File Number 1440-01
Form A24-3116-3

IBM Systems Reference Library

System Operation Reference Manual IBM 1440 Data Processing System

This reference publication contains the complete instruction set for the IBM 1440, including required and available input/output units and special features. The operation code for each instruction is given in both actual machine language and autocoder-mnemonic form. Formulas are given for calculating instruction-execution time when the time is not a constant. Programming examples are also illustrated.

Refer to the *IBM 1440 Bibliography*, Form A24-3005, for related IBM 1440 publications.

Fourth Edition (October 1966)

This publication is a reprint of form A24-3116-2, incorporating changes released in Technical Newsletter N24-0362.
The original publication and applicable Newsletters are not obsoleted.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
This manual has been prepared by the IBM Systems Development Division, Product Publications, Dept. 171, P. O. Box 6, Endicott, New York 13760

Preface

This publication is a reference text for the IBM 1440 Data Processing System. It provides a detailed explanation of all the instructions used by the system to manipulate data and to control the available input/output devices.

The reader should be familiar with the IBM *1440 System Summary,* Form A24-3006.

This manual is divided into the following independent sections:

Section A.	General Information
Section B.	System Operations
Section C.	IBM 1447 Operations
Section D.	Readers, Punches, and Printers
Section E.	Tape Input/Output Operations
Section F.	Disk Input/Output Operations
Section G.	Miscellaneous Input/Output Operations
Section H.	Special Features
Section I.	Appendix
Section J.	Index of Instructions
Section K.	Index of Branch Instructions and d-Modifiers
Section L.	Index

This manual can be placed in a 3-ring loose-leaf binder with other publications for this system. Remove the staples to avoid damage to the corners, and to facilitate page replacement as new information is made available.

Contents

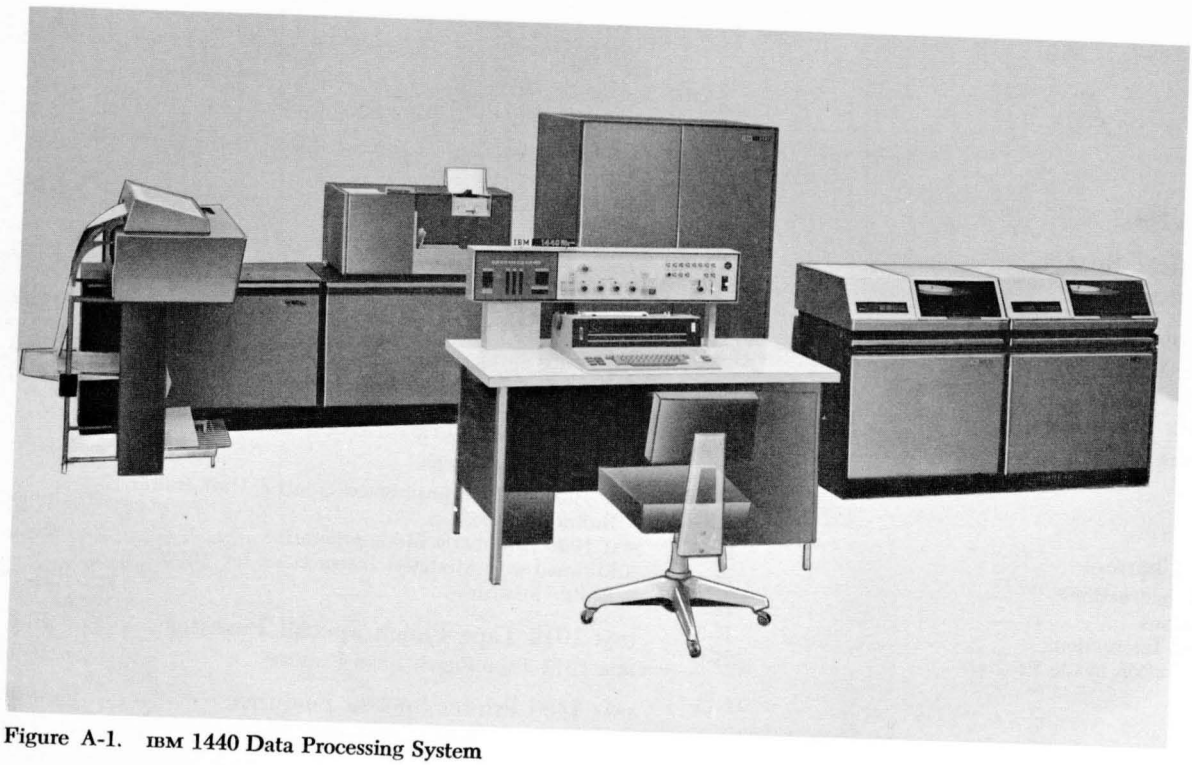


Figure A-1. IBM 1440 Data Processing System

IBM 1440 Data Processing System

The IBM 1440 Data Processing System (Figure A-1) represents a major advance in low-cost data processing systems. The IBM 1440 offers small companies the functional capabilities of large data processing systems, but at speeds and costs in keeping with their needs and abilities. The input and output devices of the 1440 enable it to be effective in system areas where there has long been a need for a data processing system but not the volume of work to justify such a system. Processing methods of the 1440 are similar to those of the IBM 1401 Data Processing System.

The IBM 1440 is a solid-state system with compact components and input/output devices. In addition to its features of compactness and low-cost, the 1440 presents a new concept in data processing with the introduction of the removable disk pack.

In 1953, the introduction of IBM magnetic tape systems provided data processing systems with the ability to process large volumes of input and output data at very high speeds. Magnetic tape offers the advantage of providing virtually unlimited storage capacity. In 1956, the RAMAC® disk file introduced a new concept in data processing, permitting, as it did, storage of large volumes of data that were accessible in a random sequence.

The IBM 1311 Disk Storage Drive for the IBM 1440 Data Processing System provides virtually unlimited random and sequential access storage. A disk pack containing 2,000,000 characters of information can be removed from the 1440 system and another pack put in its place. This operator-removable disk pack combines the large-volume and sequential-processing advantages of tape systems with the random-access abilities of a RAMAC file.

The ease of mobility of a disk pack (the weight of the pack is less than 10 pounds) and the simplicity of its removal from the drive means that 2,000,000 characters of data can be placed in the system within seconds. Data can be organized in the disk pack in *random* or *sequential* order; regardless of how the data is located on the disk pack, it can be retrieved by the system in a random or sequential order with equal facility, depending on individual requirements. Up to five disk drives, each equipped with one disk pack, can be attached "on line" to provide 10,000,000 characters of information available at one time (equivalent to 125,000, 80-column punched cards).

The 1440 is primarily a disk-storage oriented system, providing a group of balanced input/output devices to work in conjunction with the IBM 1441 Processing Unit and with the IBM 1311 Disk Storage Drive. For operations that require extensive calculating ability and do not need a disk storage, the 1440 can function as a card system.

The IBM 1440 is available in various configurations to satisfy the requirements of individual users. It can be ordered to meet the basic requirements of an accounting system, and then increased in size as data processing requirements increase. If the 1440 is expanded to its maximum size and data processing requirements continue to grow, procedures and systems developed for the IBM 1440 can be readily adapted for processing on the medium-size IBM 1401 Data Processing System. With continued expansion and growth, adaptation to larger equipment such as the IBM 1460, 1410, and System/360 Data Processing Systems can be made.

Magnetic tape capabilities are also available when the IBM 7335 Tape Adapter Feature is installed.

The Stored Program

The IBM 1440 Data Processing System performs its functions by executing a series of instructions at high speed. A particular set of instructions, designed to solve a specific problem, is known as a *program*. Because the 1440 stores its instructions internally, it is called a *stored program* system.

The 1440 system normally executes instructions sequentially. The system can also skip over a particular group of instructions, or otherwise change the sequence of the program. Branch instructions are provided in the system to make it possible to alter the program and take the next instruction from another area of the stored program. This function also makes it possible to repeat an instruction, or group of instructions, as often as desired.

A series of programmed tests determines the logical path of the program. These tests are made at various points in the program to control the course of program step execution for specific conditions that can arise during processing.

Variable Word Length

Stored programming involves the concept of *words*. A 1440 word can be a single character, or a group of characters, representing a complete unit of information. Because IBM 1440 words are not limited to a specific number of storage positions – i.e., have variable word length – and because each position of core storage is addressable, each word occupies only the number of core-storage locations actually needed for the specific instruction or data field.

Word Marks

The use of the variable-length instruction and data format requires a method of determining the instruction and data-word length. This identification is provided by a word mark. Word marks are illustrated by underlining the characters with which they are associated.

The word mark serves several functions:

1. Indicates the beginning of an instruction.
2. Defines the size of a data word.
3. Signals the end of execution of an instruction.

The rules governing the use of word marks are:

1. Predetermined locations for word marks are assigned in planning the program. These predetermined word marks are normally expected to remain in these locations throughout the complete program. The word marks are set into storage locations by a loading routine.
2. Word marks are not moved with data during processing, except when a *load* instruction (see No. 5 below) is used.
3. For an arithmetic operation, the *B-field* must have a defining word mark, and the *A-field* must have a word mark only when it is shorter than the B-field.
4. A load instruction moves the word mark and data from the A-field to the B-field, and clears any other word marks in the designated B-field, up to the length of the A-field.
5. When moving data from one location to another, only one of the fields need have a defining word mark, because the *move* instruction implies that both fields are the same length.
6. A word mark must be associated with the high-order character (operation code) of every instruction.
7. The 4-character BRANCH UNCONDITIONAL instruction, the 7-character SET WORD MARK, and CLEAR STORAGE AND BRANCH instructions are the only instructions that can be followed by a blank without a word mark. These instructions frequently terminate a series of program steps. For this reason, circuitry provisions have been made to eliminate the necessity of placing a word mark following the UNCONDITIONAL BRANCH, and 7-position SET WORD MARK and CLEAR STORAGE AND BRANCH instructions. All other instructions must be followed by a word mark.

Two operation codes are provided for setting and clearing word marks during program execution.

Stored-Program Instructions

All machine functions are initiated by instructions from the 1440 stored program. Because the 1440 uses the variable-word-length concept, the length of an instruction can vary from one to eight characters, depending on the operation to be performed.

Instruction Format

Mnemonic	*Op Code*	*A- or I-address*	*B-address*	*d-character*
X	X	XXX	XXX	X

Mnemonic. This is the mnemonic operation code that is used by the Autocoder processor program to designate the actual machine operation code.

Op Code. This is always a single character that defines the basic operation to be performed. A word mark is always associated with the operation code position of an instruction.

A-Address. This always consists of three characters. It can identify the units position of the A-field, or it can be used to select an input/output unit (card read-punch, disk storage unit, data transmission unit, paper tape reader, printer, tape punch, etc.).

I-Address. Instructions that can cause program branches use the I-address to specify the location of the next instruction to be executed if a branch occurs.

B-Address. This is a 3-character storage address that identifies the B-field. It usually addresses the units position of the B-field, but in some operations (such as move record or input/output operations) it specifies the high-order position of a record-storage area.

d-Character. The d-character is used to modify an operation code. It is a single alphabetic, numerical, or special character, positioned as the last character of an instruction.

Instruction Descriptions

Specific instructions have been described in a standard format:

Title. This is the description of the instruction.

Instruction Length. The length of an instruction can be either 1, 2, 4, 5, 7, or 8 characters. It cannot be either 3 or 6 characters long. Characters beyond the usable limit of eight do not affect the operation. Addressing advances (mod + 1) until the next word mark is sensed before the instruction is executed.

Most instructions for the 1440 must have a word mark following the instruction in core storage. This word mark is normally associated with the core-storage location immediately following the instruction itself.

Figure A-2 shows examples of the combinations possible in variable-length instructions.

Instruction Format. This is the format of the particular instruction described. The mnemonic operation code used for Autocoder is given.

Function. This is the function of the instruction.

Word Marks. This is the effect of the word marks with regard to data fields.

Timing. When the instruction-execution timing is always a constant, the actual time in milliseconds is given. When the instruction-execution time can vary because of field length or chaining, the formula is given. Figure A-3 is the key to the abbreviations used in the formulas.

Notes. These are special notations or additional information pertaining to the operation.

NUMBER OF POSITION	OPERATION	INSTRUCTION FORMAT
1	No Operation	Op Code N
2	Select Stacker	Op Code d-Character K 2
4	Unconditional Branch	Op Code I-Address B 400
5	Write Mark Tape	Op Code I-Address d-Character U %U2 M
7	Add	Op Code A-Address B-Address A 075 423
8	Write Tape	Op Code A-Address B-Address d-Character L %U1 731 W

Figure A-2. IBM 1440 Instruction Format Examples

Key to abbreviations used in formulas:

- L_A = Length of the A field
- L_B = Length of the B field
- L_C = Length of Multiplicand field
- L_I = Length of Instruction
- L_M = Length of Multiplier field
- L_P = Length of Product field
- L_Q = Length of Quotient field
- L_R = Length of Divisor field
- L_S = Number of significant digits in Divisor (excludes high-order zeros and blanks) or Number of Characters per Sector
- L_W = Length of A or B field, whichever is shorter
- L_X = Number of characters to be cleared
- L_Z = Number of characters back to rightmost zero in control field
- I/O = Timing for Input or Output cycles
- F_M = Forms movement times
- Σ = Number of fields included in an operation
- N_S = Number of disk sectors
- S_S = Number of characters in disk sector
- T_M = Tape movement time. This time is determined as follows:

  N_C = Number of characters in the record.

  C = Character rate. For the 7335, this is .050 ms 556 characters per inch.

  Read: $20.5 + CN_C$ ms = TAU interlocked.
  $7.7 + CN_C$ ms = Processing unit interlocked.

  Write: $20.3 + CN_C$ ms = TAU interlocked.
  $5.0 + CN_C$ ms = Processing unit interlocked.

  Note: See **Magnetic Tape Timing** section.

Figure A-3. Timing Formula Coding

ABBREVIATION	MEANING
A	A-address of the instruction
B	B-address of the instruction
NSI	Address of the next sequential instruction
BI	Address of the next instruction if a branch occurs
L_A	The number of characters in the A-field
L_B	The number of characters in the B-field
L_W	The number of characters in the A- or B-field, whichever is smaller
Ap	The previous setting of the A-address register
Bp	The previous setting of the B-address register
d*bb*	The d-character and blank in the units and tens position. The actual d-character is shown when possible.

Figure A-4. Address Registers after Operation Coding

Address Registers After Operation. The contents of the address registers are represented by the codes described in Figure A-4.

Chaining. This assists the programmer in determining whether instruction-chaining can be used effectively.

In some cases, chaining proves useful even though it would not ordinarily be used. For example, another instruction can be chained to the MOVE CHARACTERS AND EDIT instruction if the programmer can use the contents of the address registers to advantage. When considering the use of chaining, be certain that the contents of the address registers are valid for all conditions relating to the instructions involved. (Refer to the specific instruction section.)

Example. A practical application of the instruction is described and shown as a label for the 1440 Autocoder language, with the label of a typical actual machine address (in parentheses).

These examples for the instructions are representative, and are intended as exhibits of typical core-storage assignments, rather than specific, limited examples. Because the *Autocoder Assembler* usually establishes exact core-storage addresses for the many instructions of any given program, the programmer need not ordinarily be concerned, except when the program must be analyzed.

The few inflexible addresses of core-storage locations, such as index registers, are shown in the instructions as exact locations.

Assembled Instruction. This is the actual machine language instruction that is assembled by the Autocoder processor program from the symbolic entries shown in the example.

When an explicit Autocoder mnemonic is used, the op code, A-address, and d-modifier (when required) are automatically generated in most cases (refer to the specific operation in question).

Example: The coded instruction to cause information to print on the 1443 might be in the form: W (column 16 of the coding sheet), PRTOUT (written in column 21-26). Assume for the purpose of this example that the label PRTOUT actually represents core-storage location 0101. Autocoder would assemble this coded instruction into actual machine language as follows: M %Y1 101 W. In this instance, the machine-language op code (M), the A-address designating the 1443 (%Y1), and the d-modifier character defining the operation as a write (W) were all explicitly defined in the Autocoder mnemonic: W.

Example: When an explicit Autocoder mnemonic is *not* provided, or if the programmer uses a "general" mnemonic, the A-address and the d-modifier *must* be written as part of the operand: MU (column 16-17), unit address %Y1 (starting in column 21), B-address (starting next), and the specific d-character W (last). In this case, the results are the same: M %Y1 101 W.

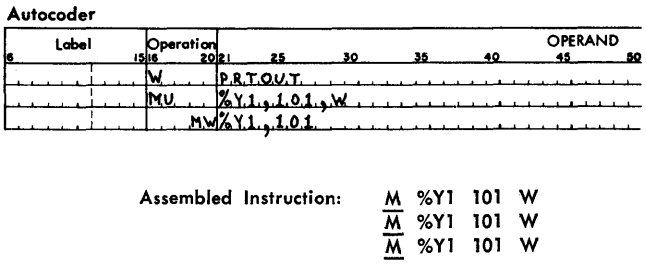


Figure A-5. Typical Autocoder Instruction Statements

Example: The instruction can be coded entirely in machine language, if desired. In this case, the actual op code is written in column 19, and the d-modifier character is written in column 20.

Figure A-5 illustrates three posible methods of coding to obtain the same result. The label PRTOUT could have been substituted for the actual machine address 101 if this label and core-storage location 101 had been equated elsewhere in the program.

Example: In other cases, the actual machine language op code is implied by the Autocoder mnemonic, but the d-modifier character must still be coded in the operand, such as for the BRANCH IF CHARACTER EQUAL (BCE) instruction. Here, the machine language op code (B) is explicit, but the flexibility of the d-character requires that the programmer code the d-modifier.

Programming Note, 1440 Autocoder Assembly

The 1440 Autocoder Assembler checks for the following valid Select Stacker(SS) mnemonic d-modifiers:

- 0 1442 Read/Punch stacker 2 (second attached device)
- 2 1442 Read/Punch stacker 2 (first attached device)
- # 1444 Read/Punch stacker 2

The programmer using the Direct Data Channel and other features or devices must recognize and check the SS mnemonic d-characters for accuracy, if they do not agree with the preceding list. The assembler halts and prints an error message if other d-modifier characters are used for the SS mnemonic. If this assembly halt occurs, refer to *Autocoder (On Disk) Program Specifications and Operating Procedures for* IBM *1401, 1440, and 1460* (Form C24-3259).

IBM 1441 Processing Unit

The IBM 1441 Processing Unit (Figure A-6) is the controlling center of the IBM 1440 Data Processing System. The processing unit can be divided into two sections:

1. The arithmetic-logical section
2. The control section

The arithmetic-logical section performs such operations as addition, subtraction, transferring, comparing, and storing. By adding the multiply-divide special feature, the 1441 can perform direct multiplication and division. This section also has logical ability – the ability to test various conditions encountered during processing and to take the action called for by the result.

The control section directs and coordinates the entire system as a single multipurpose machine. These functions involve controlling the input/output units and the arithmetic-logical operation of the processing unit, and transferring data to and from storage. This section directs the system according to the procedure originated by the programmer.

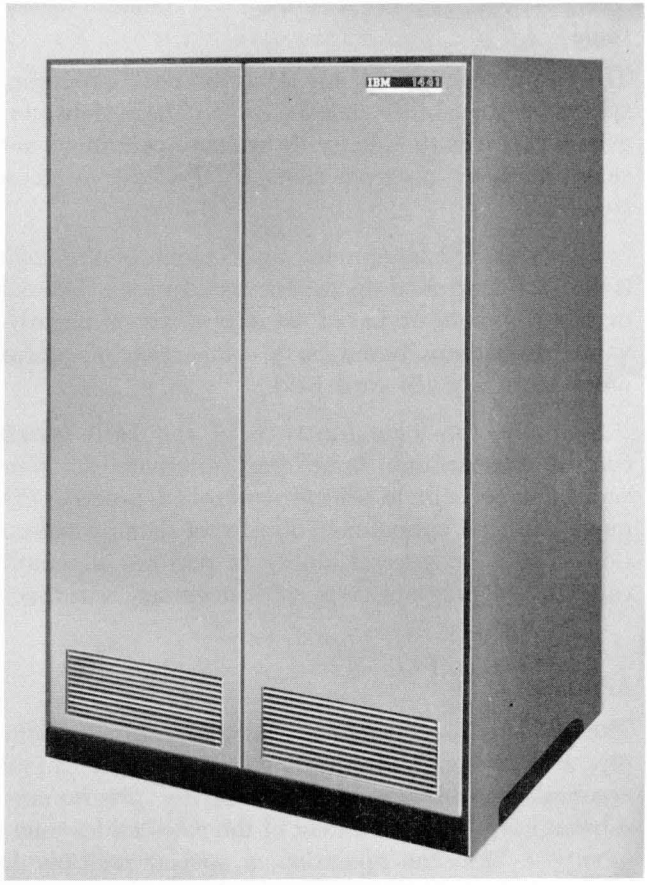


Figure A-6. IBM 1441 Processing Unit

Magnetic Core Storage

The IBM 1441 Processing Unit houses the magnetic-core storage area (Figure A-7) that is used by the 1440 system for storing the instructions and data. The data in each core-storage position is available in 11.1 microseconds. The design of the core-storage control circuits makes each position individually addressable. This means that an instruction can designate the exact storage locations that contain the data needed for that step.

The physical make-up of each core-storage location enables the IBM 1441 to perform arithmetic operations directly in the storage area. This is called *add-to-storage* logic.

Language

In the punched-card area of data processing, the language of the machine consists of holes punched in a card. As data processing needs increase, the basic card language remains the same. But in the transition from unit-record systems to the IBM 1440 Data Processing System, and from there to other computer systems, another faster, more flexible machine language emerges.

Just as each digit, letter in the alphabet, or special character is coded into a card as a punched hole or a combination of punched holes, it is coded into magnetic storage as a pattern of magnetized ferrite cores.

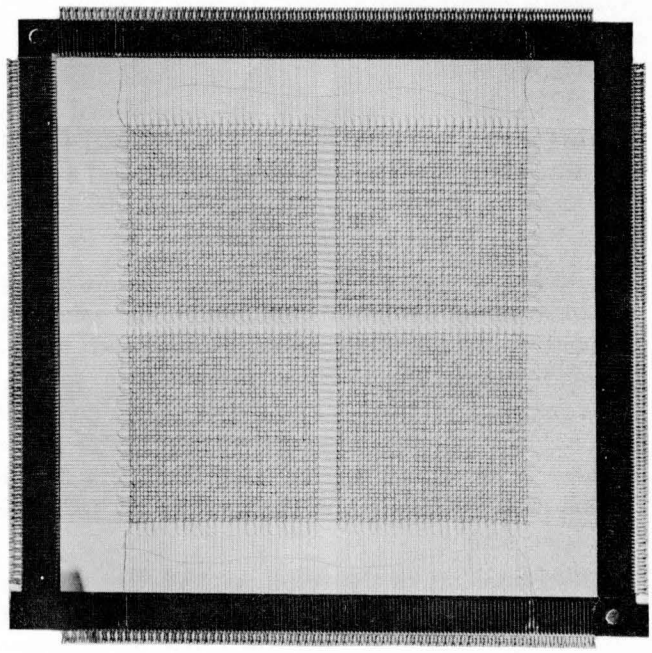


Figure A-7. Magnetic Core Storage

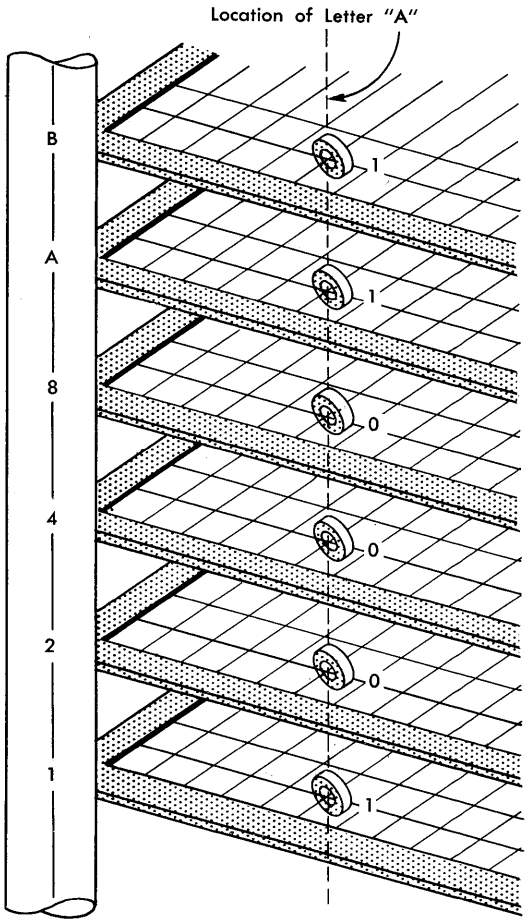


Figure A-8. The Letter A Represented in BCD Form in Core Storage

Many different code patterns can be set up. The internal code used in the IBM 1440 Data Processing System is called *binary-coded decimal* (Figure A-8). All data and instructions are translated into this code as they are stored.

