserving storage space or increasing the efficiency of access or processing.

2. A physical record so constituted, or a portion of a telecommunications message defined to be a unit of data transmission.

Branching: A deviation from the strict sequence of a program.

Buffer: A storage device needed to compensate for a difference in rate of data flow, when transmitting data from one device to another.

Busy: A condition in which part of a data processing system is not available for use.

Byte: The basic unit of information in the Programmable Terminal. Every byte consists of eight bits, each having a value of 0 or 1.

Character: One of a set of elementary signals which may include decimal digits 0 through 9, the letters A through Z, punctuation marks, and any other symbols acceptable to a computer for reading, writing, or storing.

Clear Band: On a document, an area 0.5 in. (12,7 mm) high, reserved for optical reading. It is parallel to the reference (lower) edge and centered on the ideal print centerline, and extends from the left-hand edge to the right-hand edge. The clear band must be free from dirt and foreign matter.

Code Line: A group of characters for optical reading printed along one line on a document.

Column Binary: A coding system that allows the capacity of an IBM punched card to be increased from 80 to 160 storage positions by using the six-bit BCD code. Every punch position represents a bit (-1 if hole, =0 if no hole). Sets of bits are read by card column, two 6-bit sets per column. The first set consists of the 12-11-0-1-2-3 positions; the second set consists of the 4-5-6-7-8-9 positions.

Command Chaining: The uninterrupted execution of a sequence of commands applying to the same device. The commands must be stored adjacent to each other in ascending order of address. (Chaining occurs only in the input/output channel.)

Command Code: This specifies for the input/output channel.

Common Carrier: A company which furnishes communications services to the general public, and which is regulated by the appropriate local or central authorities.

Communication: The transmission of messages between points of origin and reception without alteration of the sequence or structure of the information contents.

Condition Code: A code, located in bits 2 and 3 of the PSW, to reflect certain conditions in the central processing unit. The two binary bits permit four possible combinations (00, 01, 10, or 11) which indicate the result of successful operations or report on the conditions leading to unsuccessful operations. The condition code can be tested by the program to allow branching decisions.

Configuration: The specific combination of data processing units forming a system.

Contention: A condition on a point-to-point line that exists when the right to the first transmission is not established, and thus oth stations may attempt to transmit at the same time.

Control Character: A character used in BSCA transmission to initiate, modify, or stop operations.

Control Instruction: This direct an I/O device to perform a specified function, e.g., to select a stacker pocket or to initiate a carriage skip.

Control Unit: A device specifically designed to communicate with the input/output channel, and to start and supervise operations in the individual I/O devices connected.

Control Unit End: A signal sent by a control unit when the control-unit-busy condition ends.

Cylinder: A set of ten vertically arranged disk tracks.

Data Lnk: The communication lines, modems, and communication controls of all stations connected to the line, used in the transmission of information between two or more stations.

Data Set: A device which converts digital dc signals into frequency-encoded signals to allow transmission over telephone lines. The device also reconverts frequency-encoded signals received from a remote station into digital dc signals and performs certain control functions required to establish the data link. (Also called 'modem'.)

Decimal Number: A number in which individual decimal digits ranging from 0 to 9 are each represented by four binary bits. Decimal numbers

have a variable field length from 1 to 16 bytes and may be in zoned format or packed format. In zoned format, the sign (+ or -) is in the left half of the low-order byte; in packed format, the sign is in the right half of the low-order byte.

Demodulation: The process of retrieving intelligence (data) from a modulated carrier wave; the reverse of modulation.

Device End: A signal sent by a channel-connected device when the device-busy condition ends.

Diagnostic Routine: A routine designed to locate a malfunction, either in other routines or in the computer.

Direct Access: The retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data.

Direct Addressing: A method of addressing main storage by which the 14 low-order bits of the combined base and displacement field are used to refer directly to byte locations in main storage.

Displacement: A value contained in the D1 or D2 field of an instruction. This value may be added to the so-called base address (contained in the register specified by the B1, B2, or R2 field of an instruction) to form an effective address.

Duplex Transmission: The simultaneous two-way independent transmission in both directions. (Also called 'full-duplex' transmission.)

EBCDIC: Extended binary-coded-decimal interchange code. This code is based on eight binary bits (byte), allowing any one of 256 characters to be represented.

Effective Addressing: A method of addressing main storage by which the contents of the general register, specified by the B-field of an instruction, are added to the contents of the D-field of the instruction. (Also known as 'indexing'.)

Explicitly Addressed Operand: An operand specified by means of a main-storage address, and not located in the instruction itself (immediate operand) or in a general register.

Field: A continuous set of digits or characters forming a meaningful entity.