The numbers 0 through 9 are represented by a single bit, or a combination of bits designated 1, 2, 4, 8. Disregarding the C- or check bit, bits 2 and 8 stand for 0, bits 1 and 2 for 3, bits 1 and 4 for 5, bits 2 and 4 for 6, bits 1, 2, and 4 for 7, and bits 1 and 8 for 9.

Letters and special characters are represented by a combinations of numerical bits (8421) and zone bits. B- and A-bits, in combination, correspond to the 12-zone punch. The B-bit corresponds to the 11-zone punch, and the A-bit to the 0-zone punch. The letter C, for example, which is the third letter in the 12-zone of the alphabet (card code 12-3), is a combination of BA21 bits. BA is the same as 12, and 21 is the same as 3.

This covers six of the seven possible bits that are used to represent a character. The seventh bit (C) is a built-in checking feature that the computer automatically supplies.

Note that the check bit is not part of the character configuration when the number of BA8421 bits that represent the character is odd. It appears only for those characters where the number of bits BA8421 is even. The automatic inclusion of the check bit changes the configuration of the character from an even number of bits to an odd number of bits. Thus, all characters shown in Figure I-1 are shown in the *odd-parity* mode.

Information introduced into the system is translated to the binary-coded-decimal form for use in all data flow and processing from that point on, until it is translated into printed output as reports and documents are written, or converted to punched-card code, for punched-card output. Converting input data to the 1441 internal code, and subsequently reconverting, is completely automatic.

Processing

Processing is the manipulation of data from the time it is introduced to the system as input until the desired results are ready for output. The following functions are performed in the IBM 1441 Processing Unit.

Logic

The logic function of any kind of data processing system is the ability to execute program steps; but even more, it is the ability to evaluate conditions and select alternate program steps on the basis of those conditions.

In unit-record equipment, an example of this logic is selector-controlled operations based on an X-punch or No X-punch, or based on a positive or negative value, or perhaps based on a comparison of control numbers in a given card field.

Similarly, the logic functions of the 1440 system control comparisons, branching (alternate decisions similar in concept to selector-controlled procedures), move and load operations (transfer of data or instructions), and the general ability to perform a complicated set of program steps with necessary variations.

Arithmetic

The IBM 1441 Processing Unit can add, subtract, multiply, and divide. Multiplication and division can be accomplished in any 1440 system, by programmed subroutines. When the extent of the calculations might otherwise limit the operation, a special multiply-divide feature is available.

Editing

As the term implies, editing adds significance to output data by punctuating and inserting special characters and symbols. The 1440 system has the ability to perform this function, automatically, with simple program instructions.

Internal Checking

Advanced circuit design is built into the 1440 to assure accurate results. Self-checking with the system consists of *parity* and *validity* checking.

Parity Checking

The IBM 1441 checks characters at various locations in the unit for odd-bit configurations. The 6-bit, binary-coded-decimal internal language used by the 1440 also has a check bit for odd-bit checking purposes, and a word-mark bit. The check bit is added to all characters that would otherwise have an even number of bits.

Example: A character P has a binary-coded decimal equivalent of B 4 2 1. The check bit is added to give this character an odd number of bits (C B 4 2 1).

If the character has a word mark associated with it, the word mark is included in the test for odd-bit parity.

Example: If the character P has a word mark, the check bit is not added because the bit configuration is odd (WM B 4 2 1).

Whenever a parity error occurs, a console light turns on, indicating the place where the error occurred (see IBM *1447 Console*, Form A24-3031).

Validity Checking

Validity checking is performed to detect illogical bit combinations within the systems. The type of validity checks performed are:

1. The output from the adder is checked for a logical numeric code.
2. The operation register is checked so that only valid operation codes are processed.
3. The storage address register is checked to make sure the core-storage addresses are valid addresses within the core-storage address range of that particular processing unit. Depending on the core-storage size, the units and/or hundreds address positions contain zone bits that specify blocks of addresses. Refer to *Addressing System* section for detail information. These zone-bit combinations are checked to make sure the combinations are addressing an available core-storage address. A check is made to see if the lower or upper limits of core storage have been passed. This check is called an end-around check and is made at all times except for three special operations: clear storage, storage scan, and storage print-out. The modification of the low-order position of core storage by −1, except during a clear storage operation, or the modification of the high-order position of core storage by +1, except during storage scan and storage print-out operations, causes an invalid operation and a system stop.
4. Of the 4096 bit configurations (2 to the 12th power) possible in a given card column, only 64 are *recognizable* characters. (See *Card Image Feature* section.) All other bit configurations are considered invalid during the data transfer from the read side of the card read-punch into core storage. A detected check condition turns on the card read validity check light. Depending on the I/O check stop switch setting on the 1447, the system also stops or a program-testable indicator is set ON.

System Checking Conditions

The following procedures should be observed if a system check occurs. (These check conditions may also indicate a system failure.)

Process Error
(Functions Not Affected by Process-Check Stop Switch)

These errors indicate that one of the following conditions has occurred. (Regardless of the position of the process-check stop switch, a system stop results.)

Op Register: An invalid op code has been sensed. Either the system feature that uses this op code is not on the system, or a programming error is indicated. Determine the cause of the failure before attempting to continue the job.

Storage Address Register: An invalid address has been read into the address register(s). This may indicate an attempt to use an uninstalled feature.

Process Error
(Functions Affected by Process-Check Stop Switch)

These errors can be reset by the BRANCH ON PROCESS ERROR instruction (% d-modifier), or by operation of the check-reset key on the console. The branch instruction is only effective when the process-check stop switch is OFF.

A-Register Error: This error indicates that an invalid (or an out-of-parity) character has been sensed in the A-register.

B-Register Error: This error indicates that an invalid (or out-of-parity) character has been sensed in the B-register.

Arith Error: This indicates that an arithmetic error occurred in the logic control unit.

Storage Error: This indicates that an error occurred in main core storage.

Printer Error

If the print-check light is on:

1. Note the contents of the I-address register on the console.
2. Press the check-reset key on the printer.
3. Manually branch to the address location where the error occurred. This address is equal to the previously noted contents of the I-address register, minus 8.
4. Press the start key on the printer to re-execute the print command.

Note: The programmer can branch past a printer error if the I/O check-stop switch is off by executing a BRANCH IF PRINTER ERROR instruction (≠ d-modifier).

Reader Error

If the reader-check light is on:

1. Remove the cards in the stacker(s).
2. Remove the cards (if any) in the hopper.
3. Non-process run-out the remaining cards into the stacker.
4. Place the two non-processed cards in the read hopper.
5. Press the start key on the card reader, and the start key on the console, to re-execute the read operation.

Note: The programmer may wish to branch past a reader error. If the I/O check-stop switch is off, a BRANCH IF READER ERROR instruction (? d-modifier) can be executed.

Punch Error (1442)

If the punch-check light is on:

1. Discard the last card in the stacker.

Note: This card must be retained if it contains source information.

2. Press the start key on the 1442 and on the console.
3. The card will be repunched.

Note: If the programmer elects to ignore the punch error, or enter a subroutine in case of a punch error, he can program a BRANCH IF PUNCH ERROR instruction (! d-modifier), if the I/O check-stop switch is off.

Punch Error (1444)

If the punch-check light is on:

1. Remove and save any cards in the punch stacker.
2. Remove remaining cards from the hopper.
3. Non-process run-out the cards in the machine.
4. Discard the error card (last card in stacker before the non-process run-out operation).
5. Discard the two cards that were run out of the machine feed.
6. Press the start keys on the 1444 and on the console to continue processing.
7. The card will be repunched.

See *Note* following *Punch Error (1442).*

Addressing

Instructions and data used for processing in the 1440 system are contained in the core-storage area. Each core-storage position has its own unique address. The IBM 1441 Processing Unit is available in five different core-storage capacities.

Model A2:	2,000 core-storage positions
Model A3:	4,000 core-storage positions
Model A4:	8,000 core-storage positions
Model A5:	12,000 core-storage positions
Model A6:	16,000 core-storage positions

Addressing System

Every core-storage position in the IBM 1440 Data Processing System can be addressed with a 3-character address. To address 16,000 core-storage positions with only three characters, various zone-bit configurations are added over the hundreds position and units position of the address.

The zone-bit configuration over the *hundreds* position specifies the thousands position of core storage up to 3999. No A- or B-bit over the hundreds position specifies that the address is the actual address (000-999). An A-bit over the hundreds position of the address specifies another group of 1,000 core-storage positions (1000-1999). A B-bit over the hundreds posi-

CODED ADDRESSES IN STORAGE		
ACTUAL ADDRESSES		3-CHARACTER ADDRESSES
000 to 999	No zone bits	000 to 999
1000 to 1099	A-bit, using 0-zone	≠00 to ≠99
1100 to 1199		/00 to /99
1200 to 1299		S00 to S99
1300 to 1399		T00 to T99
1400 to 1499		U00 to U99
1500 to 1599		V00 to V99
1600 to 1699		W00 to W99
1700 to 1799		X00 to X99
1800 to 1899		Y00 to Y99
1900 to 1999		Z00 to Z99
2000 to 2099	B-bit, using 11-zone	I 00 to I 99
2100 to 2199		J00 to J99
2200 to 2299		K00 to K99
2300 to 2399		L00 to L99
2400 to 2499		M00 to M99
2500 to 2599		N00 to N99
2600 to 2699		*O00 to O99
2700 to 2799		P00 to P99
2800 to 2899		Q00 to Q99
2900 to 2999		R00 to R99
3000 to 3099	A-B-bit, using 12-zone	?00 to ?99
3100 to 3199		A00 to A99
3200 to 3299		B00 to B99
3300 to 3399		C00 to C99
3400 to 3499		D00 to D99
3500 to 3599		E00 to E99
3600 to 3699		F00 to F99
3700 to 3799		G00 to G99
3800 to 3899		H00 to H99
3900 to 3999		I00 to I99

*Letter O followed by two digits.

Figure A-9. Core-Storage Address Coding (000 to 3999)

tion of the address specifies another group of 1,000 core-storage positions (2000-2999). Both the A- and the B-bit over the hundreds position of the address specify another group of 1,000 core-storage positions (3000-3999). By using these zone-bit combinations, 4,000 positions of core storage can be addressed with a 3-character address (Figure A-9).

The same principle used to specify the various 1,000-blocks of core storage is also used to specify core-storage blocks of 4,000 positions. The zone-bit configuration over the *units* position specifies which block of 4,000 core-storage positions is being addressed.

No A- or B-bit over the units position specifies the 4,000-block in core storage that contains positions 0000-3999. An A-bit over the units position specifies the 4,000-block in core storage that contains positions 4000-7999. A B-bit over the units position specifies the 4,000-block in core storage that contains positions 8000-11999. Both the A- and the B-bit over the units position specifies the 4,000-block in core storage that contains positions 12000-15999. By combining the 3-digit address with zone-bit combinations over the hundreds and/or units position, it is possible to address 16,000 core-storage positions (Figure A-10).

Data-Field Addressing

A data field in core storage is addressed by specifying the low-order (units) position of the field in the A- or B-address of the instruction. The data field is usually read from right to left until a word mark in the high-order position is sensed.

ACTUAL ADDRESSES	ZONE BITS OVER HUNDREDS POSITION	ZONE BITS OVER UNITS POSITION	3-CHARACTER ADDRESSES
0000 to 0999	No Zone Bits	No Zone Bits	000 to 999
1000 to 1999	A-Bit (Zero-Zone)	No Zone Bits	≠00 to Z99
2000 to 2999	B-Bit (11-Zone)	No Zone Bits	!00 to R99
3000 to 3999	AB-Bits (12-Zone)	No Zone Bits	?00 to I99
4000 to 4999	No Zone Bits	A-Bit (Zero-Zone)	00≠ to 99Z
5000 to 5999	A-Bit (Zero-Zone)	A-Bit (Zero-Zone)	≠0≠ to Z9Z
6000 to 6999	B-Bit (11-Zone)	A-Bit (Zero-Zone)	!0≠ to R9Z
7000 to 7999	AB-Bits (12-Zone)	A-Bit (Zero-Zone)	?0≠ to I9Z
8000 to 8999	No Zone Bits	B-Bit (11-Zone)	00! to 99R
9000 to 9999	A-Bit (Zero-Zone)	B-Bit (11-Zone)	≠0! to Z9R
10000 to 10999	B-Bit (11-Zone)	B-Bit (11-Zone)	!0! to R9R
11000 to 11999	AB-Bits (12-Zone)	B-Bit (11-Zone)	?0! to I9R
12000 to 12999	No Zone Bits	AB-Bits (12-Zone)	00? to 99I
13000 to 13999	A-Bit (Zero-Zone)	AB-Bits (12-Zone)	≠0? to Z9I
14000 to 14999	B-Bit (11-Zone)	AB-Bits (12-Zone)	!0? to R9I
15000 to 15999	AB-Bits (12-Zone)	AB-Bits (12-Zone)	?0? to I9I

Figure A-10. IBM 1440 Addressing System (000 to 15,999)

Instruction addressed by high-order position

STORAGE ADDRESS	400	401	402	403	404	405	406	407 (NSI)
INSTRUCTION	A	5	4	2	5	6	0	WM Op code

The word mark associated with the next sequential instruction (NSI) stops the reading of this instruction.

A-address

STORAGE ADDRESS	536	537	538	539	540	541	542	543
DATA	0	0	2	5	3	4	7	8

A-field

Word mark identifies high-order position of A-field.

B-address

STORAGE ADDRESS	553	554	555	556	557	558	559	560	561
DATA	0	4	6	0	1	2	3	1	4

B-field

Word mark identifies high-order position of B-field.

Figure A-11. Data and Instruction Addressing

Instruction Addressing

An instruction in core storage is addressed by giving the high-order (operation code) position of the instruction. All operation codes must have a word mark. (This word mark is normally set by the loading routine when the instructions are loaded.) The machine reads an instruction from left to right until it senses the word mark asociated with the next sequential instruction. The final instruction in the program must have a word mark set at the right of its low-order position. (The word mark is not needed if the instruction is an UNCONDITIONAL BRANCH, SET WORD MARK, or CLEAR STORAGE AND BRANCH).

Example: Instruction address 400 (Figure A-11) contains the operation code for the following instruction:

Op Code	*A-address*	*B-address*
A	542	560

When this instruction is executed, the data in the A-field is added to the data in the B-field:

```
  0025347
+ 04601231
----------
  04626578
```

The result is stored in the B-field.

Core-Storage Area Assignment

There are two areas in core storage that are used for specific purposes. Core-storage positions 001-081 are used in conjunction with a program-load operation and core-storage positions 087-089, 092-094, and 097-099 are used as three index registers when the indexing and store address register special feature is installed. All other core-storage positions are always available for normal use, and the areas just mentioned can be used for other system operations when they are not being used as specified.

1440 Register Operation

The IBM 1440 Data Processing System operates on and processes data to produce a desired result by executing a series of instructions. A series of instructions designed to solve a problem is known as a *program*. Because these instructions are retained in core storage, it is more properly called a stored program.

The processing unit must interpret an instruction and perform the function prescribed by the instruction. To do this, various types of devices that are capable of receiving information, storing it, and transferring it as directed by control circuits are used. These devices are known as *registers*. The 1440 has seven registers, four are address registers and three are character registers (Figure A-12).

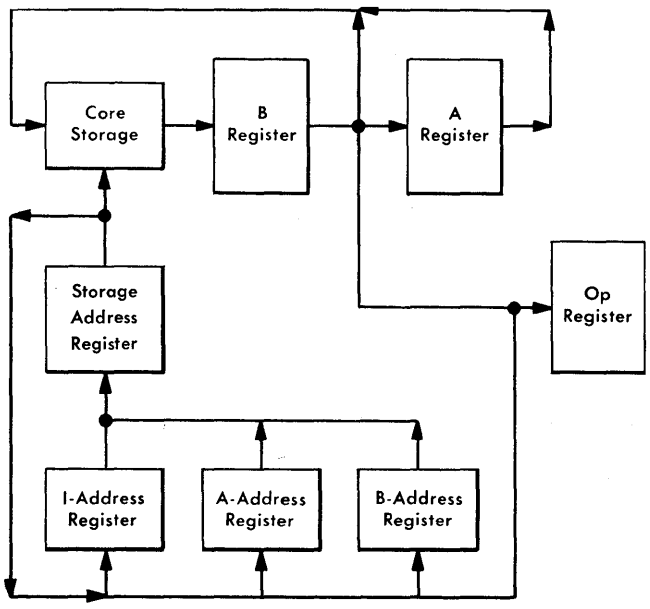


Figure A-12. Processing Unit Registers

Address Registers

There are four address registers in the IBM 1441 Processing Unit. One register controls the program sequence, and two other registers control the data transfer from one storage location to another. The fourth register specifies which storage location is active during a particular storage cycle.

The tens position of an address register never actually stores an alphabetic or special character. This information is decoded as it is read in, and the *numeric* portion of the character is stored. The decoding determines unit-addressing or indexing.

I-Address Register. The I- (Instruction) address register always contains the storage location of the next instruction character to be used by the stored program. The number in this register is increased by one as the instruction is read from left to right.

A-Address Register. The A-address register contains the storage address of the data in the A-address portion of an instruction. Normally, as the instruction is executed, the number in this register is decreased by 1 after each storage cycle that involves the A-address.

Note: If the A-address portion of the instruction does not contain a core-storage address (for example %Gx) the contents of the A-address register are not altered as the instruction is executed.

B-Address Register. This register contains the storage location of the data in the B-address portion of an instruction. Normally, as a storage cycle involving the B-address is executed, the storage address in the B-address register is decreased by 1.

Storage-Address Register. The storage-address register always contains the address of the core-storage position that will be involved in any data movement during that particular machine cycle.

Character Registers

The A- and B-character registers, the Arith register, and the Op-register are single-character registers used to store data during the execution of an instruction.

Op Register. The Op- (Operation) register stores the operation code of the instruction in process for the duration of the operation. The operation code is stored in BCD code, including the check bit but excluding the word mark.

B-Register. Each character leaving core storage enters the B-register. The character is stored in 8-bit form (BCD code, check bit, and word mark). The B-register is reset and filled with a character from core storage on every storage cycle.

A-Register. The A-register is reset and filled with the character from the B-register during each storage cycle that involves the A-address, and during all instruction cycles except the first and last I- (Instruction) cycle of each instruction. Data is stored in 8-bit form.

Arith Register. This 6-bit register contains the results of the B-register±the A-register. During arithmetic functions, this character is normally routed to main storage.

Note: Information can be written back into core storage directly from either the A- or B-register, or from the Arith register.

Figure A-13 shows the I-phase of an operation and gives a detailed schematic for loading a 7-character instruction on the operation-code register, in the A-and B-registers and in the I-, A-, and B-address registers. Eight storage cycles are required to load the complete instruction in the register. Each storage cycle requires .0111 ms.

Note: The A- and B-address registers contain 3-character addresses. The addresses shown in this schematic are 4-digit addresses because the storage display lights on the console show 4-digit addresses. Refer to Figure A-9 for the relationship between 3- and 4-digit addresses.

Chaining Instructions

In some programs, it is possible to perform a series of operations on several fields that are in consecutive storage locations. Some of the basic operations, such as add, subtract, move, and load, can be *chained* so that less time is required to perform the operations, and space is saved in storing instructions. Here is an example of the chaining technique: assume that four 5-position fields stored in sequence are to be added to four other sequential fields. This operation could be done using four 7-character instructions:

```
A  700  850
A  695  845
A  690  840
A  685  835
```

At the completion of the first instruction, the A-address register contains 695 and the B-address register contains 845. These are the same numbers that are in the A- and B-addresses in the second instruction. (Executing the second and third instructions also results in A- and B-addresses that are the same as the A- and B-addresses of the third and fourth instructions.) Eighty storage cycles would be required to

CYCLE	OPERATION	Instruction Location: A 197 | 5 198 | 6 199 | 7 200 | T 201 | 1 202 | 2 203 | S 204
I-Op	The operation code enters the B-register and the Op-register. Because this is the first I-cycle, the A-register is undisturbed.	I Register: 0 1 9 7 — B Register: A — A Register: ? — Cycle 1 OP Register: A — A Address Register: ? ? ? ? — B Address Register: ? ? ? ?
I-1	The A-address register is reset to blanks during the first part of the cycle for all instructions. The B-address register is reset to blanks during the first part of the cycle for all operations except Move, Load, Store A- and Store B-address Register operation. During the I-1 cycle, the second instruction character (first character of the A-address) enters the thousands and hundreds positions of the A- and B-address registers and the A-register by the way of the B-register.	I Register: 0 1 9 8 — B Register: 5 — A Register: 5 — Cycle 2 OP Register: A — A Address Register: 0 5 b b — B Address Register: 0 5 b b
I-2	The third character of the instruction enters the tens position of the A- and B-address registers, and the A-register through the B-register.	I Register: 0 1 9 9 — B Register: 6 — A Register: 6 — Cycle 3 OP Register: A — A Address Register: 0 5 6 b — B Address Register: 0 5 6 b
I-3	The fourth instruction character enters the units position of the A- and B-address registers, and the A-register through the B-register.	I Register: 0 2 0 0 — B Register: 7 — A Register: 7 — Cycle 4 OP Register: A — A Address Register: 0 5 6 7 — B Address Register: 0 5 6 7
I-4	The B-address register is reset at the beginning of this cycle. The fifth instruction character (first character of the B-address) enters the hundreds position of the B-address register, and the A-register through the B-register.	I Register: 0 2 0 1 — B Register: T — A Register: T — Cycle 5 OP Register: A — A Address Register: 0 5 6 7 — B Address Register: 1 3 b b
I-5	The sixth instruction character goes to the tens position of the B-address register, and the A-register through the B-register.	I Register: 0 2 0 2 — B Register: 1 — A Register: 1 — Cycle 6 OP Register: A — A Address Register: 0 5 6 7 — B Address Register: 1 3 1 b
I-6	The seventh character of the instruction (last character of the B-address) enters the units position of the B-address register and the A-register through the B-register.	I Register: 0 2 0 3 — B Register: 2 — A Register: 2 — Cycle 7 OP Register: A — A Address Register: 0 5 6 7 — B Address Register: 1 3 1 2
I-7	The first character of the next instruction enters the B-register only. Because this is the last I-cycle for this instruction, the A-register and the Op-register, the A- and B-address registers are undisturbed. The detection of a word mark associated with this character signals the machine that this is the Op code for the next instruction. The loading operations stops, and the instruction that was just loaded is executed. Note that the I-address register contains the address of the high-order position of the next sequential instruction.	I Register: 0 2 0 4 — B Register: S — A Register: 2 — Cycle 8 OP Register: A — A Address Register: 0 5 6 7 — B Address Register: 1 3 1 2

Figure A-13. Instruction Loading Schematic

execute these instructions, thus using up .888 ms. Also, 28 storage positions are required to store these instructions.