Fixed-Length Record: A record having a specific length that cannot be altered.

Format: The general makeup of data, instructions, records, or files.

General Register: An auxiliary storage unit in the control unit, used for temporary storage of data. The 2922-1 has eight general registers, each accommodating two bytes (one halfword) of data.

Graphic: The visual representation of a character or symbol.

Halfbyte: The leftmost or rightmost four bits of an eight-bit byte. It can contain the representation of a digit, a zone, or the sign of a number.

Half-duplex Transmission: In communications facilities, this is an alternate, one-way-at-a-time transmission.

Halfword: Two adjacent bytes where the left byte is on a halfword boundary.

Halfword Binary Number: A whole number in binary notation with a fixed length of two bytes, having its sign (+ or -) in the leftmost bit.

Hexadecimal: A number system whose successive digits are interpreted as coefficients of the successive power of the base 16. The values 0 through 15 are written as digits 0 through 9 (for values 0-9) and alphabetic characters A through F (for values 10-15); thus a hexadecimal position may have any value from zero to F (see Appendix D).

High-Order Bit: The bit position of a given storage area representing the highest numeric value in binary notation. (Also referred to as the 'leftmost bit'.)

Immediate Operand: A byte of data located in the instruction.

Indexing: See "Effective Addressing".

Input: The data that is fed into a terminal. The input may be data that is to be processed or may be variable parameters or instructions that are supplied to implement certain operations.

Input/Output Operations: The transference of information to main storage from sources external to the control unit and/or from main storage to external destinations.

Integral Boundary: An address boundary which is a multiple of the length of the unit of information stored. A halfword (two bytes) binary number, for example, is addressed at an integral boundary that is a multiple of the number 2.

Interface: The line of demarcation between two pieces of equipment having different functions.

Interruption (Interrupt): An automatic branching in the stored program which alerts the system to the end of a data transfer. The object

of the interrupt system is to indicate the availability of data received from I/O devices at the earliest moment possible, without forcing the program to continually check on the I/O device to find its end condition.

Leased Line: A communications facility reserved for the sole use of a single leading customer.

Leftmost Bit: See "High-Order Bit".

Limited Conversational Mode: In teleprocessing, a special type of operation permitting a slave station that has received a complete message to reply with heading or test matter instead of with a standard acknowledgement. Thus, during transmission of a conversational reply, the roles of the two interacting stations are reversed, and the original transmitter becomes the receiver.

Logical Data: This consists of alphabetic or numeric character codes and may be in a one-byte fixed length or in variable length ranging from oen to 256 bytes. There is no sign.

Logical Functions: Non-arithmetic data processing steps in the control unit such as moving and comparing operands, editing, and testing.

Logical Record: A record from the standpoint of its contents, function, and use, rather than its physical attributes; i.e., a record that is defined in terms of the information it contains.

Low-Order Bit: The bit position of a given storage area representing the lower numeric value in binary notation. (Also referred to as the 'rightmost bit'.)

Main Storage: All addressable storage, from which instructions or data can be read out and executed or into which instructions or data can be loaded. General-purpose registers are outisde of main storage.

Master Station: In teleprocessing, the station which has the right to transmit a message or a request for a response to a message, at a given instant. The master station can be a control, tributary, primary, or secondary station.

Mnemonic: In programming terminology, a contraction or abbreviation whose characters are suggestive of the full expression and are therefore easy to remember.

Mode Set Command: A command issued to set recording density, parity mode, data converter and translator on or off in magnetic tape units. A particular setting is retained until a new mode set command is issued for the control unit concerned.

Modem: See "Data Set".

Modulation: The process by which some characteristic of one wave or signal is varied in accordance with another wave or signal. This technique is used in data sets (modems) to make data-processing machine signals suitable for transmission over telephone lines.

Multipoint Data Exchange System: A teleprocessing system in which one station always has control and the other stations act as tributaries.

Next Sequential Instruction: The instruction due to be read out immediately following the current instruction, and addressed by the rightmost two bytes of the PSW.

Non-Overlap Mode: A mode of operation in which an input/output operation excludes other input/output operations and internal control unit processing.

Not Operational: A condition in which part of a data processing system is in a not-ready status, or has an error, or a data check condition, or power off.

Offline: Pertaining to equipment or devices not under direct control of the central control unit.

Online: Pertaining to equipment or devices under direct control of the central control unit.

Operand: An item of information which is to be processed by an instruction. Instructions will therefore always contain addresses that refer to the location of one or two operands so that these may be fetched from their location and processed. An operand can also be located in the instruction.

Operation Code: The leftmost byte of an instruction which specifies the length of the instruction, the data format, and the operation to be performed.

Output: The resultant data produced by a control unit program.