By taking advantage of the fact that the A- and B-address registers contain the necessary information to perform the next instruction, this same sequence of operations can be excuted as follows:

```
A  700  850
A
A
A
```

Connecting instructions together in this manner is called *chaining*. The first add instruction contains both the A- and B-addresses. The following three instructions contain only the operation code for those instructions. The A- and B-addresses are the results left in the A- and B-address registers from the previous instruction. This type of operation requires 62 storage cycles, and takes .688 ms to execute. Storing these chained instructions requires only ten storage positions.

The ability to chain a series of instructions does not depend on the use of the same operation code. Chained instructions may have various op codes. To be operated on, the A-fields must be in sequence, and the B-fields must be in sequence. *Example:*

```
A  900  850
M
A
M
```

Assume that the data fields are each ten characters long:

The ten characters at location 900 were added to 850.
The ten characters at location 890 were moved to 840.
The ten characters at location 880 were added to 830.
The ten characters at location 870 were moved to 820.

The description of each instruction includes the contents of the address registers after the operation has been performed. Figure A-4 shows the abbreviations that indicate the contents of these registers.

By using this information, the programmer can determine the status of the registers and decide whether chaining is practical in specific cases.

Note: Instructions that do not contain core-storage addresses cannot be chained. For example, M %Gn xxx R is a READ CARD instruction. The card read-punch is signaled as the machine reads the instruction. Although the A-address register contains %7n after the operation, chaining is impossible because the machine does not select the unit from the contents of the A-address register.

Most single-address instructions (op code and an A-address) cause the A-address to be inserted in both the A-address and B-address registers (for example, A xxx). However, execution of MOVE, LOAD, or STORE ADDRESS REGISTER instructions does not disturb the B-address register, and permits the programmer to use the previous contents of that register as part of the instruction.

All branch instructions (op code and I-address) depend on whether the indexing and store address register special feature is installed on the system:

1. With the special feature installed, the B-address register contains the address of the next sequential instruction, if a branch occurs.
2. Without the special feature installed, the B-address register is cleared to blanks whenever a branch occurs.

Address Modification

It becomes necessary in some 1440 programs to perform the same operations repetitively, with a change only in the A- or B-address. Changing of an address while retaining the rest of the instruction is called *address modification.* Address modification can result in savings in the number of program steps and in storage requirements. In some cases, the program itself determines if, and how, addresses are to be changed to perform the correct program steps for conditions arising during data processing.

The methods that can be used to modify addresses on a specific system depend on the core-storage capacity of that system.

On 1440 systems equipped with 2,000 or 4,000 positions of core storage, address modification is accomplished by either using modulus 4 arithmetic or installing the indexing and store address register special feature.

On 1440 systems equipped with more than 4,000 positions of core storage, the two previously mentioned methods of address modification can be used. However, these systems have a MODIFY ADDRESS instruction that greatly simplifies address modification.

Modulus 4 Arithmetic Method

When modifying addresses by modulus 4 arithmetic, the modified address should be located in the same 4,000-block of core storage as the original address.

This is because a zone-bit overflow of over three in the hundreds position of the address cannot be transferred to the units position of the address.

To set up a workable modulus 4 system, these digital values are assigned the four possible zone-bit configurations that appear in the hundreds position:

No A, No B-bit = 0
A-bit = 1
B-bit = 2
A- and B-bit = 3

As can be seen, the highest possible digit is three. Values in excess of three are equal to that value minus four. For example, a value of five is represented as a value of 1 (Figure A-14).

A + A	= B	or	1 + 1 = 2
A + B	= AB	or	1 + 2 = 3
B + B	= NoANoB	or	2 + 2 = 0
A + AB	= NoANoB	or	1 + 3 = 0
A + NoANoB	= A	or	1 + 0 = 1
B + AB	= A	or	2 + 3 = 1
B + NoANoB	= B	or	2 + 0 = 2
AB + AB	= B	or	3 + 3 = 2

Figure A-14. A-Bit and B-Bit Values

Address modification to a higher address in the 000-999 address range is:

Increase address 472 by 345
472 + 345 = 817

This is a normal add operation with no overflow involved.

Address modification to an address greater than 1000 is:

Increase address 912 by 314

912 + 314 = 1226 or S 26

S = A2 (*Overflow in high-order position sets an A-bit using modulus 4 arithmetic and turns on the arithmetic overflow indicator.*)

Increase address 1754 (X54) by 1204 (S04)

1754 + 1204 = 2958
X54 + S04 = R58
X = (A7)
S = (A2)

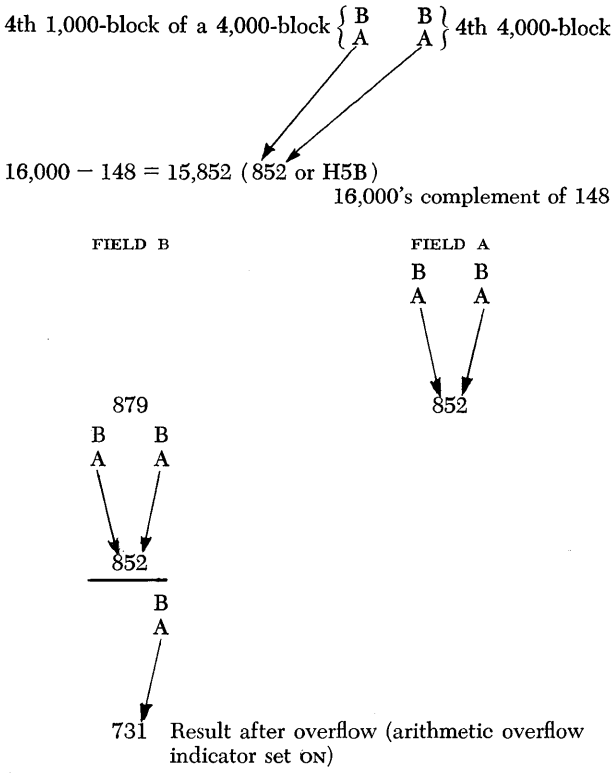
Using the rules of modulus 4 arithmetic, A + A = B-bit, the new address is:

958 with a B-bit over the high-order position (B9 = R) or R58 (2958).

To decrease an address, a different means must be used. Modulus 4 arithmetic operates for addition only. Decreasing an address requires the addition of a complement, rather than doing a conventional subtract operation.

In systems equipped with 2,000 or 4,000 core-storage positions, the 16,000's complement of the decrement figure is added to the address to be modified (modulus 16 arithmetic).

Decrease address 879 by 148
879 − 148 = 731



The add operation is performed as shown. The A-field figure is added to the B-field figure. The digital result is 731 and the arithmetic overflow indicator is set ON. Because an add operation has taken place, the units position ends up with a plus sign (an A- and a B-bit). The arithmetic overflow in the hundreds position adds an A-bit to the A- and B-bits already there, resulting in a zone-bit configuration of no A- and no B-bit (see Figure A-14). The A-bit addition increases the zone-bit value to 16. A value of 16, according to modulus 16 rules, has a new address value of 0 (000-999 core-storage address block). This means that 731 is the actual address.

Modulus 4 arithmetic is normally used in 1440 systems that contain 2,000 or 4,000 core-storage positions. With care, this address modification method could be used on systems with more core-storage capacity.

However, this is not generally practical because 1440 systems with more than 4,000 core-storage positions are equipped with the MODIFY ADDRESS instruction.

Modify Address Instruction Method

IBM 1440 systems with more than 4,000 core-storage positions can easily modify any address by using the MODIFY ADDRESS instruction.

Modify Address (Two Addresses)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MA	#	AAA	BBB

Function. This instruction causes the 3-character field, specified by the A-address (A-field), to be added to the 3-character field specified by the B-address (B-field). The result is stored in the B-field. The three numerical portions and the zones of the units and hundreds positions of the B-field make up the 3-character result. For example:

Location	*Contents*	*3-Character Address*	*Actual Address*
A-address	A-field	100	100
B-address	B-field	L2F	14326
	B-field	M2F	14426

Word Marks. Word marks are not affected, and are not required to define the A- or B-fields. If word marks are present, they are ignored and remain unchanged in both fields.

Timing. $T = .0111\ (L_I + 9)$ ms.

Note: Rules for the addition of zone bits are the same as in modulus 4 arithmetic, with one addition. This instruction makes it possible to reflect the hundreds position zone-bit overflow in the units position when the address is modified to a higher 4,000-block of core storage. When a zone-bit overflow occurs during the hundreds position modification, an additional cycle is executed to adjust the units position zone-bit configuration.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-3	B-1 or B-3

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Add the 3-character address labeled ADDA (0985) to the 3-character address labeled ADDB (1313); Figure A-15.

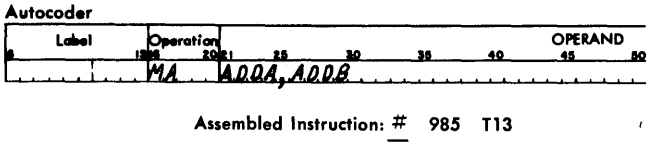


Figure A-15. Modify Address (Two Addresses)

Modify Address (One Address)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
MA	#	AAA

Function. This format of the MODIFY ADDRESS instruction causes the 3-character field, specified by the A-address, to be added to itself. The result is stored in the A-field.

Word Marks. Word marks are not required to define the A-field. If they are present, they are ignored and remain undisturbed in the A-field.

Timing. $T = .0111\ (L_I + 9)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-3	A-1 or A-3

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Double the address labeled ADDC (2956), and store the result at ADDC (Figure A-16).

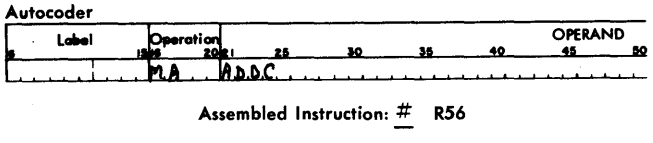


Figure A-16. Modify Address (One Address)

Indexing Method

Any 1440 system can modify addresses by installing the indexing and store address register special feature. A complete description of this feature can be found in the *Special Features* section.

The operations performed by an IBM 1440 Data Processing System can be arranged into these general classifications:

1. Arithmetic operations
2. Logic operations
3. Data-moving operations
4. Miscellaneous operations
5. Edit operation

Arithmetic Operations

The IBM 1440 Data Processing System adds and subtracts, by applying the add-to-storage method of operation. The two factors to be combined are added within core storage without the use of special accumulators or counters. Because any storage area can be used as an accumulator field, the capacity for performing arithmetic functions is not limited by standard-size accumulators or by a predetermined number of accumulators within the system. In arithmetic operations, the 1440 system considers blanks and zeros the same. An unsigned field is considered positive by the system.

All arithmetic functions are performed under complete algebraic sign control. The sign of a factor is determined by the combination of zone bits in the units position of the fields specified by the instruction being executed.

SIGN	BCD CODE BIT CONFIGURATION	CARD CODE CONFIGURATION
Plus	No A- or B-Bit	No Zone
Plus	A- and B-Bits	12 Zone
Minus	B-Bit Only	11 Zone
Plus	A-Bit Only	0 Zone

Figure B-1. Sign Bit Equivalents

TYPE OF OPER.	A-FLD. SIGN	B-FLD. SIGN	TYPE OF ADD CYCLE	SIGN OF RESULT
ADD+	+	+	True Add	+
		−	Compl. Add	Sign of Field with larger magnitude
	−	+	Compl. Add	
		−	True Add	−
SUBTRACT−	+	−	True Add	−
		+	Compl. Add	Sign of Field with larger magnitude
	−	−	Compl. Add	
		+	True Add	+

Figure B-2. Types of Add Cycles and Sign of Result

Figure B-1 shows the four possible combinations of zone bits and the values of the signs they represent.

The standard machine method of signing a field is to indicate a positive factor with A- and B-bits (12-zone), and to indicate a negative factor with a B-bit (11-zone).

The arithmetic operations in the IBM 1440 Data Processing System are performed by using one of two types of add cycles incorporated in the system. The two types of add cycles are:

1. true add
2. complement add

The type of add cycle performed depends on the arithmetic operation and the signs and values of the two factors involved (Figure B-2).

Because all arithmetic operations are performed with algebraic sign control, the *sign* of the result depends both on the operation, and on the magnitude and signs of the factors involved (Figure B-3).

	Sign-Bits of A-Field						A	A	A	A	B	B	B	B	AB	AB	AB	AB
	Sign-Bits of B-Field			A	B	AB		A	B	AB		A	B	AB		A	B	AB
ADD	Resultant Sign	When: A > B		A	AB	AB		A	AB	AB	B	B	B	B		A	AB	AB
		When: A ≤ B		A	B	AB		A	B	AB	AB	AB	B	AB		A	B	AB
SUB	Resultant Sign	When: A > B	B	B	B	B	B	B	B	B		A	AB	AB	B	B	B	B
		When: A ≤ B	AB	AB	B	AB	AB	AB	B	AB		A	B	AB	AB	AB	B	AB

Figure B-3. Zone-Bit Table for Add and Subtract Operations

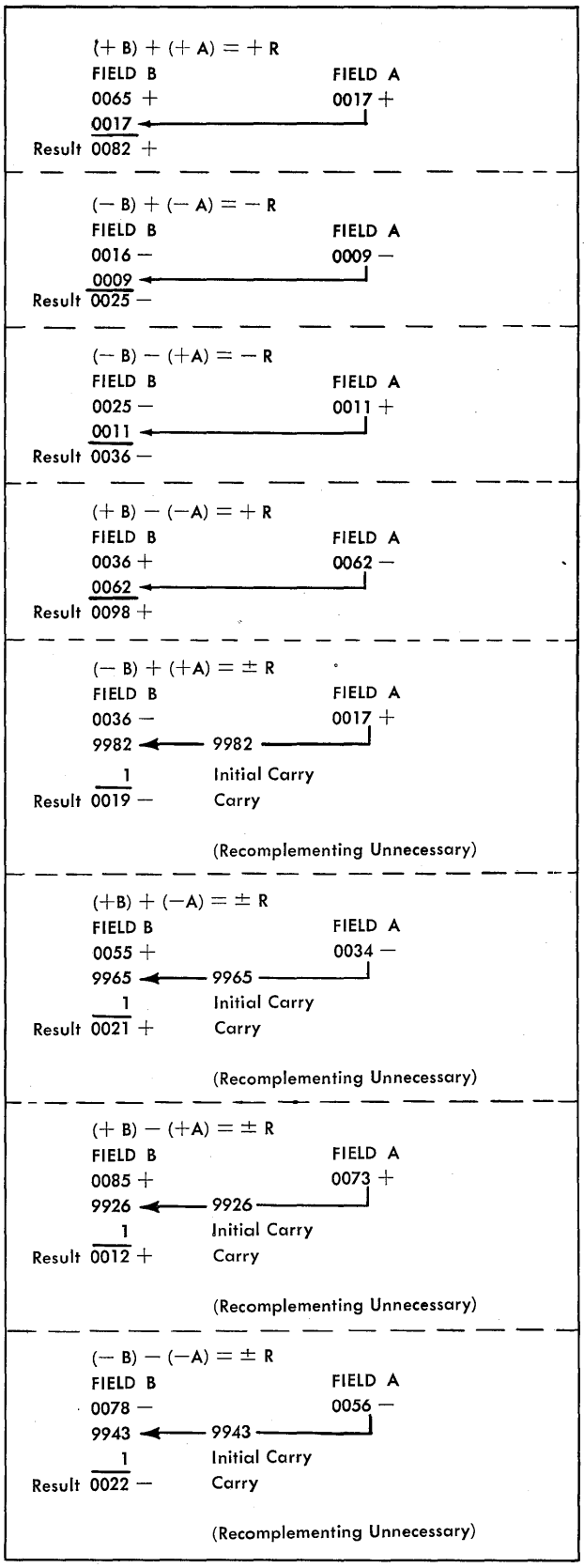


Figure B-4. True-Add and Complement-Add Cycle Examples

True Add

A true-add cycle is specified when the total number of minus signs is an even number (0 or 2). The signs considered are the signs of the factors and the sign of the operation.

The sign of the result after a true-add cycle carries the original sign of the B-field when either an add or a subtract operation is performed (Figure B-4).

Complement Add

An uneven number of minus signs (1 or 3) specifies a complement-add cycle. The system converts the A-field factor to its nines complement figure and adds it to the B-field factor (plus one initial carry). The

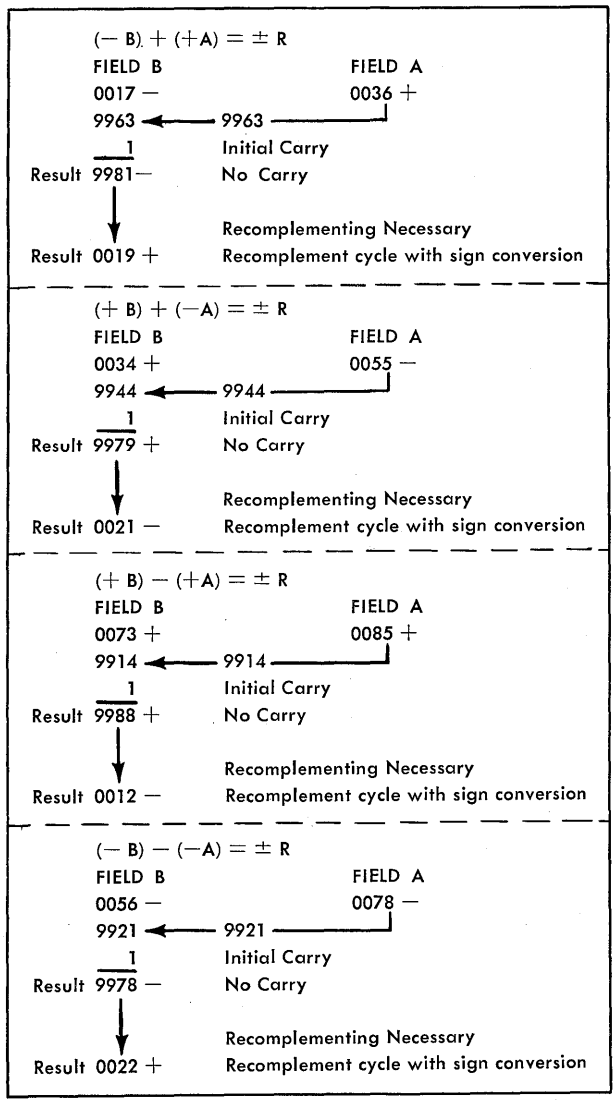


Figure B-5. Complement-Add (with Recomplement) Cycle Examples

system then initiates a carry test to determine whether a carry occurred from the high-order position of the B-field. The presence of a carry indicates that the result in the B-field is a true figure (Figure B-4). The original sign of the B-field is the sign of the result.

If there was no carry from the high-order position of the B-field, the result in the B-field is not a true figure. A recomplement cycle is performed to convert the result to a true figure. In an add operation that results in a negative figure, the sign of the result is always changed during a recomplement cycle, (Figure B-5). The system generates the new sign automatically. A positive factor is indicated by the presence of an A- and B-bit over the units position of the factor. After a complement-add cycle, the sign of the result carries the sign of the field with the larger magnitude.

An accumulator field positioned in the last few available core-storage locations, such as units position of the field in 3999, will cause a wrap-around error, if an arithmetic function involving recomplementation is performed.

Arithmetic Instructions

Add (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
A	A	AAA	BBB

Function. The data in the A-field is added algebraically to the data in the B-field. The result is stored in the B-field.

Word Marks. The B-field must have a defining word mark, because it is this word mark that actually stops the add operation.

The A-field must have a word mark, only if it is shorter than the B-field. In this case, the transmission of data from the A-field stops after the A-field word mark is sensed. Zeros are then inserted in the A-register until the B-field word mark is sensed.

If the A-field is longer than the B-field, the high-order positions of the A-field that exceed the limits imposed by the B-field word mark are not processed. For overflow conditions and considerations, assume that the A-field is the same length as the B-field. (See *Address Modification.*)

Timing.

1. If the operation does not require a recomplement cycle:

   $T = .0111 (L_I + 1 + L_A + L_B)$ ms.

2. If a recomplement cycle is taken:

   $T = .0111 (L_I + 1 + L_A + 3L_B)$ ms.

If the multiply-divide special feature is installed, the 1440 timing for a recomplement cycle is:

$T = .0111 (L_I + 1 + L_A + 2L_B)$ ms.

Notes.

1. *Sign control* (see Figure B-2):

   If a recomplement cycle is taken, the sign of the B- (result) field is changed and the result is stored in true form.

2. *Zone bits:*

   If the fields to be added contain zone bits in other than the high-order position of the B-field and the sign positions of both fields, only the digits are used in a true-add operation. B-field zone bits are removed except for the units and high-order positions in a true-add operation. If a complement add takes place, zone bits are removed from all but the units positions of the B-field.

3. *Overflow indication:*

   If an overflow occurs during a true-add operation, the overflow indicator is set ON, and the overflow indications are stored over the high-order digit of the B-field. When the A-field exceeds, or is equal to, the B-field length, and the A-field position that corresponds to the high-order B-field position contains a zone bit, this zone bit is added to any zone bits present in the high-order B-field position.

Condition	*Result*
First overflow	A-bit
Second overflow	B-bit
Third overflow	A- and B-bits
Fourth overflow	No A- or B-bits

For subsequent overflows repeat conditions 1 through 4. Overflow indication does not occur for a 1-position field.

The BRANCH IF ARITHMETIC OVERFLOW INDICATOR ON, B (III) Z, instruction tests and turns off the overflow indicator, and branches to an instruction or group of instructions if an overflow condition occurred. There is only one overflow indicator in the system. It is turned off either by executing a BRANCH IF ARITHMETIC OVERFLOW INDICATOR ON instruction or pressing the start reset key on the 1447 operator panel.

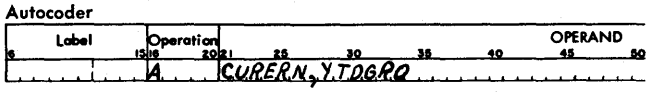
Overflow indication does not occur for a 1-position field.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_W	B-L_B

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Add CURERN (0506) to YTDGRO (0708), (Figure B-6).



Assembled Instruction: A 506 708

Figure B-6. Add (Two Fields)

Add (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
A	A	AAA

Function. This format of the ADD instruction causes the data in the A-field to be added to itself.

Word Marks. The A-field must have a defining word mark. It is this word mark that stops the add operation.

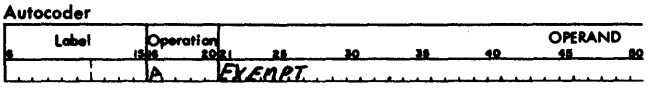
Timing. $T = .0111\ (L_I + 1 + 2L_A)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$A\text{-}L_A$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Add to itself the data at EXEMPT (0981), (Figure B-7).



Assembled Instruction: A 981

Figure B-7. Add (One Field)

Subtract (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
S	S	AAA	BBB

Function. The numerical data in the A-field is subtracted algebraically from the numerical data in the B-field. The result is stored in the B-field. Refer to Figure B- 2 for the sign that results from a specific subtract operation.

Word Marks. A word mark is required to define the B-field. An A-field requires a word mark, only if it is shorter than the B-field. In this case, the A-field word mark stops transmission of data from the A-field.

Timing.

1. If the operation does not require a recomplement cycle:
   $T = .0111\ (L_I + 1 + L_A + L_B)$ ms.
2. Subtract – recomplement cycle necessary:
   $T = .0111\ (L_I + 1 + L_A + 3L_B)$ ms.

If the multiply-divide special feature is installed, the 1440 timing for a recomplement cycle is:

$T = .0111\ (L_I + 1 + L_A + 2L_B)$ ms.

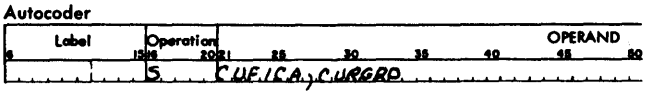
Note. If a recomplement cycle is taken, the sign of the B-(result) field is changed, and the result is stored in true form.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_W$	$B\text{-}L_B$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Subtract CUFICA (00753) from CURGRO (0896); Figure B-8.



Assembled Instruction: S 753 896

Figure B-8. Subtract (Two Fields)

Subtract (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
S	S	AAA

Function. The data at the A-address is subtracted from itself. If the A-field sign is minus, the result is a minus zero. If the A-field sign is plus, the result is a plus zero.

Word Marks. The A-field must have a defining word mark.

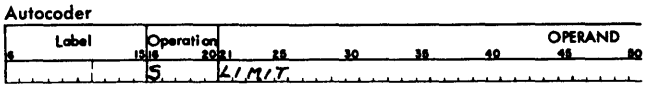
Timing. $T = .0111\ (L_I + 1 + 2L_A)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$A\text{-}L_A$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Subtract from itself the field labeled LIMIT (units position is 0395); Figure B-9.



Assembled Instruction: S 395

Figure B-9. Subtract (One Field)

Zero and Add (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
ZA	?	AAA	BBB

Function. This instruction functionally adds the A-field to a *zeroed* B-field. Technically, this is accomplished by moving the A-field to the B-field. The high-order positions of the B-field are set to zero if the B-field is larger than the A-field. The data from the A-field moves directly from the A-register to storage. Zone bits are stripped from all positions except the units position where the resultant sign will be represented in standard form. Blanks in the A-field are stored as blanks in the B-field.

Word Marks. A word mark is required for definition of the B-field. It is required in the A-field, only if it is shorter than the B-field. If the A-field is shorter than the B-field, all extra high-order B-field positions contain zeros. But the transmission of data from A stops when the A-field word mark is detected.

Timing. $T = .0111\ (L_I + 1 + L_A + L_B)$ ms.

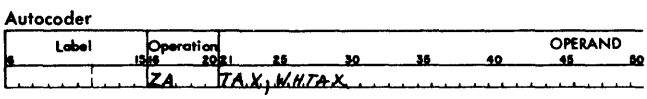
Note. The sign of the result always has both A- and B-bits if it is positive. If the sign is negative, it has only a B-bit.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_W$	$B\text{-}L_B$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Zero WHTAX area (0796-0802) and add new TAX (0749-0754) to WHTAX (Figure B-10).



Assembled Instruction: ? 754 802

Figure B-10. Zero and Add (Two Fields)

Zero and Add (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
ZA	?	AAA

Function. This format of the ZERO AND ADD instruction is used to strip the A-field of all zone bits, except in the units (sign) position. The A-field sign is retained. If the A-field plus sign bit configuration is not an A- and B-bit, it is changed to the A- and B-bit configuration.

Word Marks. The A-field must have a word mark in its high-order position.

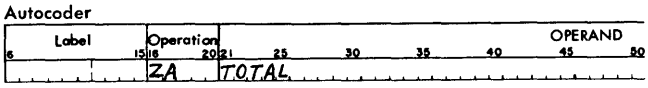
Timing. $T = .0111\ (L_I + 1 + 2L_A)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$A\text{-}L_A$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Strip zone bits from TOTAL (0560) area (Figure B-11).



Assembled Instruction: ? 560

Figure B-11. Zero and Add (One Field)

Zero and Subtract (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
ZS	!	AAA	BBB

Function. This instruction functionally subtracts the A-field from a *zeroed* B-field. Technically, this is accomplished by moving the A-field to the B-field. The high-order positions of the B-field are set to zero if the B-field is moved directly from the A-register to the B-field. Zone bits are stripped from all but the sign (units) position. The sign is represented in standard form. Blanks in the A-field are stored as blanks in the B-field.

Word Marks. A word mark is required to define the B-field. If the A-field is shorter than the B-field, the A-field must have a defining word mark to stop transmission of data to B. The extra high-order B-field positions contain zeros, if A is shorter than B.

Timing. T = .0111 ($L_I + 1 + L_A + L_B$) ms.

Note. If the A-field is positive, the B-field result is negative. If the A-field is negative, the B-field result is positive.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_W	B-L_B

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Zero ACCUM1 (0755) and subtract TAXEXP (0699) from ACCUM1; Figure B-12.

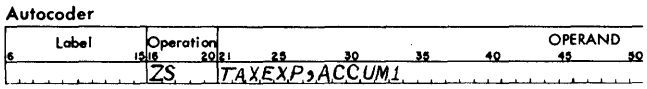


Figure B-12. Zero and Subtract (Two Fields)

Zero and Subtract (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
ZS	!	AAA

Function. This instruction changes the A-field sign, and strips all A-field zone bits, except in the units (sign) position.

Word Marks. The data in the A-field requires a word mark in its high-order position.

Timing. T = .0111 ($L_I + 1 + 2L_A$) ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_A	A-L_A

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Subtract LIMIT (0495) from zero, and change sign of LIMIT's value (Figure B-13).

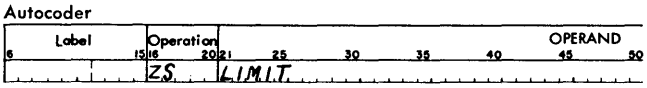


Figure B-13. Zero and Subtract (One Field)

Logic Operations

The 1440 program can test for certain conditions that may arise during processing, and can transfer the program to a predetermined set of instructions or subroutines, as a result of these specific tests. The operations that perform these testing operations are called logic operations.

For example, if an overflow occurs in an arithmetic operation, a routine to handle this conditon can be initiated by executing a BRANCH IF ARITHMETIC OVERFLOW INDICATOR ON instruction. Branching to this routine is called a *conditional* branch. The sequential execution of program steps is bypassed, and the program branches to the address of the instruction specified by the I-address of this conditional branch instruction. If the condition had not been present, the system would have started reading the instruction that appears at the immediate right of the conditional branch instruction (next sequential instruction). All conditional branch instructions have a d-character that is used to specify the conditions necessary for a program transfer.

A branch that occurs as a direct result of the execution of the instruction itself is called an *unconditional branch.* No special condition (other than the execution of the program step) is needed to transfer the program out of its normal sequential execution.

Any branch operation that terminates with a successful branch to another portion of core storage for the next instruction address operates as follows:

- The B-address register is reset to blanks during the next instruction operation (I-op) cycle.
- If the indexing and store address register special feature is installed on the system, the next sequential instruction (NSI) is placed in the B-address register and during the following instruction the B-address register is not set to blanks. This stored address can then be operated on by the store B-register feature to facilitate re-entry into the main program after the subroutine to which the branch occurred is completed.

Logic Instructions

Branch (Unconditional)

Instruction Format.

Mnemonic	*Op Code*	*I-address*
B	B	III

Function. This instruction always causes the program to branch to the address specified by the I-address position of the instruction. This address contains the op code of some instruction.

This unconditional branch operation is used to interrupt normal program sequence, and to continue the program at some other desired point, without testing for specific conditions.

Word Marks. The instruction is executed correctly if the core-storage position next to the I-address units position contains either a blank or a word mark.

Timing.

Branch (without indexing): T = .0555 ms.
Branch (with indexing): T = .0666 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Unconditionally branch to AGAIN (3498); Figure B-14.

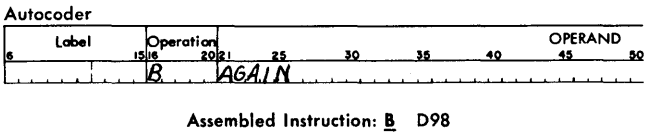


Figure B-14. Branch (Unconditional)

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
See Figure B-15	B	III	d

Function. The d-character specifies the indicator tested. If the indicator is on, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken. Figure B-15 shows commonly used valid d-characters, the indicators they test, and the conditions that turn the indicators off. Refer to the *Appendix* for full listing.

Word Marks. Word marks are not affected.

Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

MNEMONIC	d-CHARACTER	BRANCH ON	RESET BY
BC9 BCV	9 @	Carriage Channel # 9 Carriage Channel # 12	Branch Test or Channel 1 Punch
BPB	P	Printer Busy	Machine Circuitry
BIN +	‡	Printer Error with I/O Check Stop Switch Off	Branch Test
BLC	A	"Last Card" Switch (sense switch A) First Card Read Punch	Manual System Operator (Switch) or next card feed cycle
BLC 2	&	"Last Card" Switch (sense switch A) Second Card Read Punch	
BIN + BIN +	? !	Read Error Punch Error } If I/O Check Stop Switch is Off	Branch Test
BSS + BSS + BSS + BSS + BSS + BSS +	B C D E F G	Sense Switch B Sense Switch C Sense Switch D Sense Switch E Sense Switch F Sense Switch G	System Operator
BAV BIN +	Z %	Arithmetic Overflow Processing Check with Check Stop Switch Off	Branch Test
BIN + BIN + BIN + BIN + BIN + BIN +	N \ (Left Oblique) V W X Y	Access Inoperable Access Busy Disk Read or Write Error Wrong-Length Record Unequal-Address Compare Any-Disk Condition	Next Disk Storage Operation
BU BE BL BH	/ (Diagonal) S T U	Unequal Compare (B≠A) Equal Compare (B=A) Low Compare (B<A) High Compare (B>A)	Next Compare or Disk Storage Operation
BEF BER	K L	End of Reel Tape Error	Branch Test

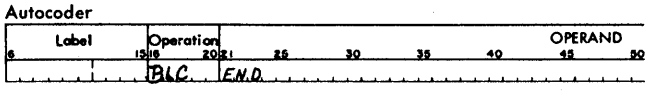
+ d-Character must be coded in the operand portion of the instruction.

Figure B-15. Frequently Used Conditional Branch Instructions (See *Appendix* for Complete List)

Address Registers After Operation. All d-characters.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test for last card. If it is the last card, branch to END (0599); Figure B-16.



Assembled Instruction: B 599 A

Figure B-16. Branch If Indicator On

Branch if Character Equal

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*B-address*	*d-character*
BCE	B	III	BBB	d

Function. This instruction causes the single character at the B-address to be compared to the d-character. If the comparison is equal, the program branches to the I-address for the next instruction. If the two characters are not the same, the program continues with the next sequential instruction.

Word Marks. Word marks in the location tested have no effect on the operation.

Timing.

No Branch: T = .0111 (L_I + 2) ms.
Branch (without indexing): T = .0111 (L_I + 2) ms.
Branch (with indexing): T = .0111 (L_I + 3) ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	B-1
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. This example shows how the chaining method can be used to test an entire field for blank characters. Each position in the area labeled AMOUNT (0350, 0349, 0348 and 0347) is individually tested for a blank character. If a blank is found, the program branches to BLANK (0601) for the next instruction. If the position tested contains a character, the program continues in sequence (Figure B-17).

Autocoder

Label	Operation	OPERAND
	BCE	BLANK,AMOUNT,2
	BCE	
	BCE	
	BCE	

Assembled Instruction: B 601 350 bl
B
B
B

Figure B-17. Branch If Character Equal

Branch if Word Mark or Zone

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*B-address*	*d-character*
See Figure B-18	V	III	BBB	d

Function. This instruction examines the character located at the B-address for the zone or word-mark combinations specified by the d-character. A correct comparison branches the program to the specified I-address. If the program does not branch to the I-address, it continues with the next sequential instruction. The d-characters, the associated mnemonics, and the conditions they test are shown in Figure B-18.

Word Marks. These have been explained previously.

MNEMONIC	d-CHARACTER	CONDITION
BW	1	Word mark
BWZ	2	No zone (No A, No B-bit)
BWZ	3	Either a word mark, or no zone
BWZ	B	12 zone (A B-bits)
BWZ	C	Either a word mark, or 12 zone
BWZ	K	11 zone (B, No A-bit)
BWZ	L	Either a word mark, or 11 zone
BWZ	S	Zero zone (A, No B-bit)
BWZ	T	Either a word mark, or zero zone

Figure B-18. Branch If Word Mark and/or Zone Mnemonics, d-Characters and Conditions

Timing.

No Branch: T = .0111 (L_I + 2) ms.
Branch (without indexing): T = .0111 (L_I + 2) ms.
Branch (with indexing): T = .0111 (L_I + 3) ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	B-1
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Test the units position of GROAMT (2498) for an 11-zone, and branch to NEGRTE (0598) for the next instruction. If there is no 11-zone, continue the program sequence (Figure B-19).

Autocoder

Label	Operation	OPERAND
	BWZ	NEGRTE,GROAMT,K

Assembled Instruction: V 598 M98 K

Figure B-19. Branch If Word Mark and/or Zone

Compare

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
C	C	AAA	BBB

Function. The characters in the B-field are compared to an equal number of characters in the A-field. The

comparison turns on an indicator that can be tested by a subsequent BRANCH IF INDICATOR ON instruction. The indicator is reset by either the next COMPARE instruction or the next disk-storage operation.

The same indicators set by the COMPARE instruction are also affected by a disk operation (seek, read, write, and write check). The disk-storage drive performs an address-compare operation automatically on the address in core storage, with the address on the disk record, by using the compare circuits and by setting the appropriate indicator (equal, high, or low). Therefore, careful consideration must be made in the use of a COMPARE instruction and subsequent BRANCH IF INDICATOR ON instructions for testing the results of the COMPARE instruction when disk operations are to be performed.

Word Marks. The first word mark encountered stops the operation. If the A-field is longer than the B-field, extra A-field positions at the left of the B-field word mark are not compared. If the B-field is longer than the A-field, an unequal-compare results. In this case, the high-compare indicator is set ON.

Timing. $T = .0111\ (L_I + 1 + 2L_W)$ ms.

Note. Both fields must have exactly the same bit configurations to be equal. For example, 00? ($? = \bar{0}$) compared to 00! ($! = \bar{0}$) results in unequal comparison.

All characters that can appear in storage can be compared. The ascending sequence of characters is:

blank ' □ [< ǂ & $ *] ; Δ − / , % ϒ \ ǂ ϕ # @ : > √ ? A through I ! J through R ǂ S through Z 0 through 9.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_W	B-L_W

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code. When compare operations are chained, the compare-result indicators are set by the first unequal condition encountered in the composite field.

Example. Compare the department numbers punched in two cards. Department numbers are located in:

Card	*Label*	*Actual Address*
1	DEPTNO	1098
2	DEPTCD	0004

Then test the results of the compare operation. If the department numbers are equal, continue the program in sequence. If they are unequal, branch to TOTAL (0495) for the next instruction (Figure B-20).

Autocoder

Label	Operation	OPERAND
	C	DEPTCD,DEPTNO
	BU	TOTAL

Assembled Instruction: C 004 ǂ98
B 495 /

Figure B-20. Compare

Data-Moving Operations

The 1440 data-moving operations are used to manipulate data within core storage during processing. Depending on the specific operation, one character, a group of characters, or a part of one character can be involved in the operation. A move operation does not affect word marks, but a load operation causes word marks as well as data to be transferred.

Data-Moving Instructions

Move Characters to A or B Word Mark (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-Address*	*B-Address*
MLC	M	AAA	BBB

Function. The data in the A-field is moved to the B-field.

Word Marks. If both fields are the same length, only one of the fields must have a defining word mark. The first word mark encountered stops the operation. If the word mark is sensed in the A-field, the machine takes one more B-cycle to move the high-order character from A to B. At the end of the operation, the A-address register and the B-address register contain the addresses of the storage locations immediately to the left of the A- and B-fields processed by the instruction. The data at the A-address is unaffected by the move operation. Word marks in both fields are undisturbed.

Timing. $T = .0111\ (L_I + 1 + 2L_W)$ ms.

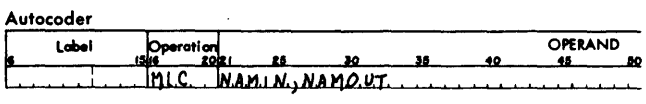
Note. If the fields are unequal in length, chaining can produce unwanted results, because one of the fields has not been completely processed. Thus, one of the registers will not contain the address of the units position of the left-adjacent field.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_W	B-L_W

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code, or the operation code and the A-address.

Example. Move the 5-character field NAMIN (0750) to the 5-character field NAMOUT (0850), Figure B-21.



Assembled Instruction: M 750 850

Figure B-21. Move Characters to A or B Word Mark (Two Fields)

Move Characters to A or B Word Mark (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
MLC	M	AAA

Function. This format of the move operation can be used when it is desired to move fields from the A-area and store them sequentially in the B-area. It saves program storage space and time, because the B-address is automatically taken from the B-address register, and does not have to be written or interpreted as part of the instruction.

Word Marks. A word mark is required in the high-order position of the A- or B-field. The first word mark encountered stops the move operation.

Timing. $T = .0111\ (L_I + 1 + 2L_W)$ ms.

Address Registers After Operation.

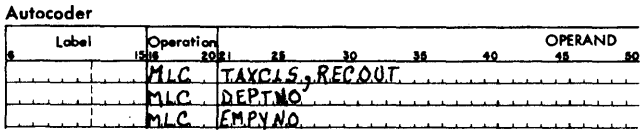
I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_W	Bp-L_W

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code, or the operation code and the A-address.

Example. Move the following three fields (labeled EMPNO, DEPTNO and TAXCLS) and store them sequentially at RECOUT (units position at 0204), Figure B-22.

Note: If the B-address register already contains the correct address, the B-label of the first instruction in the example can be eliminated.

	A-label	*A-actual address*	*B-label*	*B-actual address*
Employee number	EMPYNO	0101-0104		0201-0204
Department	DEPTNO	0108-0110		0205-0207
Tax Class	TAXCLS	0114-0115	RECOUT	0208-0209



Assembled Instruction: M 115 209
M 110
M 104

Figure B-22. Move Characters to A or B Word Mark (One Field)

Move Characters and Suppress Zeros

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MCS	Z	AAA	BBB

Example	Op Code	A–address	B–address
Move Char. and Suppress Zeros	Z	xxx	xxx
Storage before		A–field (data)	B–field (data)
		± 001206	bbbbbbb
Storage after		± 001206	bbb1206

Figure B-23. Move Characters and Suppress Zeros Operation Example

Function. The data in the A-field is moved to the B-field. After the move, high-order zeros and commas are replaced by blanks in the B-field. Any character that is not a comma, hyphen, blank, significant digit, or zero causes zero suppression to begin again. The sign is removed from the units position of the data field. Refer to Figure B-23 for a move characters and suppress zeros operation example.

Figure B-24 is another example of a move characters and suppress zeros operation involving a multiple field transfer. In this operation there are effectively two groups of high-order zeros. The @ sign is recognized as not being a significant digit or a zero, blank, comma, decimal, or minus sign. Thus, not only are the two high-order zeros suppressed, but also the two zeros to the right of the @ sign.

Word Marks. The A-field word mark stops transmission of data. B-field word marks, encountered during the move operation, are erased.

Timing. $T = .0111 (L_I + 1 + 3L_A)$ ms.

Note. This description of the instruction assumes a 1440 system without the expanded print edit special feature. If the feature is installed, a decimal does not restart zero suppression. (See *Special Features* section.)

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$B + 1$

Chaining. This instruction is not normally chained.

Example	Op Code	A–address	B–address
Move Char. and Suppress Zeros	Z	xxx	xxx
Storage before		A–field (data)	B–field (data)
		± 0010b @ 00.25	bbbbbbbbbbb
Storage after		± 0010b @ 00.25	bbb10b @ bb.25

Figure B-24. Move Characters and Suppress Zeros Operation Example

Example. Move and suppress the zeros in the 10-character field labeled INVBAL (0958) to the area labeled OUTPT4 (0448), Figure B-25.

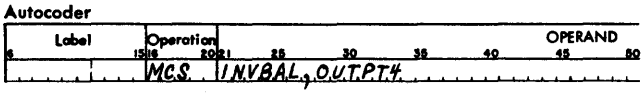


Figure B-25. Move Characters and Suppress Zeros

Move Characters to Record Mark or Group Mark with a Word Mark

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MRCM	P	AAA	BBB

Function. This instruction makes it possible to move an entire record from one core-storage area to another, regardless of the presence of word marks in either field. The A- and B-addresses specify the high-order position of the respective areas. Transmission starts from the high-order addresses, and continues until a record mark (A82 bits) or a group mark with a word mark (CBA8421WM bits) is sensed in the A-field. The record mark or group mark transfers to the B-field.

Word Marks. Word marks within the area do not affect the operation. Any word marks in the B-field remain unchanged. A-field word marks are not transmitted to the B-field.

Timing. $T = .0111 (L_I + 1 + 2L_A)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A + L_A$ (The length of the A-field includes the group mark with a word mark or record mark)	$B + L_A$

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Move the disk record that has its high-order character in the location labeled TARCIN (0679) to another area of core storage beginning at the label WTAREC (0985), Figure B-26.

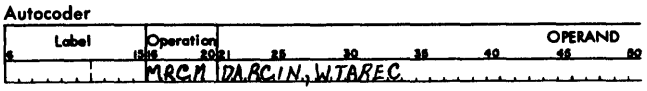


Figure B-26. Move Characters to Record Mark or Group Mark with a Word Mark

Move Numeric

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MLNS	D	AAA	BBB

Function. The numerical portion (8-4-2-1 bits) of the single character in the A-address is moved to the B-address. The zone portions (AB bits) are undisturbed at both addresses The entire character in the A-address is left undisturbed.

Word Marks. Word marks are not required at either address, because the nature of the instruction always specifies that only one digit is to be transmitted.

Timing. $T = .0111\ (L_I + 3)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	B-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Move the numerical portion of the units position of ONHAND (0986) to OUT5 (0789), Figure B-27.

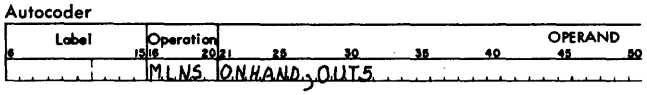


Figure B-27. Move Numeric

Move Zone

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MLZS	Y	AAA	BBB

Function. Only the zone portion (AB bits) is moved from the A-address to the B-address. The digit portions (8-4-2-1 bits) are undisturbed at both addresses. The entire character in the A-address is left undisturbed.

Word Marks. Word marks are not required at either the A- or B-addresses, because this instruction involves a single character.

Timing. $T = .0111\ (L_I + 3)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	B-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Move the zone bits from the units position of NEWBAL (3100) to the area labeled REC2 (3195); Figure B-28.

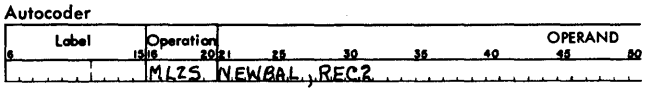


Figure B-28. Move Zone

Load Characters to A Word Mark (Two Fields)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MLCWA	L	AAA	BBB

Function. This instruction is commonly used to load data into designated printer or punch output areas of storage, and also to transfer data or instructions from a designated read-in area to another storage area. The data and word mark from the A-field are transferred to the B-field, and all other word marks in the B-field are cleared.