Overlap Mode: An extension of the 2922-1 time-sharing principle that allows a maximum number of I/O devices to participate in overlapping.

Pack: To convert decimal data from zoned to packed format.

Packed Decimal: A data format in which two numeric digits, or one digit and a sign, are stored in one eight-bit byte.

Parity Bit: A binary digit appended to (or omitted from) an array of bits to make the sum of all 1-bits in the array always odd (or always even).

Pending Interrupt: As interrupt that has been requested but cannot be performed because the channel mask bit in the current PSW is zero.

Physical Record: A record from the standpoint of the manner or form in which it is stored, retrieved and moved; i.e. a record that is defined in terms of physical qualities.

Point-to-Point Data Exchange System: A teleprocessing system connecting only two stations for any one transmission.

Primary Station: In teleporcessing, the station on a point-to-point line, where contention exists, which will gain control of the line and have priority to transmit messages should a contest for the right to transmit arise.

Program Status Word: A four-byte area of information in auxiliary storage used by the internal control to effect instruction sequencing and to hold and indicate the status of the terminal in relation to the program being executed.

Protected Area: The main-storage locations 0 through 143 which are reserved for internaal control and are not available to the program. Addresses which refer to this area are rejected as invalid.

Read Operation: This pertains to data transfer from an I/O device to the control unit.

Record: A general term for any unit of data that is distinct from all others when considrred in a particular context.

Rightmost Bit: See "Low-Order Bit".

RR Format: This denotes a register-to-register operation, i.e., both operands are located in general registers.

RX Format. This denotes a register-to-storage or storage-to-register operation, i.e., one operand is located in core stoage and the other is located in a general register.

Secondary Station: In teleprocessing, the station on a point-to-point line that will relinquish its bid for the line in a contentio situation.

Sense Operation: An operation similar to a read operation, except that the information read into main storage originates from status indicators and not from data files.

SI Format: This denotes a storage-immediate operation, i.e., the first operand is located in core storage, and the second operand is located in the instruction.

Slave Station: In teleprocessing, the staiton which is receiving a message from a master station or which is obliged to transmit an appropriate reply to the message. The slave station can be a control, tributary, primary, or secondary station.

SS Format: This denotes a storage-to-storage operation, i.e., both operands are located in main storage.

Station: In teleprocessing, the equipment and communication controls attached to any one of the several ends of a communication channel.

Switched Network: A term that denotes common carrier (public) telephone lines.

Synchronization Character: A character that is automatically inserted into the data stream of a communications facility to establish and maintain synchronization.

Synchronize: To lock one element of a system into step with another. The term usually refers to locking a receiver to a transmitter, to allow the receiver to recognize the beginning and end of a data unit within the bit stream.

Synchronous: Having a constant time interval between bits or characters. The term implies that all equipment in the system is in step.

Telecommunications: A general term expressing information exchange over long distances.

Teleprocessing: A form of information handling which is characterized by the fact that data generated at one location is transmitted to another location via telephone lines for remote processing, with subsequent retransmission of the results.

Tete-a-Tete Mode: See "Limited Conversational Mode."

Text Mode: A mode reserved for the information portion of a message transmitted by a communications facility.

Time Sharing: A means of overlapping input/output operations with each other and with processing.

Timeout: In teleprocessing, the time interval allotted between operations before system operation is interrupted and must be restarted.

Turnaround Time: In modems or communication adapters, the actual time required to reverse the direction of transmission from send to receive, or vice versa, when operating in half-duplex mode.

Two's Complement: This pertains to the format of a halfword binary number which represents a negative value. The two's complement of a number is obtained by inverting each bit of the number and adding the value one to the low-order position.

Unblock: To change the format of a file so that a physical record comprises only one logical record.

Unpack: To convert decimal data from packed to zoned format.

Udate: To modify a unit of information or a data file so that it reflects the most recent events.

Variable-Length Records: Logical records in which the number of bytes in each record is not fixed but may vary within prescribed limits.

Working: A condition in which part of a data processing system is executing a previously initiated operation.

Write Operation: This pertains to data transfer from the control unit to an I/O device.

Zone: The four leftmost bits of an unpacked (zoned) decimal number as represented in the EBCDIC or ASCII code. In EBCDIC, the zone has the value 15 and, in ASCII, it has the value 5. Zone bits do not affect the value of the numeric digit contained in the four rightmost bits of the byte.

Zoned Decimal: A data format in which the four rightmost bits of each 8-bit byte represent a decimal digit and the four leftmost bits represent a zone or sign.

A

B

C

D

E

J

K

L

M

N

O

T

U

V

W

X

Z

2922 Programmable Terminal Component Description Printed in U.S.A. GL24-3597-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)