Word Marks. The A-field must have a defining word mark, because the A-field word mark stops the operation.

Timing. $T = .0111\ (L_I + 1 + 2L_A)$ ms.

Note: If the B-field is larger than the A-field, the B-field word mark is not cleared.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$B\text{-}L_A$

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code, or the operation code and the A-address.

Example. Transfer the data and word marks from REC4 (0950) to OUT8 (0650); Figure B-29.

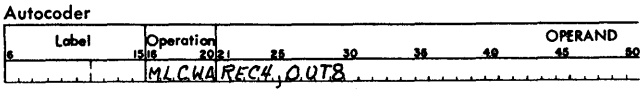


Figure B-29. Load Characters to A Word Mark

Load Characters to A Word Mark (One Field)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
MLCWA	L	AAA

Function. This format can be used when several A-fields (not necessarily in sequence) are to be loaded sequentially in the B-field. This instruction causes the A-field data and word mark to be moved to the B-field. B-field word marks are cleared, up to the A-field word mark.

Word Marks. The A-field word mark stops the operation. Therefore, B-field word marks, beyond the left limit of the A-field, are not cleared.

Timing. $T = .0111\ (L_I + 1 + 2L_A)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	$A\text{-}L_A$	$Bp\text{-}L_A$

Chaining. This instruction can be chained to the preceding instruction (if that instruction left usable address-register contents) by supplying only the operation code, or the operation code and the A-address.

Example. Load the three fields, EMPYNO, DEPTNO, and TAXCLS, with their word marks to sequential locations, beginning at storage location (0201), Figure B-30.

	A-label	*A-actual address*	*B-label*	*B-actual address*
Employee number	EMPYNO	0101-0104		0201-0104
Department	DEPTNO	0108-0110		0205-0207
Tax Class	TAXCLS	0114-0115	PRINT1	0208-0209

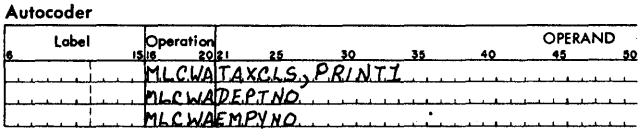


Figure B-30. Load Character to A Word Mark (One Field)

Miscellaneous Operations

The miscellaneous operations in an IBM 1440 Data Processing System involve the insertion and removal of word marks from specific core-storage locations, the clearing of core-storage areas, programmed halt operations, and other similar operations.

Miscellaneous Instructions

Set Word Mark (Two Addresses)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
SW	,	AAA	BBB

Function. A word mark is set at each address specified in the instruction. The data at each address is undisturbed. A word mark cannot be set in core-storage position 000.

Word Marks. Word marks are set at both the A- and B-addresses specified. A word mark is not required in the core-storage position following this instruction.

Timing. T = .0111 (L_I + 3) ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	B-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Set word marks at locations BEGIN1 (3950) and BEGIN2 (3970); Figure B-31.

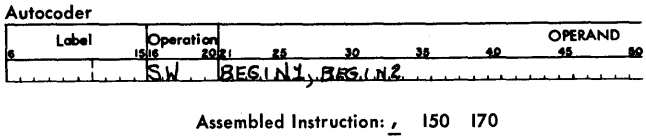


Figure B-31. Set Word Mark (Two Addresses)

Set Word Mark (One Address)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
SW	,	AAA

Function. This format of the SET WORD MARK instruction causes a word mark to be set at the A-address.

Data at this address is undisturbed. A word mark cannot be set in core-storage position 000.

Word Marks. A word mark is set at the A-address.

Timing. T = .0111 (L_I + 3) ms

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	A-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Set a word mark at AREA2 (2901); Figure B-32.

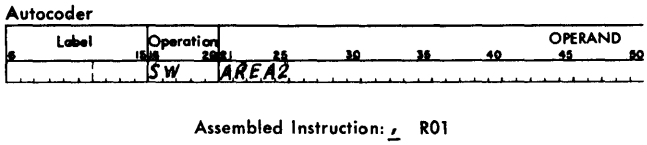


Figure B-32. Set Word Mark (One Address)

Clear Word Mark (Two Addresses)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
CW	□	AAA	BBB

Function. This instruction clears word marks at the locations specified by the A- and B-addresses, without disturbing the data there. A process error occurs if the specified A- or B-address is core-storage position 000 (end-around check condition).

Word Marks. Word marks are cleared at the A- and B-addresses.

Timing. T = .0111 (L_I + 3) ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	B-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Clear the word marks at NETPAY (1924) and ACCUM4 (3309); Figure B-33.

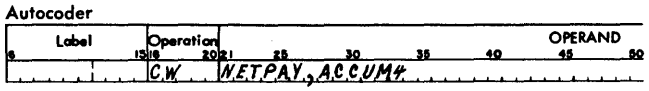


Figure B-33. Clear Word Mark (Two Addresses)

Clear Word Mark (One Address)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
CW	□	AAA

Function. This format of the CLEAR WORD MARK instruction causes the word mark to be cleared at the A-address. Data at the A-address is not disturbed. A process error occurs if the specified A-address is core-storage position 000 (end-around check condition).

Word Marks. Word marks are cleared at the A-address only.

Timing. $T = .0111\ (L_I + 3)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-1	A-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Clear the word mark at RECNO1 (3608); Figure B-34.

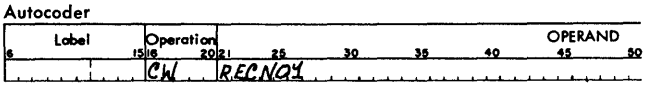


Figure B-34. Clear Word Mark (One Address)

Clear Storage

Instruction Format.

Mnemonic	*Op Code*	*A-address*
CS	/	AAA

Function. As many as 100 positions of core storage can be cleared of data and word marks when this instruction is executed. Clearing starts at the A-address and continues in descending address sequence to the nearest hundreds position. The cleared area is set to blanks (C-bits).

Word Marks. Word marks are also cleared, but do not stop the operation.

Timing. $T = .0111\ (L_I + 1 + L_X)$ ms.

Note: During the execution of this instruction, only the B-address register is used. Therefore, when chaining is being considered, the contents of the A-address register can be ignored.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A	x 00-1

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Clear WAREA5 (0500-0563); Figure B-35.

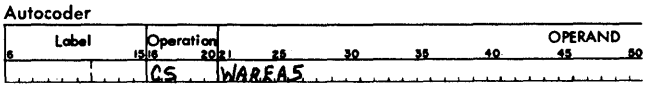


Figure B-35. Clear Storage

Clear Storage and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*B-address*
CS	/	III	BBB

Function. This is the same as the CLEAR STORAGE instruction, except that the clearing starts at the B-address. The I-address specifies the location of the next instruction.

Word Marks. Word marks do not stop the operation. It is not necessary to follow this instruction with a character and an associated word mark.

Timing.
Without indexing: $T = .0111\ (7 + L_X)$ ms.
With indexing: $T = .0111\ (8 + L_X)$ ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Without indexing	NSI	BI	blank
With indexing	NSI	BI	NSI

Example. Clear WAREA2 (0800-0898) and branch to START4 (0498) for the next instruction (Figure B-36).

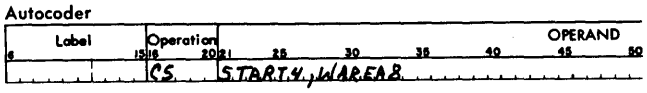


Figure B-36. Clear Storage and Branch

No Operation

Instruction Format.

Mnemonic	*Op Code*
NOP	N

Function. This code performs no operation. It can be substituted for the operation code of any instruction to make that instruction ineffective. It is commonly used in program modification to cause the machine to skip over specific instructions.

Instructions that have A-addresses of %xx or @xx should have their A-address field set to valid numeric values (all zeros, for example), or all N's with associated word marks to perform a no-operation function successfully. If this is not done, the A-address may contain characters that cause indexing and/or invalid core-storage addressing problems.

Word Marks. The program operation resumes at the next operation code identified by a word mark.

Timing. T = .0111 (L_I + 1) ms.

Note. If characters without word marks follow an N operation code, these characters enter the A- and B-field registers. For example:

N 1234 A xxxx

In this instance, the address registers after operation would be:

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	123	4bb*

*If this address is subsequently used (chained or stored) an invalid-address check stop condition occurs. (See *Instruction Length* section.)

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A	B

Example. Leave eight storage positions open for an instruction core such as READ CARD M (000) (000) R. The correct instruction can be inserted when needed (Figure B-37).

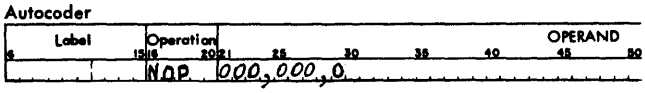


Figure B-37. No Operation

Halt

Instruction Format.

Mnemonic	*Op Code*
H	.

Function. This instruction causes the machine to stop and the stop-key light to turn ON. Pressing the start key causes the program to start at the next instruction in sequence.

Word Marks. Word marks are not affected.

Timing. T = .0222 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Ap	Bp

Example. Figure B-38 is a symbolic example of the HALT instruction.

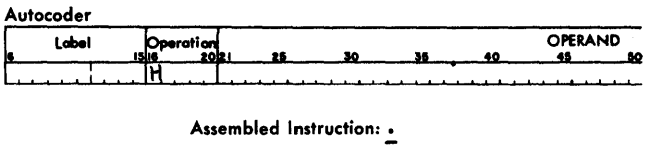


Figure B-38. Halt

Halt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*
H	.	III

Function. This is the same as HALT, except that the next instruction is at the I-address.

Word Marks. Word marks are not affected.

Timing.
Without indexing: T = .0555 ms.
With indexing: T = .0666 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Without indexing	NSI	BI	blank
With indexing	NSI	BI	NSI

Example. Stop the system, and branch to START2 (0895) for the next instruction when the start key is pressed (Figure B-39).

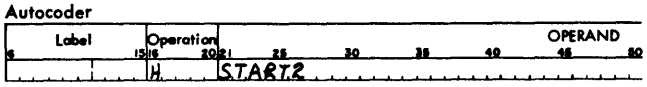


Figure B-39. Halt and Branch

Coded Halt

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*B-address*
H	.	C_1	
	.	C_1 C_2 C_3	C_4 C_5 C_6

Function. These forms of the HALT instruction place coded information in the A- and B-address and d-character positions. The coded information is then used to identify the halt. The coding used in these positions is left to the discretion of the programmer, but the system's valid addressing and indexing rules must be followed. The coding (specified by the user) can be entered by a DC statement following the HALT instruction:

```
H
DC  @J@
```
and will assemble as : . J

A 6-character DC statement will produce a halt with identification which fills both the A- and B-storage address registers.

Another method is to code a DCW statement where the coded halt is to appear in the program, in the form:

```
DCW  @•123456@
```
This then assembles as: . 123456

Word Marks. A word mark is required in the core-storage position adjacent to the instruction to specify the instruction length. (See *Instruction Length* section.)

Timing. T = .0111 (L_I + 1) ms.

Note. The last coded character also appears in the A-register.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	C_1 b b	C_1 b b
NSI	C_1 C_2 C_3	C_4 C_5 C_6

Example. Stop the system, and label the stop as 22 (Figure B-40).

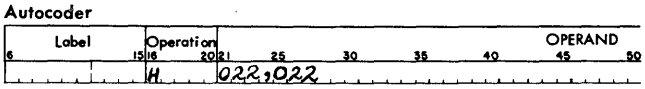


Figure B-40. Coded Halt

Edit Operation

The IBM 1440 Data Processing System has a powerful edit instruction that can cause all desired commas, decimals, dollar signs, asterisks, credit symbols, and minus signs to be inserted automatically in a numerical output field. Unwanted zeros to the left of significant digits can be suppressed. Thus, editing in the 1440 system is the automatic control of zero suppression, inserting of identifying symbols, and punctuation of an output field (Figure B-41).

In editing, two fields are needed: the data field and a control field. The data field is the data edited for output. The control field specifies how the data field is edited. It specifies the location of punctuation and condition of special characters and indicates where zero suppression occurs. The two fields are operated on character-by-character, under control of editing rules.

The control word has two parts: the *body* (which punctuates the A-field), and the *status* portion (which contains the dollar signs, sign-symbols, and class-of-total asterisks). The sign of the A-field determines whether or not sign symbols will print. The sign of the A-field is removed.

To edit a field, a LOAD CHARACTERS TO A WORD MARK instruction loads the control word into the specified printer output area. This puts the control word where the edited information will eventually go. Then, a MOVE CHARACTERS AND EDIT instruction (with the same B-address as the previous load instruction) performs the editing function as it moves the data into the output area.

Note: A 1-position field cannot be edited. Figure B-42 shows the use of these rules as applied to the data in Figure B-41.

Move Characters and Edit

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
MCE	E	AAA	BBB

Function. The data field (A-field) is modified by the contents of the edit control field (B-field), and the result is stored in the B-field. The data field and the control field are read from storage character-by-character, under control of the word marks and the editing rules. Any sign in the units position of the data field is removed during the operation.

	OP	A-address	B-address	
Edit instruction	E	789	300	
		A-field (data)	B-field (control word)	
Storage		00257426	$ bbb, bb0.bb & CR &**	
			B-field	
Result of edit		00257426	$ 2,574.26	**

Figure B-41. Editing Operation

Editing Rules

Rule 1. All numerical, alphabetic, and special characters can be used in the control word. However, some of these characters have special meanings:

Control Character	*Function*
b (blank)	This is replaced with the character from the corresponding position of the A-field.
0 (zero)	This is used for zero suppression, and is replaced with a corresponding character from the A-field. Also the right-most "0" in the control word indicates the right-most limit of zero suppression.
. (decimal)	This remains in the edited field in the position where written. It is removed during a zero-suppress operation if it is to the left of the high-order significant digit. When used with the expanded print edit feature, it has an additional function (see *Expanded Print Edit* section).
, (comma)	This remains in the edited field in the position where written. It is removed during a zero-suppress operation if it is to the left of the high-order significant digit.
CR (credit)	This is undisturbed if the data sign is negative. It is blanked out if the data sign is positive. It can be used in body of control word without being subject to sign control.
− (minus)	This is the same as CR.
& (ampersand)	This causes a space in the edited field. It can be used in multiples.
* (asterisk)	This can be used in singular or in multiples, usually to indicate class of total. When it is used with the expanded print edit feature, it takes on an additional function (see *Expanded Print Edit* section).
$ (dollar sign)	This is undisturbed in the position where it is written. When used with the expanded print edit feature, it has an additional function (see *Expanded Print Edit* section).

Rule 2. A word mark in the high-order position of the B-field controls the move characters and edit operation.

Rule 3. When the A-field word mark is sensed, the remaining commas in the control field are set to blanks. An A-field word mark is required for proper operation.

Rule 4. The body of the control word is that portion beginning with the right-most blank or zero, and continuing to the left to the control character that governs the transfer of the last position of the data field. The remaining portion of the control field is the *status* portion.

Cycle	TYPE OF CYCLE	ADDRESS REGISTERS			REG.		PUT BACK INTO STORAGE	"B" FIELD AT END OF CYCLE	REMARKS
		I	A	B	B	A			
1	I_{OP}	002	?	?	E		E	$bbb,bb0.bb&CR**	Read Instr. OP Code
2	I_1	003	07bb	07bb	7	7	7	same	Load A Address Register
3	I_2	004	078b	078b	8	8	8	same	Load A Address Register
4	I_3	005	0789	0789	9	9	9	same	Load A Address Register
5	I_4	006	0789	03bb	3	3	3	same	Load B Address Register
6	I_5	007	0789	030b	0	0	0	same	Load B Address Register
7	I_6	008	0789	0300	0	0	0	same	Load B Address Register
8	I_7	008	0789	0300	OP	0	OP	same	OP code of next instr.
9	A	008	0788	0300	6	6	6	same	Execute EDIT instr.
10	B	008	0788	0299	*	6	*	same	Rule 1
11	B	008	0788	0298	*	6	*	same	Rule 1
12	B	008	0788	0297	&	6	Blank	$bbb,bb0.bb&CRb**	Rule 1
13	B	008	0788	0296	R	6	Blank	$bbb,bb0.bb&Cbb**	Rule 1 and 5
14	B	008	0788	0295	C	6	Blank	$bbb,bb0.bb&bbb**	Rule 1 and 5
15	B	008	0788	0294	&	6	Blank	$bbb,bb0.bbbbbb**	Rule 1
16	B	008	0788	0293	b	6	6	$bbb,bb0.b6bbbb**	Rule 1
17	A	008	0787	0293	2	2	2	same	Rule 1
18	B	008	0787	0292	b	2	2	$bbb,bb0.26bbbb**	Rule 1
19	A	008	0786	0292	4	4	4	same	Rule 1
20	B	008	0786	0291	.	4	.	same	Rule 1
21	B	008	0786	0290	0	4	4	$bbb,bb4.26bbbb**	Zero Suppress—Rule1 and 7
22	A	008	0785	0290	7	7	7	same	Rule 1
23	B	008	0785	0289	b	7	7	$bbb,b74.26bbbb**	Rule 1
24	A	008	0784	0289	5	5	5	same	Rule 1
25	B	008	0784	0288	b	5	5	$bbb,574.2bbbbb**	Rule 1
26	A	008	0783	0288	2	2	2	same	Rule 1
27	B	008	0783	0287	,	2	,	same	Rule 1
28	B	008	0783	0286	b	2	2	$bb2,574.26bbbb**	Rule 1
29	A	008	0782	0286	0	0	0	same	Rule 1
30	B	008	0782	0285	b	0	0	$b02,574.26bbbb**	Rule 1
31	A	008	0781	0285	0	0	0	same	Rule 1
32	B	008	0781	0284	b	0	0	$002,574.26bbbb**	Rule 1
33	B	008	0781	0284	$	0	$	$002,574.26bbbb**	Sense Word Mark—Rev. Scan—Rule 1 and 6
34	B	008	0781	0285	$	0	$	same	Rule 6
35	B	008	0781	0286	0	0	Blank	$b02,574.26bbbb**	Rule 6
36	B	008	0781	0287	0	0	Blank	$bb2,574.26bbbb**	Rule 6
37	B	008	0781	0288	2	0	2	same	Rule 6
38	B	008	0781	0289	,	0	,	same	Rule 6
39	B	008	0781	0290	5	0	5	same	Rule 6
40	B	008	0781	0291	7	0	7	same	Rule 6
41	B	008	0781	0292	4	0	4	$bb2,574.26bbbb**	Rule 6

Figure B-42. Step-by-step Editing Operation

Rule 5. If the data field is positive, and if the CR or – symbols are located in the status portion of the control word, they are blanked out.

Rule 6. The data field can contain fewer, but must not contain more positions than the number of blanks and zeros in the body of the control word. Dollar signs and asterisks are included in the body of the control word with the expanded print edit special feature.

Rule 7. Zero suppression is used if unwanted zeros to the left of significant digits in a data field are to be deleted (see Figure B-43).

Zero Suppression Operation

Zero suppression is the deletion of unwanted zeros at the left of significant digits in an output field (Figure B-43).

A special 0 is placed (in the body of the control word) in the right-most limit of zero suppression.

To perform zero-suppression operations properly, there must be at least one character to the left of the zero-suppression character in the control word.

Forward Scan:

1. The positions in the output field at the right of this special zero are replaced by the corresponding digits from the A-field.

A-field	0010900
Control word (B-field)	$ bb, bb0. bb
Forward scan	$ 00,109. 00
Reverse scan	$ bbb109. 00
Results of edit	$ 109. 00

Figure B-43. Zero Suppression

2. The special zero is replaced by the corresponding digit from the A-field, when it is detected in the control field.
3. A word mark is automatically set in this position of the B- (output) field.
4. The scan continues until the B-field (high order) word mark is sensed and removed.

Reverse Scan:

1. In the output field, blanks replace all zeros and punctuation, except hyphens at the left of the first significant character (up to, and including, the zero-suppression code position).
2. When the automatically-set zero suppression word mark is sensed, it is erased and the operation ends.

Timing. $T = .0111\ (L_I + 1 + L_A + L_B + L_Z)$ ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Without zero suppression	NSI	$A\text{-}L_A$	$B\text{-}L_B$
With zero suppression	NSI	$A\text{-}L_A$	Location of the special control zero plus 1.

Chaining. This instruction is not normally chained.

Example. Edit the data labeled GROPAY (0985) by the edit-control word EDCONT (0325). Store the result in PRINT6 (00250); Figure B-44.

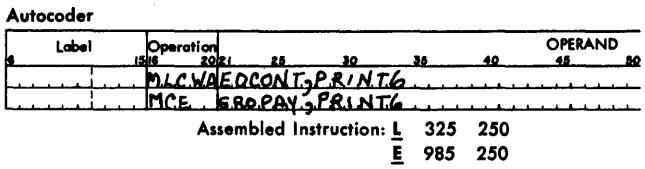


Figure B-44. Move Characters and Edit

The IBM 1447 Console (Model 1, 2, or 4), Figure C-1, is a required unit on an IBM 1440 Data Processing System. The console contains the system operating keys, lights and switches which give the operator external control for setting up and checking system operation. For more detail on the keys, lights, switches, and operating procedures, refer to IBM *1447 Console* (Form A24-3031).

Console Instruction Format

A program-initiated data transmission between the IBM 1447 Console (Model 2 or 4) and the attached system is started by executing the proper console instruction. If the data transmission is from the 1447 console to the system, a READ FROM 1447 CONSOLE instruction is executed. The format for the 1447 console is shown in Figure C-2.

The various parts of a 1447 console instruction and their uses are:

General Mode of Operation

This part of the instruction identifies the operation as either a move operation or a load operation. A move operation specifies that only the character coding is transmitted. A load operation specifies that both the character coding and any associated word marks are transmitted.

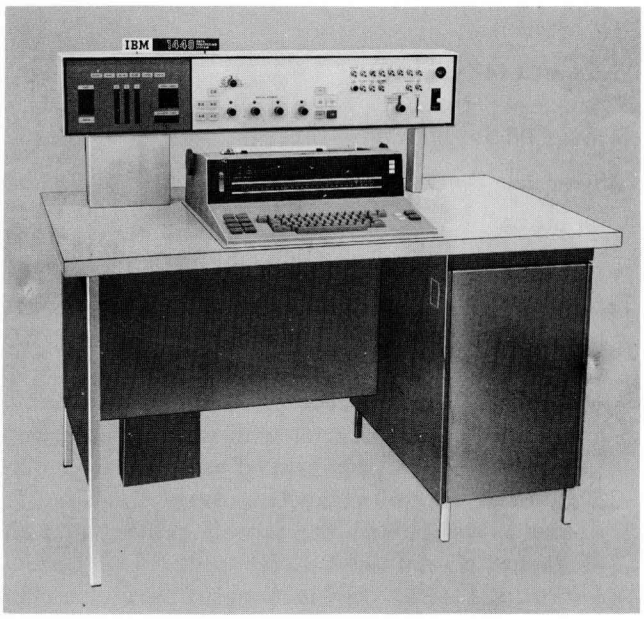


Figure C-1. IBM 1447 Console, Model 2

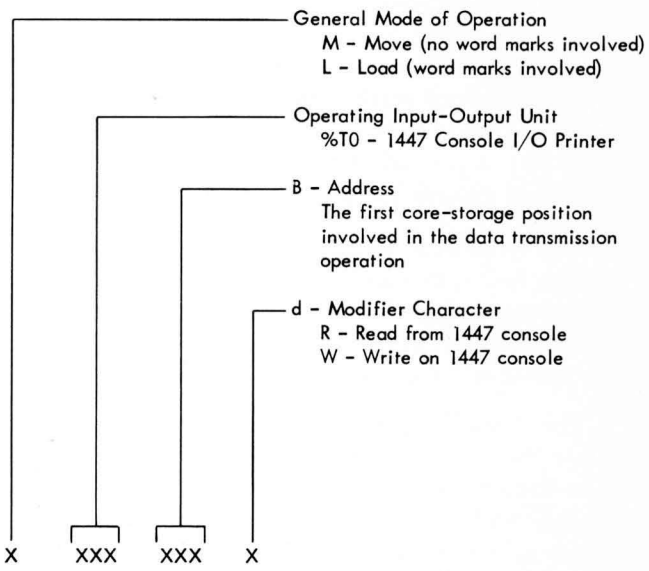


Figure C-2. IBM 1447 Console I/O Printer Instruction Format

Operating Input/Output Unit

This part of the instruction specifies the console I/O printer as the active input/output unit for this operation.

B-Address

This part of the instruction specifies the first leftmost core-storage position that will be involved in the operation.

d-Modifier Character

This part of the instruction specifies the data transmission direction. An R specifies a console printer-to-system data transmission; a W specifies a system-to-console printer data transmission.

IBM 1447 Console Instructions

Read from 1447 Console

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RCP	M	%T0	BBB	R
RCPW	L	%T0	BBB	R

Function. This instruction is used to enter data into core storage from the console I/O printer. The op code specifies the mode of operation. If the operation takes place in the *move* mode (M Op code), word marks cannot be transmitted from the console printer into core storage. Any word marks already in the area that accepts the message will remain there.

If the operation takes place in the *load* mode (L op code), word marks can be transmitted from the console printer into core storage when the word-mark key is pressed. Any word marks already in the area that accepts the message will be removed.

The A-address specifies the console I/O printer as the I/O unit involved in the operation. The B-address specifies the first core-storage position that accepts data from the console printer. The d-character specifies a console printer-to-system operation.

The console operator can start keying the data when the white type light on the console comes ON. The console operator types the data on the console printer and the characters enter core storage, beginning at the location specified by the B-address portion of the instruction.

The operator transmits a word mark by pressing the shift key and the word-mark key. The upper case (word-mark position) of the period key prints an inverted circumflex. The next character printed to enter core storage will have a word mark associated with it.

When the number of data positions to be entered into core storage exceeds the number of printing positions on one printer line, the print element automatically returns from the right-hand margin, executes a line feed in operation, and the keying operation continues on the next line.

The operation is normally ended when the operator presses the release key. This key operation inserts a group mark with a word mark in core storage, initiates a carrier-return and line-feed operation, and disconnects the printer from the system.

The operation can also be ended if a group mark with a word mark is sensed in core storage. This signifies that the input message exceeded the core-storage area capacity and:

1. The operation ends and the printer is disconnected from the system.
2. The inquiry clear (*) indicator in the system comes on.
3. The red type light on the console comes on.
4. A carrier-return and line-feed operation is initiated.
5. The keyboard is interlocked.

Word Marks. Depends on mode of operation. To end the operation correctly, a group mark with a word mark must be inserted into the 1440 core-storage position to the right of the position that contains the last character sent to the system from the console printer. If in load mode, existing word marks are erased, and new word marks are inserted in core-storage where applicable.

Timing. T = .0999 ms + operator keying time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%30	$B + L_B + 1$

Chaining. This instruction cannot be successfully chained.

Example. Transfer the data keyed on the console I/O printer to the area in 1440 core storage labeled INQIN (0785); Figure C-3.

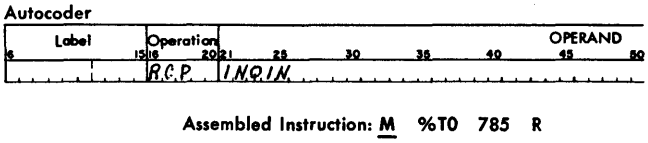


Figure C-3. Read from 1447 Console

Write on 1447 Console

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WCP	M	%T0	BBB	W
WCPW	L	%T0	BBB	W

Function. This instruction is used to transfer data from core storage to the console I/O printer. The Op code specifies the mode of operation. If the operation takes place in the move mode, word marks are ignored. The character with an associated word mark in core storage is printed as a character only. Functional control characters cause the specified carrier movement on the console printer, and the characters do not print. Refer to IBM *1447 Console* (Form A24-3031) for functional control characters and associated printer operation.

If the operation takes place in the load mode, the word marks are transmitted and printed. The word mark is printed before the associated character is printed. Functional control characters are also printed. The carrier movement normally specified by the character does not occur.

The A-address specifies the console I/O printer as the I/O unit involved in the operation and turns on the white type light if the printer is available for use. The B-address specifies the first core-storage position of the area that contains the data to be printed. The d-character specifies a system-to-console printer operation.

The data reads out of core storage, beginning at the address specified in the instruction and continuing until a group mark with a word mark is encountered. The group mark with a word mark ends the operation, but does not print. A carrier-return operation, with an associated line-feed operation, occurs and the system advances to the next instruction.

If the end of a printed line is reached before the group mark with a word mark is sensed, printing is suspended and a carrier-return and line-feed operation is executed. When the carrier reaches the left-hand margin, the print-out operation continues.

Word Marks. Depends on mode of operation. Characters in core storage which have associated word marks are preceded on the print-out by an inverted circumflex, when the console-write operation is in load mode. When the console-write is in move mode, word marks are not indicated on the printed output. A group mark with a word mark in core storage ends the operation.

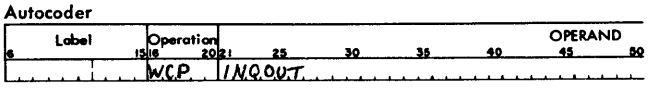
Timing. T = .0999 + 68 (L_B) + 800 (number of carrier return operations) ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%30	B + L_B + 1

Chaining. This instruction cannot be successfully chained.

Example. Print out the data, beginning in the area labeled INQOUT (0785) and ending with a group-mark with a word-mark (Figure C-4).



Assembled Instruction: M %T0 785 W

Figure C-4. Write on 1447 Console

Line Feed Operation

The 1447 console printer forms can be spaced up with the WRITE ON 1447 CONSOLE instruction by addressing a core-storage position containing a group mark with a word mark. See *Write on 1447 Console* section.

Branch if Inquiry Request

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	Q

Function. This branch indicator usually signifies that the console I/O printer has a message to send to the system. The indicator turns on during a console-inquiry operation when the operator presses the type key.

Note: Refer to *IBM 1447 Console,* Form A24-3031, for special feature considerations and additional information concerning this instruction.

Word Marks. Word marks are not affected.

Timing.

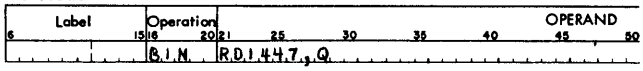
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	Qbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the inquiry-request indicator and branch to a subroutine labeled RD1447 (4766) if the test is positive (Figure C-5).



Assembled Instruction: B 76W Q

Figure C-5. Branch if Inquiry-Request Indicator On

Branch if Inquiry Clear

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	*

Function. When special features are not involved, this branch indicator and the associated red type light are turned on when:

1. The console operator makes a keying mistake and instructs the system to disregard the message (by pressing the cancel key).
2. The input message exceeds the core-storage area capacity.
3. The 1447 circuitry detects a parity error during the 1447-to-core-storage transfer.
4. The 1447 circuitry detects a parity error during the core-storage-to-1447 transfer.

Note: Refer to *IBM 1447 Console,* Form A24-3031, for special feature considerations and additional information concerning this instructon.

Word Marks. Word marks are not affected.

Timing.

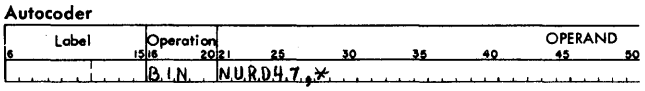
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	*bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the inquiry-clear indicator and branch to a subroutine labeled NURD47 (6531) if the test is positive (Figure C-6).



Assembled Instruction: B N3 / *

Figure C-6. Branch If Inquiry-Clear Indicator On

Console Printer Timing

The console I/O printer is used for input to, and output from, the IBM 1440 Data Processing System.

The timing involved during an input operation is: T = .0111 (L_I + 1) + console operator keying time.

The timing involved during an output operation is: T = .0111 (L_I + 1) + 68 (L_B) + 800 (number of carrier return operations) ms.*

* Only one portion of either operation is overlapped by processing. This is the last carrier-return and line-feed operation that occurs at the end of an output operation. (See IBM *1447 Console,* Form A24-3031.)

IBM 1403 Printer, Models 2, 3, 5, and 6

The printing power of the 1440 system spans the range from 120 lines per minute with the 63-character set on the 1443 Model 3, up to 1400 lines per minute with the 1403, Model 3, with the preferred character set.

IBM 1403 Printer Operations

This section describes the instructions the 1440 uses to control the IBM 1403 Printer (Figure D-1). The basic unit timings and other general information are included.

The various models of the IBM 1403 Printer give the 1440 system greater printed-output speed than can be achieved with the 1443 printer.

The number of lines that can be printed per minute depends on the 1403 model. Refer to *IBM 1403 Printer*, Form A24-3073.

Data Flow

When a WRITE LINE instruction is given, the data to be printed is read out of core storage to the print buffer.

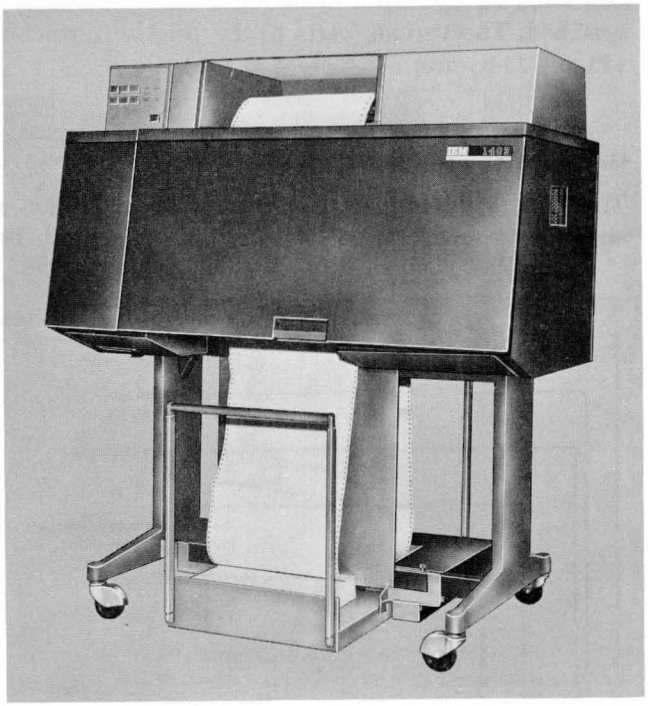


Figure D-1. IBM 1403 Printer

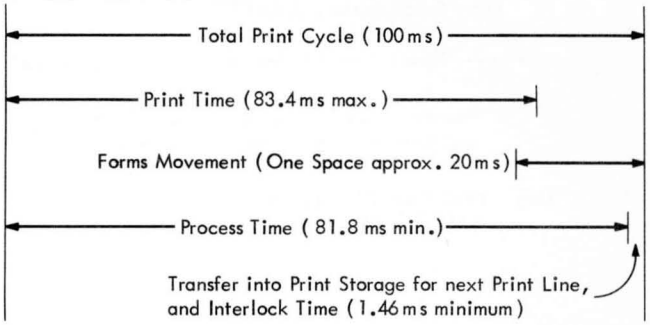


Figure D-2. Print Operation Timing Chart—IBM 1403, Model 2

The system is then released for other processing while the print buffer relays the data on to the printer.

IBM 1403 Printer, Model 2

The IBM 1403 Printer, Model 2, operates at a maximum rate of 600 lines per minute when the printer is impulsed to print.

The 100-millisecond print cycle begins as soon as the print request is made. It is subdivided into three parts (Figure D-2):

1. Print time requires 83.4 ms. The line prints during this part of the cycle. The system is not interlocked during this time because print storage is standard.
2. Process time is 81.8 ms. This is the normal processing time available during the print cycle.
3. Forms movement time is approximately 20 ms. The normal forms movement time (one space) is always overlapped by processing time.

IBM 1403 Printer, Model 3

The IBM 1403 Printer, Model 3, operates at a rate of 1100 lines per minute. The 54.5 millisecond print cycle is divided into three parts (Figure D-3):

1. Print time is 36.25 ms. The line prints during this part of the cycle. The system is not interlocked during this time because print storage is standard.
2. Process time is 53.5 ms. This is the normal processing time available during the print cycle.

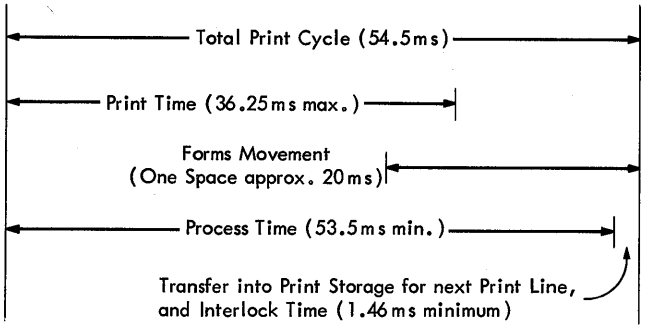


Figure D-3. Print Operation Timing Chart—IBM 1403, Model 3

3. Forms-movement time requires approximately 20 ms for a single space. The normal forms-movement time is always overlapped by processing time.

IBM 1403 Printer, Model 5

The IBM 1403 Printer, Model 5, is capable of printing 465 lines per minute. The 129 millisecond print cycle is subdivided into three parts (Figure D-4).

1. Print time is 110.3 ms. The line prints during this part of the cycle. The system is not interlocked during this time because print storage is standard.
2. Process time is 125.0 ms. This is the normal process time available during the print cycle.
3. Forms-movement requires approximately 20 ms. The normal forms-movement time (one space) is always overlapped with processing time.

IBM 1403 Printer, Model 6

The 1403 Printer, Model 6, has a maximum print-span of 120 positions. It can print 340 lines per minute, and has a single-speed carriage. The 176.5 millisecond print cycle is subdivided into three parts (Figure D-5):

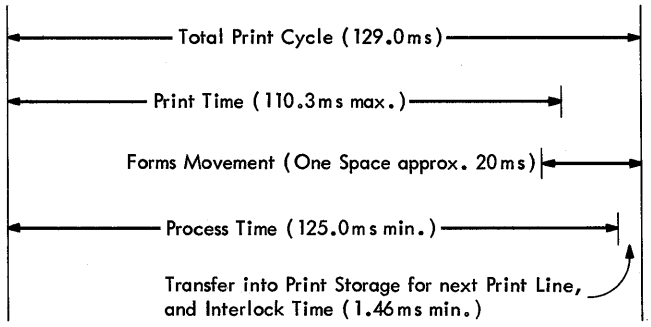


Figure D-4. Print Operation Timing Chart—IBM 1403, Model 5

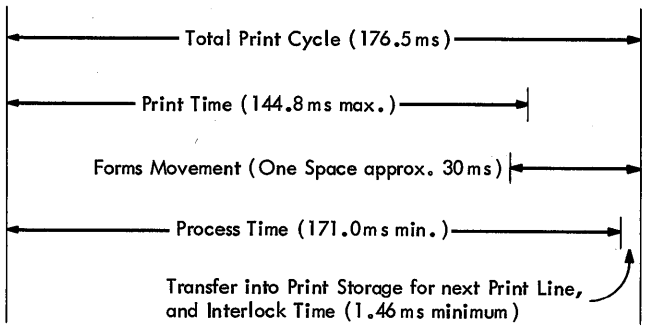


Figure D-5. Print Operation Timing Chart—IBM 1403, Model 6

1. Print time requires 144.8 ms. The line prints during this part of the cycle. The system is not interlocked during this time because print storage is standard.
2. Process time is 171.0 ms. This is the normal process time available during the print cycle.
3. Forms-movement time requries approximately 20 ms. The normal forms-movement time (one space) is always overlapped by process time.

IBM 1403 Printer Instruction Format

All printing operations are initiated by either one of two types of printer instructions. If the instruction is two characters long, an operation involving the printer carriage is specified. If the instruction is eight characters long, an operation involving a write operation is specified. The various parts of the printer instruction (Figure D-6) are:

General Mode of Operation

This part of the instruction identifies the operation as either a write operation or a carriage operation. Be-

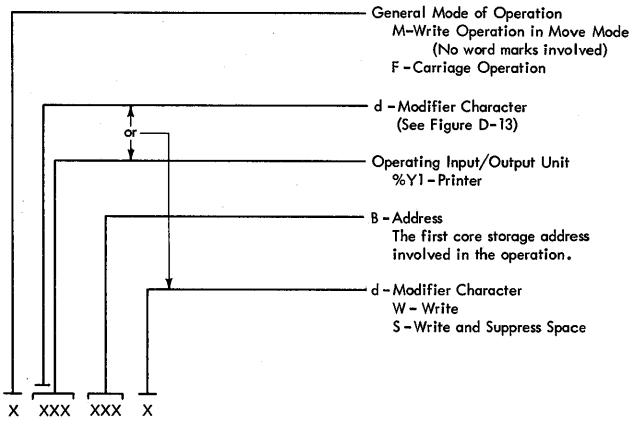


Figure D-6. IBM 1403 Printer Instruction Format

cause the write operation is performed in the move mode, word marks in the specified core-storage area are not affected during the operation.

d-Modifier Character

If the instruction is two characters long, the second character is the d-modifier character. This character specifies the type of carriage operation to occur. Refer to Figure D-13 for a list of the d-characters and the carriage operations they initiate.

Operating Input/Output Unit

This part of the instruction specifies the printer as the active unit for this operation.

B-Address

This part of the instruction specifies the first core-storage position to be involved in the operation.

d-Modifier Character

This part of the instruction specifies the type of write operation to be performed in the printer.

IBM 1403 Printer Instructions

Instructions applying to the 1403 cannot be successfully chained.

Write Line

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
W	M	%Y1	BBB	W

Function. This instruction is used to transfer data from core storage, through the print buffer to the 1403 printer.

The high-order position of data in the core-storage position specified by the B-address is printed in print-position 1. The rest of the data located in the adjacent core-storage positions is printed in the adjacent print positions until a group mark with a word mark in core storage is sensed, or until the print span of the 1403 is satisfied.

The number of characters printed depends on the B-field length established in core storage and on the model of the attached 1403. One position past the last character to be printed contains a group mark with a word mark if the print line is less than the maximum possible for the model of 1403 installed. If the group mark with a word mark is omitted, the print span of the printer determines the number of characters printed. An address validity check occurs if the B-address specifies the last 100-character block of core storage. The B-field length can be from 1 to 132 (Models 2, 3, and 5) or 1 to 120 (Model 6).

An automatic single space operation occurs after the actual printing ends unless a different carriage operation is programmed.

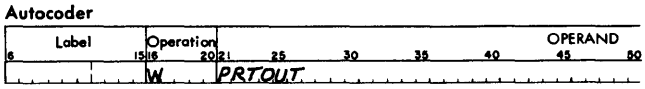
Word Marks. Word marks are not affected.

Timing. T = .0999 ms + I/O

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + 132 (for 132-position printer)

Example. Print the data beginning in the area labeled PRTOUT (0101) and ending with a group mark with a word mark (Figure D-7).



Assembled Instruction: M %Y1 101 W

Figure D-7. Write Line

Write Line and Suppress Space

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WS	M	%Y1	BBB	S

Function. This instruction is used to transfer data from core storage, through the print buffer to the 1403 printer. The automatic single space, normally taken after printing, is suppressed.

The high-order position of data in the core-storage position specified by the B-address is printed in

print-position 1. The rest of the data located in the adjacent core-storage positions is printed in the adjacent print positions until a group mark with a word mark in core storage is sensed, or until the print span of the 1403 is satisfied.

The number of characters printed depends on the B-field length established in core storage and on the model of the attached 1403. One position past the last character to be printed contains a group mark with a word mark if the print-line is less than the maximum possible for the model of 1403 installed. If the group mark with a word mark is omitted, the print span of the printer determines the number of characters printed. An address validity check occurs if the B-address specifies the last 100-character block of core storage. The B-field length can be from 1 to 132 (Models 2, 3, and 5) or 1 to 120 (Model 6).

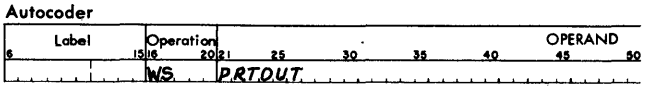
Word Marks. Word marks are not affected.

Timing. T = .0999 ms + I/O.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + 132 (if 132-position printer)

Example. Print the data beginning in the area labeled PRTOUT (0101) and ending with a group mark with a word mark, and suppress the automatic single space (Figure D-8).



Assembled Instruction: M %Y1 101 S

Figure D-8. Write Line and Suppress Space

Branch if Printer Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	≠

Function. If an error occurs during a printer operation, this indicator is set ON, and the printer light glows on the console. This indicator can be tested to effect a branch. If the indicator is on, it is reset. The next instruction to be executed is at the location specified by the I-address of the BRANCH IF INDICATOR ON instruction.

Note: This error indicator applies to any 1403, 1443, or 1445 attached to the system, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing:

T = .0666 ms.

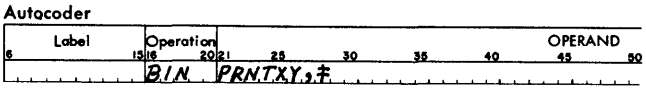
Branch with indexing:

T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	‡bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the printer-error indicator to determine if an error occurred on the immediately preceding print instruction. If an error occurred, branch to the instruction labeled PRNTXY (0661). If no error occurred, continue processing with execution of NSI (Figure D-9).



Assembled Instruction: B 661 ‡

Figure D-9. Branch If Printer Error

Branch if Printer Busy

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BPB	B	III	P

Function. If the printer or printer carriage is currently occupied with another operation, this indicator can be tested to effect a branch to another series of instruction. The indicator is reset as soon as the printer is available for another operation. Using this instruction allows processing to continue while the printer is busy, thus, in effect, allowing temporary overlapping of processing and printer operation.

Note 1: This indicator is effective for any 1403, 1443, or 1445 attached to the system.

Note 2: The *Branch if Printer Busy* and *Branch if Printer Carriage Busy* instructions should be included in any program where one print and space/skip operation has not been completed before the next print and space operation is initiated. If these instructions are not used, incorrect carriage spacing or skipping can result. The following sample program segment illustrates how these instructions might be coded:

```
PRBUSY    BPBB     PRBUSY,
PCBUSY    BPCB     PCBUSY
SPACE     CC       S
WRITE     W        NET PAY
          NSI
```

Frequently, there are other processing steps that can be performed while waiting for the carriage to complete the last print/space/skip function. The *Branch if Printer Busy* and *Branch if Printer Carriage Busy* instructions could have branched to perform these other functions.

This precaution applies only to systems having buffered printers.

Word Marks. Word marks are not affected.

Timing.

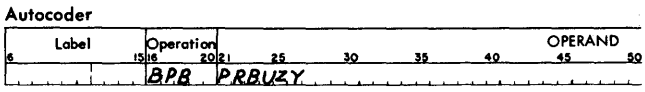
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Pbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer-busy indicator to determine if the printer is occupied with some operation. If it is, branch to another series of instructions beginning at the instruction labeled PRBUZY (0486) while waiting for the printer to become available. If the printer is *not* busy, the NSI is executed (Figure D-10).



Assembled Instruction: B 486 P

Figure D-10. Branch If Printer Busy

Branch if Printer Carriage Busy (1403 Only)

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BPCB	B	III	R

Function. If the printer carriage is executing a forms-movement operation, this instruction can be used to branch to other instructions until the carriage is again available. The indicator is reset as soon as the printer carriage is no longer busy. Using this instruction allows processing to continue while the printer carriage is busy, thus, in effect, permitting temporary overlapping of processing and printer operations.

Note: See Note 2 under Branch if Printer Busy section.

Word Marks. Word marks are not affected.

Timing.

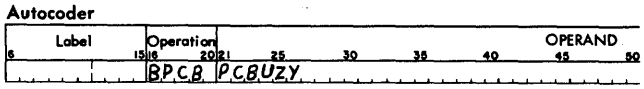
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Rbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer carriage-busy indicator to determine if the carriage is occupied with some operation. If it is, branch to another series of instruction beginning at the instruction labeled PCBUZY (9444) while waiting for the printer to become available. If the printer carriage is *not* busy, the NSI is executed (Figure D-11).



Assembled Instruction: B U4M R

Figure D-11. Branch If Printer Carriage Busy

Branch if Channel 9

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BC9	B	III	9

Function. If the channel-9 position of the carriage control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	9bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test to determine if the carriage control tape (forms) channel 9 has been sensed. Branch to a subroutine called BC9YES (0784) if the test is positive. (Figure D-12).

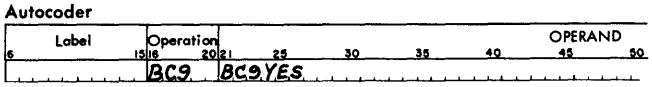


Figure D-12. Branch If Printer Carriage Channel 9

Branch if Channel 12

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-Character*
BCV	B	III	@

Function. If the channel 12 (forms overflow) position of the carriage control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	@bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to a subroutine labeled OVFLHD (0659) if the hole in channel 12 of the carriage-control tape has been sensed, indicating that the present form is filled (Figure D-13).

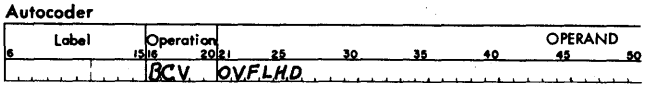


Figure D-13. Branch If Printer Carriage Forms Overflow

Control Carriage

Instruction Format.

Mnemonic	*Op Code*	*d-character*
CC	F	d

Function. This instruction causes the carriage to move as specified by the d-character. If the d-character is:

1. a digit, an immediate skip to the specified channel in the carriage tape occurs.
2. an alphabetic character containing a 12-zone, a skip to the specified channel in the carriage tape occurs after the next line is printed.
3. an alphabetic character containing an 11-zone, an immediate space operation, as specified by the digit portion of the character, occurs.
4. an alphabetic character containing a zero-zone, a space operation, as specified by the digit portion of the character, occurs after the next line is printed.

Refer to Figure D-14 for a list of the d-characters and the carriage operations they specify. If the carriage is already in motion when another CONTROL CARRIAGE instruction is given, the stored program execution is suspended until the carriage operation being performed is completed. At that time, the carriage action specified by the instruction begins, and the program advances to the next instruction.

Note: There is no CARRIAGE CONTROL AND BRANCH (CCB) instruction for the 1440 system.

Word Marks. Word marks are not affected.

d	Immediate skip to	d	Skip after print to
1	Channel 1	A	Channel 1
2	Channel 2	B	Channel 2
3	Channel 3	C	Channel 3
4	Channel 4	D	Channel 4
5	Channel 5	E	Channel 5
6	Channel 6	F	Channel 6
7	Channel 7	G	Channel 7
8	Channel 8	H	Channel 8
9	Channel 9	I	Channel 9
0	Channel 10	?	Channel 10
#	Channel 11	•	Channel 11
@	Channel 12	□	Channel 12
d	**Immediate space**	**d**	**After print-space**
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

Figure D-14. Control Carriage d-Characters

Timing. T = .0333 ms + remaining form-movement time, if carriage is already in motion when this instruction is given. The total form movement time depends on the specific carriage operation being performed. Refer to the *IBM 1403 Printer Timing* section for more detail. The form-movement time is determined by the number of spaces the form moves. Allow 20 ms for the first space, plus 5 ms for each additional space less than 8 and 2.5 ms for additional spaces over 8, for the 1403, models 2, 3, and 5. Allow 30 ms for the first space, plus 5 ms for each additional space for the 1403, Model 6 printer. (The 1403, Model 6, has a single-speed carriage drive mechanism).

Note: When an IMMEDIATE SKIP OR IMMEDIATE SPACE instruction is used, an additional space caused by the automatic carriage space is taken after printing results. When a SKIP AFTER PRINT OR SPACE AFTER PRINT instruction is used, the automatic space is ignored.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

Example. Skip to channel 1 after print operation (Figure D-15).

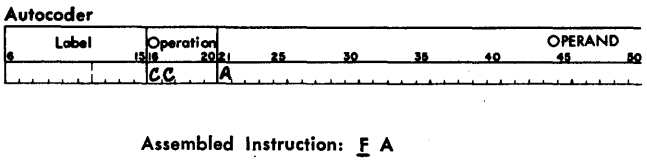


Figure D-15. Control Carriage

IBM 1403, Model 2, Timings

The IBM 1403 Printer, Model 2, operates at a rated speed of 600 lines per minute. The 100 ms print cycle (Figure D-16) allows a minimum of 97.3 ms for processing time because print storage is standard.

Forms movement for single spacing requires the last 20 ms of the print cycle. If additional forms-movement time is required by the application, this must be added to the 100 ms print cycle to determine the effective printing speed.

Lines Printed Per Minute	Length of Cycle (ms)	Available Process Time (ms)	Max. Spaces Skipped Within Cycle
600	100	97.3	1
572	105	102.3	2
545	110	107.3	3
522	115	112.3	4
500	120	117.3	5
480	125	122.3	6
462	130	127.3	7
444	135	132.3	8

Figure D-16. Effective Printing Speeds – IBM 1403, Model 2

Lines Skipped	Time Required (ms)
1	20
2	25
3	30
4	35
5	40
6	45
7	50
8	55

Figure D-17. Forms Movement Time – IBM 1403, Models 2, 3, and 5

Figure D-17 shows the effective printing speeds under various forms-movement considerations.

Additional forms skipping beyond the first 8 lines is calculated by multiplying the number of lines skipped by the 2.3 ms.

Some program instructions cause form movement to start immediately. If the printer is printing when an immediate forms control instruction is given, or if the carriage is already in motion, the system waits until the previous carriage operation is completed before the immediate skip is executed.

Immediate skips require 20 ms for the first space, 5 ms for each additional space up to 8, and then 2.3 ms for each space thereafter.

Figure D-17 shows form-movement timing requirements for immediate-skip instructions.

IBM 1403, Model 3, Timing

The IBM 1403 Printer, Model 3, operates at a rated speed of 1100 lines per minute. The 54.5 ms print cycle (see Figure D-3) allows 53.5 ms of processing time because print storage is standard.

Forms movement for single spacing requires the last 20 ms of the print cycle. If additional forms movement time is required by the application, this time must be added to the 54.5 ms cycle to determine printing speed.

Figure D-18 shows the effective printing speeds under various form-movement considerations.

Additional form-skipping time beyond the first 8 lines is calculated by multiplying the number of lines skipped by 2.3 ms.

Some program instructions cause form movement to start immediately. If the printer is printing when an immediate forms-control instruction is given, or if the carriage is already in motion, the 1440 waits until the previous carriage operation is completed before the immediate skip is executed.

Lines Printed Per Minute	Length of Cycle (ms)	Available Process Time (ms)	Max. Spaces Skipped Within Cycle
1100	54.5	53.1	1
1007	59.5	58.1	2
930	64.5	63.1	3
863	69.5	68.1	4
805	74.5	73.1	5
755	79.5	78.1	6
710	84.5	83.1	7
670	89.5	88.1	8

Figure D-18. Effective Printing Speeds – IBM 1403, Model 3

Immediate skips require 20 ms for the first space, 5 ms for each additional space up to 8, and then 2.3 ms for each space thereafter.

Refer to Figure D-17 for the form-movement timing requirements for immediate-skip instructions.

IBM 1403, Model 5, Timing

The IBM 1403 Printer, Model 5, operates at a rated speed of 465 lines per minute. The 129 ms print cycle (see Figure D-4) allows 125 ms of processing time because print storage is standard.

Forms movement for single spacing requires the last 20 ms of the print cycle. If additional forms-movement time is required by the application, time must be added to the 129 ms to determine actual printer speed.

Figure D-19 shows the effective printing speeds under various forms-movement considerations. Additional forms skipping time beyond the first 8 lines is calculated by multiplying the number of lines skipped by 2.3 ms.

Some program instructions cause form movement to start immediately. If the printer is printing when an immediate forms-control instruction is given, or if the carriage is already in motion, the system waits until the previous operation is complete before the immediate skip is executed.

Lines Printed Per Minute	Length of Cycle (ms)	Available Process Time (ms)	Max. Spaces Skipped Within Cycle
465	129.0	125.0	1
447	134.0	130.0	2
431	139.0	135.0	3
416	144.0	140.0	4
412	149.0	145.0	5
389	154.0	150.0	6
377	159.0	155.0	7
365	164.0	160.0	8

Figure D-19. Effective Printing Speeds – IBM 1403, Model 5

Immediate skips require 20 ms for the first space, 5 ms for each additional space up to and including 8, and 2.3 ms for each space thereafter.

Refer to Figure D-17 for forms-movement timing requirements for immediate-skip instructions.

IBM 1403, Model 6, Timing

The IBM 1403 Printer, Model 6, operates at a rated speed of 340 lines per minute. The 176.5 ms print cycle (see Figure D-5) allows 171 ms of processing time because print storage is standard.

Lines Printed Per Minute	Length of Cycle (ms)	Available Process Time (ms)	Max. Spaces Skipped Within Cycle
340	176.5	171.0	1
330	181.5	176.0	2
321	186.5	181.0	3
313	191.5	186.0	4
305	196.5	191.0	5
297	201.5	196.0	6
290	206.5	201.0	7
283	211.5	206.0	8

Figure D-20. Effective Printing Speeds – IBM 1403, Model 6

Forms movement for single spacing requires the last 30 ms of the print cycle. If additional forms movement is required by the application, time must be added to the 171 ms to determine actual printing speed. Figure D-20 shows the effective printing speeds under various forms-movement considerations.

Some program instructions cause form movement to start immediately. If the printer is already busy when an immediate forms-control instruction is given, the system waits until the previous operation is complete before the immediate skip is executed.

Immediate skips require 30 ms for the first space, plus 5 ms for any additional spaces.

Figure D-21 shows forms-movement timing requirements for immediate-skip instructions.

Lines Skipped	Time Required (ms)
1	30
2	35
3	40
4	45
5	50
6	55
7	60
8	65

Figure D-21. Forms-Movement Time – IBM 1403, Model 6

IBM 1442 Card Read-Punch

This section describes the instructions the IBM 1440 Data Processing System uses to control the IBM 1442 Card Read-Punch, and the IBM 1442, Model 4, Card Reader. The IBM 1442, Model 4, Card Reader operates under the control of the same read and stacker instructions as the IBM 1442 Card Read-Punch. (Refer to *IBM 1442 Card Read-Punch*, Form A24-3119.)

Data Flow

The card path and data flow for the IBM 1442 Card Read-Punch (Figure D-22) is shown in Figure D-23. The cards are placed in the 1,200-card capacity hopper face down, 9-edge first. The first card cycle moves the card from the hopper to the read station where it is registered at column zero. During the second card feed cycle, the card is fed to the reading station by a READ CARD instruction. This operation causes each card column to be read twice as the card moves by the reading station column-by-column.

During read cycle 1, the punched-card code for a column is translated to BCD code and stored in core-storage positions specified by the B-address of the READ CARD instruction. On read cycle 2, the punched-card code for the same column is read a second time. The resultant BCD-coded character from the second reading is compared to the BCD-coded character read into storage from the first read cycle. If no error is detected, the process continues for each column until a group mark with a word mark is detected at the end of the B-field.

After the read operation is completed, the card is registered in column 1 at the punch station.

During the third card-feed cycle, which is started by a PUNCH AND FEED instruction, the BCD-coded characters to be punched are read from core storage, translated to punched-card code, and punched column-by-column into the card at the punch station.

A second core-storage read-out cycle occurs that compares the BCD characters in storage to the BCD translation of the punched-card code punched in the card. If no error is detected, this operation continues for the length of the B-field in storage identified by a group mark with a word mark.

When the card leaves the punch station, it is carried to the stacker by a continuously-moving mechanism.

The data flow for the IBM 1442, Model 4, Card Reader is the same as the read operation on the IBM 1442 Card Read-Punch.

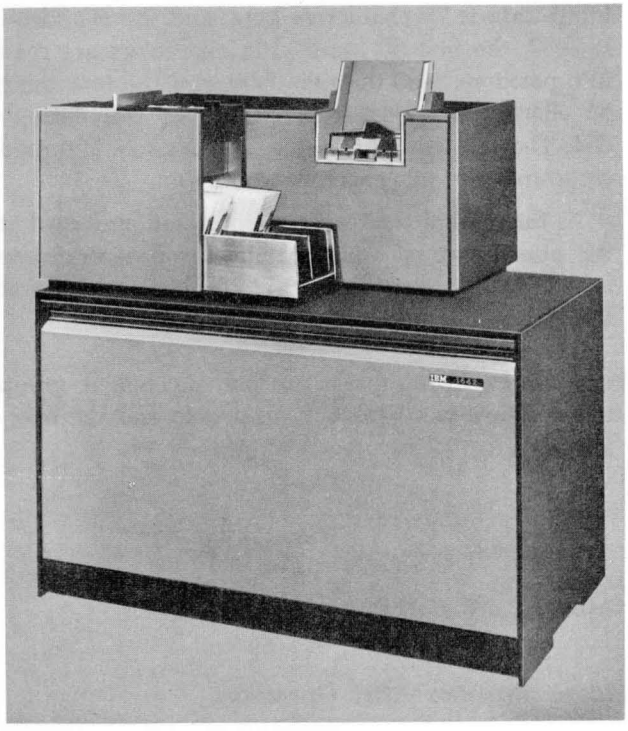


Figure D-22. IBM 1442 Card Read-Punch

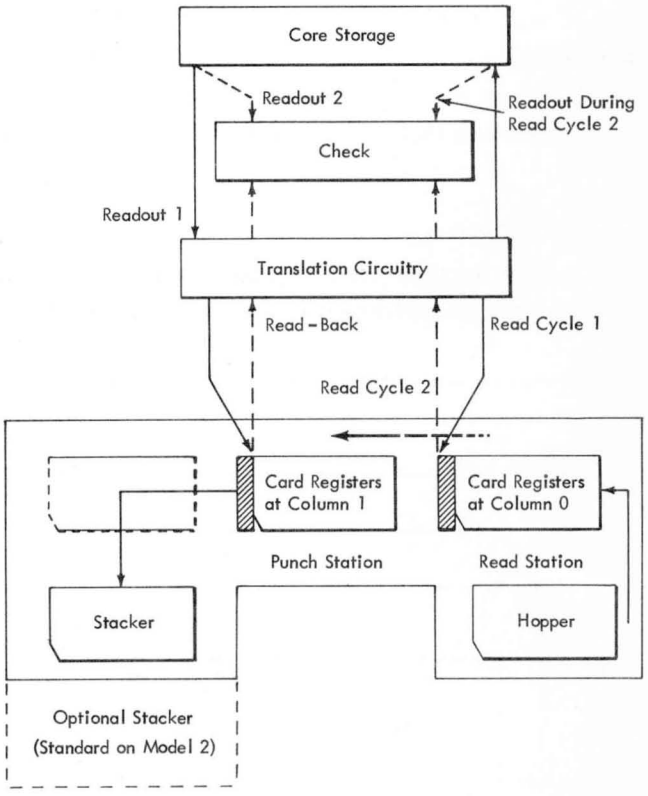


Figure D-23. IBM 1442 Card Read-Punch Data Flow

Card Read-Punch Instruction Format

All card read-punch operations are initiated by a CARD READ PUNCH instruction. This instruction can initiate different card read-punch operations by using specific characters in certain locations of the actual instruction (Figure D-24).

The various parts of the CARD READ PUNCH instruction and their uses are:

General Mode of Operation

This part of the instruction identifies the operation as a move operation. Word marks are not affected in the specified core-storage area during punching or reading operations.

Unit Number

This part of the instruction specifies which one of the operating units will be active when there is more than one card read-punch attached to the system. The first 1442, Model 1, 2, or 4 attached to the system has a unit-select number of 1. The second 1442, Model 1, 2, or 4, has a unit-select number of 2. Only one 1444 can be attached to the system. Its unit-select number is always 3.

B-Address

This part of the instruction specifies the first core-storage position that will be involved in the operation.

d-Modifier Character

This part of the instruction specifies the type of operation that will be performed in the card read-punch.

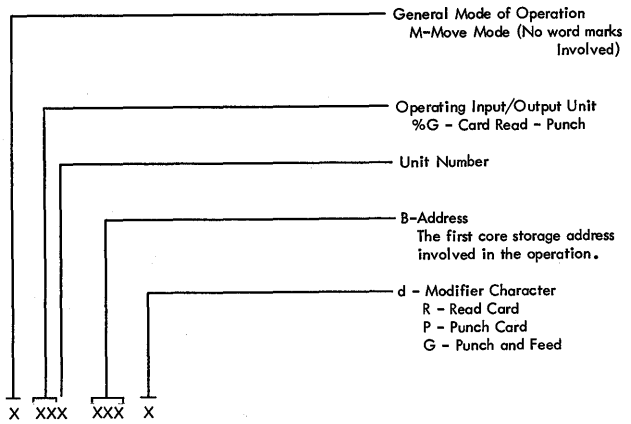


Figure D-24. IBM 1442 Card Read-Punch Instruction Format

IBM 1442 Card Read-Punch Instructions

Instructions applying to the 1442 cannot be successfully chained.

Read Card

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
R	M	%Gn	BBB	R

Function. This instruction is used to transfer data read at the card read-punch read station into a specified core-storage area.

The data punched in card column 1 is translated and stored in the core-storage position specified by the B-address. The rest of the data punched in the card is transferred, column-by-column, into the adjacent core-storage positions until a group mark with a word mark in core storage is sensed. The number of characters read from the card depends on the B-field length that is established in core storage. The B-field length can be from 1 to 80 positions, plus one position for the group mark with a word mark. (The system will *hang-up* in a read operation with the reader-ready light OFF, if the group mark with a word mark is missing.

An end-around check condition occurs when the data record length is longer than the number of core-storage positions from the B-address to the highest-numbered position in core storage. In a system of 4,000 storage positions, for example, if the input data is 75 characters long, and the B-address is 3980, the first 20 input data characters are read into positions 3980 through 3999, and the remaining 55 characters are read into positions 000 through 054. The storage light on the 1447 console is turned ON to indicate this check condition.

As the card at the read station is read, any card at the punch station is also being moved at the same speed, and is ejected into the number 1 stacker at the end of the read operation.

Word Marks. Word marks are not affected. A group mark with a word mark is needed to end the operation.

Timing.

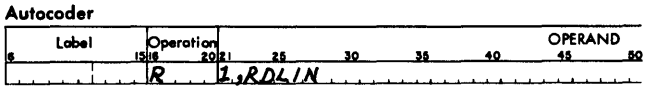
Model 1: $T = 25.0999 + (L_B + 1)$ ms.

Model 2: $T = 15.0999 + (L_B + 1)$ ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B + L_B + 1$

Example. Transfer the data read from card read-punch 1 to the area in core storage labeled RDLIN (0303), Figure D-25.



Assembled Instruction: M %G1 303 R

Figure D-25. Read Card

Punch and Stop

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
PS	M	%Gn	BBB	P

Function. This instruction is used to transfer data from core storage to the card read-punch where it is punched in a card.

The data in the core-storage position specified by the B-address is transferred and punched in the card column registered beneath the punching mechanism. The rest of the data located in the adjacent core-storage positions is transferred, column-by-column, and punched in the adjacent card columns until a group mark with a word mark in core storage is sensed. The number of characters punched in the card depends on the B-field length that is established in core storage. The B-field length can be from 1 to 80 positions, plus one position for the group mark with a word mark. (All characters in excess of 80 are punched in column 81 and lost.) When the punching operation ends, the card movement also ends. No other card movement takes place during a punch-and-stop operation.

Word Marks. Word marks are not affected. A group mark with a word mark is needed to end the operation.

Timing.

Model 1: $T = 6.3499 + 12.5\ (L_B)$ ms.

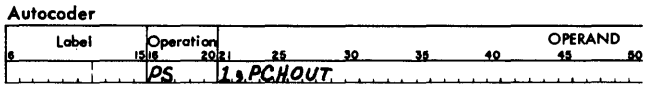
Model 2: $T = 3.2299 + 6.25\ (L_B)$ ms.

Note. When a punch-and-stop operation follows either a read-card or a punch-and-feed operation, the card at the punch station is registered in column 1, and punching begins in column 1. When a punch-and-stop operation follows another punch-and-stop operation, the card at the punch station is the card that was punched during a previous operation, and punching begins in the column adjacent to the last column previously punched.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B + L_B + 1$

Example. Punch the data on card read-punch 1, beginning in the area labeled PCHOUT (0303) and ending with a group mark with a word mark (Figure D-26).



Assembled Instruction: M %G1 303 P

Figure D-26. Punch and Stop

Punch and Feed

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
P	M	%Gn	BBB	G

Function. This instruction is used to transfer data from core storage to the card read-punch where it is punched in a card. When the punching operation ends, the card is ejected from the punch station and selected into a stacker.

The data stored in the core-storage position specified by the B-address is transferred and punched in the card column registered beneath the punching mechanism. The rest of the data located in the adjacent core-storage positions is transferred, column-by-column, and punched in the adjacent card columns until a group mark with a word mark in core storage is sensed. The number of characters punched in the card depends on the B-field length that is established in core storage. The B-field length can be from 1 to 80 positions, plus one position for the group mark with a word mark. (All characters in excess of 80 are punched in column 81 and lost.) When the punching operation ends, the card is ejected from the punch station and selected into a stacker.

The card located at the read station advances during this operation also, but the data in the card *is not* transferred into core storage. A card from the hopper is also advanced and registered at the read station during the punch-and-feed operation.

Word Marks. Word marks are not affected. A group mark with a word mark is needed to end the operation.

Timing.

Model 1: $T = 6.3499 + 12.5\ (L_B) + 210^*$ ms.

Model 2: $T = 3.2299 + 6.25\ (L_B) + 160$ ms.

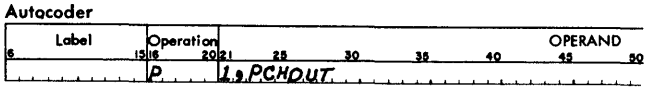
*When a PUNCH AND FEED instruction is initiated, a period of 210 ms elapses before another card read-punch operation can be executed.

Note: When a punch-and-feed operation follows either a read-card or a punch-and-feed operation, the card at the punch station is registered in column 1, and punching begins in column 1. When a punch-and-feed operation follows a punch-and-stop operation, the card at the punch station is the card that was punched during a previous operation, and punching begins in the column adjacent to the last column previously punched. The card is punched and stacked.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B + L_B + 1$

Example. Punch the data on card read-punch 1, beginning in the area labeled PCHOUT (0303) and ending with a group mark with a word mark, and then eject the card (Figure D-27).



Assembled Instruction: M %G1 303 G

Figure D-27. Punch and Feed

Select Stacker

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	2 or 0*

*NOTE: The d-character for the first card read-punch or card reader installed on the system is designated by 2, and the second card read-punch or card reader installed on the system is designated by 0.

This instruction must be issued prior to the PUNCH AND CO or READ CARD instruction that moves the card on through the feed.

Function. This instruction causes the card at the punch station to fall into stacker 2. Unless stacker 2 has been selected before the operation that ejects the card (read or punch feed), the ejected card is directed to stacker 1.

Note: The IBM 1444 and the IBM 1442, Models 2 and 4, have two stackers as standard equipment. A second stacker is provided on the IBM 1442, Model 1, as a special feature.

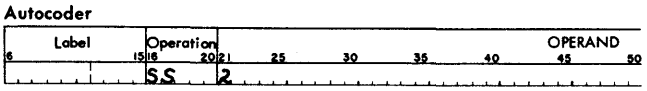
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	2bb or 0bb	2bb or 0bb

Example. Select the following card into pocket 2 (Figure D-28).



Assembled Instruction: K 2

Figure D-28. Select Stacker

Branch if Reader Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	? (plus zero)

Function. If an error occurs during the card-read operation, this indicator is set, and the reader light glows on the console. If the I/O check-stop switch is OFF, this indicator can be tested and reset by the BRANCH-IF INDICATOR ON instruction. If the indicator is ON, the program is interrupted, and a branch to the I-address occurs. If the indicator is not ON, no branch occurs.

Note: This error indicator operates for both the first and second 1442 card reader, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

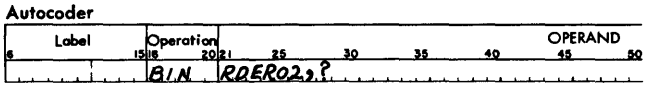
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	?bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the reader-error indicator to determine if an error occurred on the immediately preceding read operation. If an error occurred, branch to the instruction labeled RDER02 (0943). If no read error occurred, continue processing with execution of NSI, Figure D-29.



Assembled Instruction: B 943 ?

Figure D-29. Branch If Reader Error

Branch if Punch Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	! (minus zero)

Function. Same as for *Reader Error*, except that specifics apply to the punch operation.

Note: This error indicator is effective for the 1444, and first or second 1442 Model 1 or 2 punch operations, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

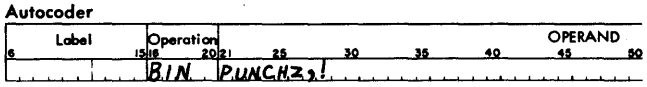
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	!bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to the instruction labeled PUNCHZ (3775) if a punch operation error occurred on any attached punch device. If no punch error occurred, proceed to the NSI (Figure D-30).



Assembled Instruction: B G75 !

Figure D-30. Branch If Punch Error

Branch if Last Card

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BLC	B	III	A (first read-punch)
BLC2	B	III	& (second read-punch)

Function. This instruction causes a branch to the address specified by the I-address, if the last card has *been* read and is *ready* for punching.

Word Marks. Word marks are not affected.

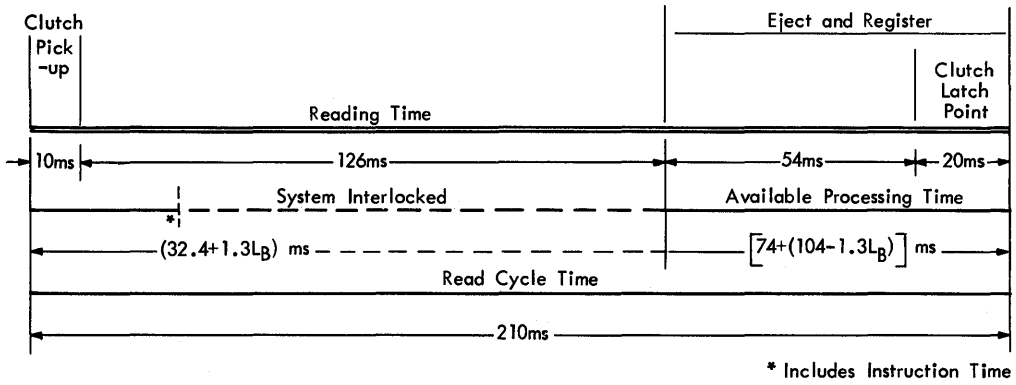


Figure D-32. Card Read Cycle—285 cpm (80 Card Columns)

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Abb or &bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. If the last card has been read and is ready for punching in the second 1442 card reader-punch, branch to a subroutine labeled LCTOT (0884). If the last card has not been read in the second card reader-punch, advance to the next sequential instruction (Figure D-31).

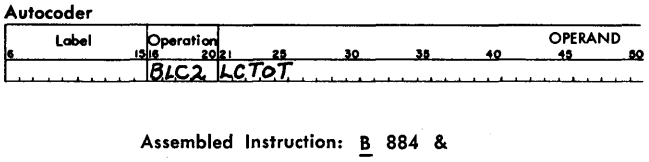


Figure D-31. Branch If Last Card, Second Card Read-Punch

IBM 1442 Card Read-Punch Timing

Model 1 Card Reading

Card reading on the IBM 1442, Model 1, can be done at either 285 or 300 cards per minute (cpm)

285 Cards per Minute

The 285-cpm cycle occurs when the next CARD READ instruction is given during the last 20 ms of a card-read cycle (Figure D-32). The clutch is allowed to latch up and the 10-ms clutch-pickup time must take place at the beginning of the next card-read cycle.

The time the system is interlocked during a read operation, when the clutch latches up after each operation, is shown in Figures D-32 and D-33. This timing formula can be consolidated as shown in Figure D-33. By using the consolidated formula, the total time available for other processing during one card-read cycle can be found by subtracting the consolidated formula from 210. The bottom portion of Figure D-33 shows the approximate processing times available by the number of card columns being read.

300 Cards per Minute

The 300-cpm cycle occurs when the next CARD READ instruction is given before the last 20 ms of a card-read

$$T = .0999 + 10 + [21 + 1.3\ (L_B + 1)]\ ms$$

with:

.0999	=	Instruction Reading Time
10.0	=	Clutch Pickup Time
21.0	=	Reading Setup Time
1.3 (L_B + 1)	=	1.3 ms per core storage position in the B-field (L_B), plus one additional position for the group-mark with a word-mark

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
10.0	Clutch Pickup Time
21.0	Reading Setup Time
1.3	Time to read GM-WM

$T = 32.4 + 1.3\ L_B$ ms

AVAILABLE PROCESSING TIME (APPROXIMATE)

No. of Card Columns Read	System Interlocked (ms)	Process Time Available (ms)
1	34	176
20	58	152
40	84	126
60	110	100
80	136	74

Figure D-33. IBM 1442, Model 1; Timing—285 cpm

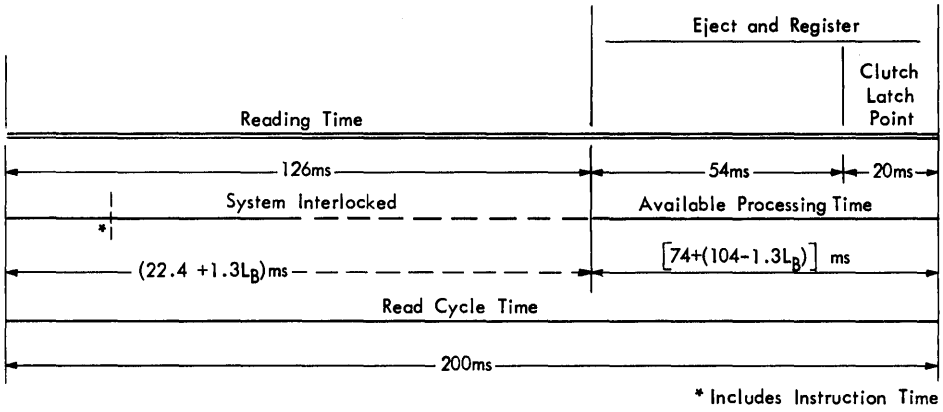


Figure D-34. Card-Read Cycle—300 cpm (80 Card Columns)

cycle (Figure D-34). The clutch remains engaged, and does not latch up. The clutch-pickup time of 10 ms is not needed, and the card-cycle time is reduced to 200 ms, which is equivalent to 300 cpm.

The time the system is interlocked during a card-read operation, when the clutch does not latch up after each operation, is shown in Figures D-34 and D-35. This timing formula can be consolidated as shown in Figure D-35. By using the consolidated formula, the total time available for other processing during one card-read cycle can be found by subtracting the consolidated formula from 200. The bottom portion of Figure D-35 shows the approximate processing time available by the number of card columns being read.

$T = .0999 + [21 + 1.3(L_B + 1)]$ ms

with:

.0999	=	Instruction Reading Time
21.0	=	Reading Setup Time
1.3 (L_B + 1)	=	1.3 ms per core storage position in the B-field (L_B), plus one additional position for the group-mark with a word-mark

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
21.0	Reading Setup Time
1.3	Time to Read GM-WM

T= 22.4 + 1.3 L_B ms

AVAILABLE PROCESSING TIME (APPROXIMATE)

No. of Card Columns Read	System Interlocked (ms)	Process Time available (ms)
1	24	176
20	48	152
40	74	126
60	100	100
80	126	74

Figure D-35. IBM 1442, Model 1; Timing—300 cpm

Fewer than 285 Cards per Minute

If the card-reading time and the necessary processing time between card-read cycles exceeds 210 ms, there is a corresponding drop in the number of cards read per minute. The formula to compute the number of cards read per minute is shown in Figure D-36. The table in Figure D-36 shows the approximate number of cards read per minute when the elapsed time between a given card column in a card-read operation is at least 210 ms later than the same card column in the preceding card-read operation.

Model 2 Card Reading

Card reading on the IBM 1442, Model 2, can be done at either 375 or 400 cards per minute (cpm).

CARDS READ PER MINUTE FORMULA

$$CPM = \frac{60{,}000}{X}$$

with:

60,000	=	number of ms in 1 minute
X	=	elapsed time between card read operations in ms. (This will be greater than, or equal to 210 ms)

Time Between Card Read Operations (ms)	CPM
210	285
300	200
400	150
500	120
600	100

Figure D-36. IBM 1442, Model 1; Timing, Fewer Than 285 cpm

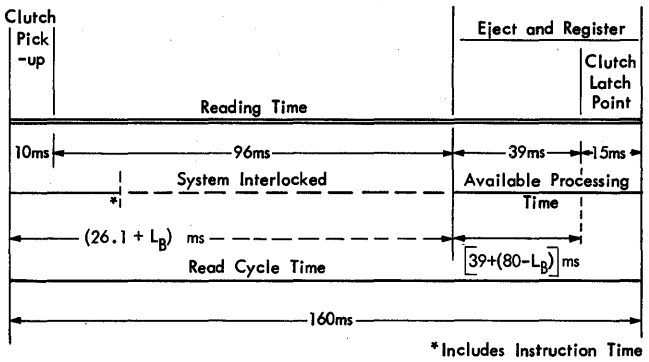


Figure D-37. Card-Read Cycle—375 cpm (80 Card Columns)

375 Cards per Minute

The 375-cpm cycle occurs when the next CARD READ instruction is given during the last 15 ms of a card-read cycle (Figure D-37). The clutch is allowed to latch up and the 10-ms clutch-pickup time must take place at the beginning of the next card-feed cycle.

The time the system is interlocked during a card-read operation, when the clutch latches up after each operation, is shown in Figures D-37 and D-38. This timing formula can be consolidated as shown in Figure D-38. By using the consolidated formula, the total time available for other processing during one card-read cycle can be found by subtracting the consolidated formula from 160. The bottom portion of Figure D-38

$T = .0999 + 10 + [15 + (L_B + 1)]$ ms

with:

.0999	=	Instruction Reading Time
10.0	=	Clutch Pickup Time
15.0	=	Reading Set up Time
(L_B + 1)	=	1.0 ms per core storage position in the B-field (L_B), plus one additional position for the group-mark with a word-mark

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
10.0	Clutch Pickup Time
15.0	Reading Setup Time
1.0	Time to Read GM-WM

$T = 26.1 + L_B$ ms

AVAILABLE PROCESSING TIME (APPROXIMATE)

No. of Card Columns Read	System Interlocked (ms)	Process Time available (ms)
1	27	133
20	47	113
40	67	93
60	87	73
80	107	53

Figure D-38. IBM 1442, Model 2; Timing—375 cpm

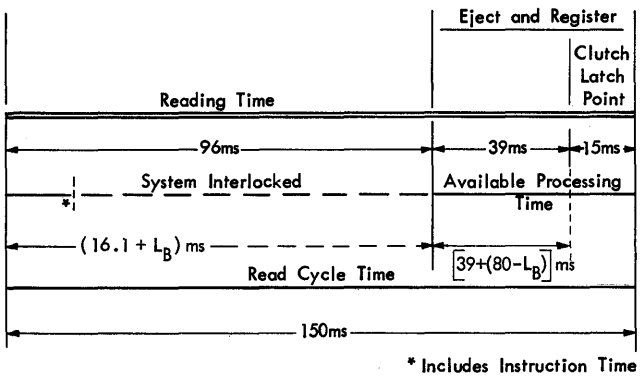


Figure D-39. Card-Read Cycle—400 cpm (80 Card Columns)

shows the approximate processing time available by the number of card columns being read.

400 Cards per Minute

The 400-cpm cycle occurs when the next CARD READ instruction is given before the last 15 ms of a card-read cycle (Figure D-39). The clutch-pickup time of 10 ms is not needed, and the card-cycle time is reduced to 150 ms, which is equivalent to 400 cpm.

The time the system is interlocked during a card-read operation, when the clutch does not latch up after each operation, is shown in Figures D-39 and D-40.

$T = .0999 + [15 + (L_B + 1)]$ ms

with:

.0999	=	Instruction Reading Time
15.0	=	Reading Setup Time
(L_B + 1)	=	1.0 ms per core storage position in the B-field (L_B), plus one additional position for the group-mark with a word-mark

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
15.0	Reading Setup Time
1.0	Time to Read GM-WM

$T = 16.1 + L_B$ ms

AVAILABLE PROCESSING TIME (APPROXIMATE)

No. of Card Columns Read	System Interlocked (ms)	Process Time available (ms)
1	17	133
20	37	113
40	57	93
60	77	73
80	97	53

Figure D-40. IBM 1442, Model 2; Timing—400 cpm

This timing formula can be consolidated as shown in Figure D-40. By using the consolidated formula, the total time available for other processing during one card-read cycle can be found by subtracting the consolidated formula from 150. The bottom portion of Figure D-40 shows the approximate processing time available by the number of card columns being read.

Fewer than 375 Cards per Minute

If the card-reading time and the necessary processing time between card-read cycles exceeds 160 ms, there is a corresponding drop in the number of cards read per minute. The formula to compute the number of cards read per minute is shown in Figure D-41. The bottom portion of Figure D-41 shows the approximate number of cards read per minute when the elapsed time between a fixed point in a card-read operation is at least, or more than, 160 ms later than the same fixed point in the preceding card-read operation.

Model 1 Card Punching

The total punch-cycle time in an IBM 1442, Model 1, depends on the number of card columns being punched and the operation being performed. Punching one card column requires 12.5 ms. Eighty card columns at 12.5 ms per column, therefore, require 1,000 ms.

Punch and Stop Operation

The only timing involved in a punch-and-stop operation is the instruction-reading time, the clutch-pickup time, and the card-column-punching time as shown in Figure D-42. Card movement is not considered, because the card is already located under the punching mechanism at the beginning of the operation, and the card is still under the punching mechanism when the operation ends.

CARDS READ PER MINUTE FORMULA

$$CPM = \frac{60{,}000}{X}$$

with:

60,000	=	Number of ms in 1 minute
X	=	Elapsed time between card read operations in ms. (This will be greater than, or equal to 160 ms).

TIME BETWEEN CARD READ OPERATIONS (ms)	CPM
160	375
200	300
300	200
400	150
500	120

Figure D-41. IBM 1442, Model 2; Timing—Fewer Than 375 cpm

$$T = .0999 + 6.25 + 12.5\ (L_B)\ ms$$

with:

.0999	=	Instruction Reading Time
6.25	=	Average clutch pickup time
12.5 (L_B)	=	12.5 ms punching cycle per core storage position in the B-field (L_B)

Figure D-42. IBM 1442, Model 1; Punch-and-Stop Timing

Punch and Feed Operation

There are three parts to a punch-and-feed operation: the instruction-reading operation, the punching operation, and the feeding operation. The instruction is read during the instruction-reading portion of the punch-and-feed operation, and requires .0111 ($L_I + 1$) ms. The time varies during the punching portion of the operation from 12.5 ms (1 column) to 1,000 ms (80 columns). When the punching is done, the card is ejected from the punching station, another card moves from the reading station to the punching station, and another card moves from the hopper to the reading station. This portion of the operation takes 210 ms, and other processing can take place during that time.

The timing formula can be consolidated as shown in Figure D-43. The cards per minute can be determined by dividing 60,000 (number of milliseconds in one minute) by the consolidated formula. The bottom portion of Figure D-43 shows the breakdown of punch-cycle time and the associated cards per minute rate by the number of card columns punched.

Fewer than 50 Cards per Minute

If the total punch-cycle time is greater than 1,216 ms (the time required to punch 80 columns plus 216 ms), there is a corresponding drop in the number of cards punched per minute. The formula to compute the number of cards punched per minute is shown in Figure D-44. The bottom portion of Figure D-44 shows the approximate number of cards punched per minute when the elapsed time between a fixed point in a punch and feed operation is at least, or more than, 1,216 ms later than the same fixed point in the preceding punch-and-feed operation.

$T = .0999 + 6.25 + 12.5\ (L_B) + 210.0$ ms

with:

.0999	=	Instruction Reading Time
6.25	=	Average clutch pickup time
12.5 (L_B)	=	12.5 ms punching cycle per core storage position in the B-field (L_B)
210.0	=	Card read cycle at end of punch cycle that moves card into punching station.

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
6.25	Clutch pickup
210.0	Card Read Cycle

$T = 216.35 + 12.5\ L_B$ ms

PUNCH CYCLE TIMES AND CPM RATES (APPROXIMATE)

No. of Card Columns Punched	System Interlocked (ms)	Total Punch Cycle Time (ms)	CPM
80	1000	1216	49
75	938	1154	52
70	875	1091	55
65	813	1029	58
60	750	966	62
55	688	904	66
50	625	841	71
45	563	779	77
40	500	716	84
35	438	654	92
30	375	591	102
25	313	529	113
20	250	466	127
15	188	404	149
10	125	341	176
5	63	279	215
1	13	229	202

Figure D-43. IBM 1442, Model 1, Punch-and-Feed Timing

Model 2 Card Punching

The total punch-cycle time on an IBM 1442, Model 2, depends on the number of card columns being punched and the operation being performed. Punching one card column requires 6.25 ms. Eighty card columns at 6.25 ms per column, therefore, requires 500 ms.

Punch and Stop Operation

The only timing involved in a punch-and-stop operation is the instruction-reading time, the clutch-pickup time, and the card-column-punching time as shown in Figure D-45. Card movement is not considered, because the card is already located under the punching mechanism at the beginning of the operation, and the card is still under the punching mechanism when the operation ends.

CARDS PUNCHED PER MINUTE FORMULA

$$CPM = \frac{60{,}000}{X}$$

with:

60,000	=	Number of ms in one minute
X	=	Elapsed time between punch and feed operations in ms. (This will be greater than, or equal to 1210 ms)

TIME BETWEEN PUNCH AND FEED OPERATION (ms)	CPM
1216	49
1300	46
1500	40
2000	30

Figure D-44. IBM 1442, Model 1, Punch and Feed Timing—Fewer Than 50 cpm

Punch and Feed Operation

There are three parts to a punch-and-feed operation: the instruction-reading operation, the punching operation, and the feeding operation. The instruction is read during the instruction-reading portion of the punch-and-feed operation and requires .0111 ($L_I + 1$) ms. The time varies during the punching portion of the operation from 6.25 ms (1 column) to 500 ms (80 columns). When the punching is done, the card is ejected from the punching station, another card moves from the reading station to the punching station, and another

$T = .0999 + 3.13 + 6.25\ (L_B)$ ms

with:

.0999	=	Instruction Reading Time
3.13	=	Average clutch pickup time
6.25 (L_B)	=	12.5 ms punching cycle per core storage position in the B-field (L_B)

Figure D-45. IBM 1442, Model 2, Punch and Stop Timing

card moves from the hopper to the reading station. This portion of the operation takes 160 ms, and other processing can take place during that time.

The timing formula shown in Figure 46 can be consolidated as shown. The cards per minute can be determined by dividing 60,000 (number of ms in one minute) by the consolidated formula. The bottom portion of Figure D-46 shows the breakdown of punch-cycle time and the associated cards per minute rate by the number of card columns punched.

Fewer than 91 Cards per Minute

If the total punch-cycle time is greater than 663 ms (the time required to punch 80 columns plus 163 ms), there is a corresponding drop in the number of cards punched per minute. The formula to compute the number of cards punched per minute is shown in Figure D-47. The bottom portion of Figure D-47 shows the approximate number of cards punched per minute when the elapsed time between a fixed point in a punch-and-feed operation is at least, or more than, 663 ms later than the same fixed point in the preceding punch-and-feed operation.

$T = .0999 + 3.13 + 6.25\ (L_B) + 160$ ms

with:

.0999	=	Instruction Reading Time
3.13	=	Average Clutch Pickup Time
6.25 (L_B)	=	6.25 punching cycle per core storage position in the B-field (L_B)
160.0	=	Card read cycle at end of punch cycle that moves card into punching station.

TIMING FORMULA CONSOLIDATION

.1 ms	Instruction Reading Time (.0999)
3.13	Clutch Pickup
160.0	Card Read Cycle

$T = 163.23 + 6.25\ L_B$ ms

PUNCH CYCLE TIMES AND CPM RATES (APPROXIMATE)

No. of Card Columns Punched	System Interlocked (ms)	Total Punch Cycle Time (ms)	CPM
80	500	663	91
75	469	632	95
70	438	601	100
65	406	569	105
60	375	538	112
55	344	507	118
50	313	476	126
45	281	444	135
40	250	413	145
35	219	382	157
30	188	351	171
25	156	319	188
20	125	288	208
15	94	257	233
10	63	226	265
5	31	194	309
1	6	169	355

Figure D-46. IBM 1442, Model 2, Punch and Feed Timing

CARDS PUNCHED PER MINUTE FORMULA

$$CPM = \frac{60{,}000}{X}$$

with:

60,000	=	Number of ms in a minute
X	=	Elapsed time between punch and feed operations in ms. (This will be greater than, or equal to 660 ms).

TIME BETWEEN PUNCH AND FEED OPERATIONS (ms)	CPM
663	91
750	80
900	66
1000	60

Figure D-47. IBM 1442, Model 2, Punch and Feed Timing—Fewer Than 91 cpm

Combined Reading and Punching, Models 1 and 2

Certain applications call for reading a card, processing the information read, and then punching the result in the same card. Because the IBM 1442 is a serial-type machine, the card design significantly affects the card-per-minute rate through the 1442, as shown in Figures D-48 and D-49.

The example illustrated in Figure D-48 assumes that card columns 1-40 are read during the read cycle and card columns 41-80 are punched during the punch operation.

The read operation requires 210 ms (Model 1 timing at 285 cpm). Because only 40 columns are read, the

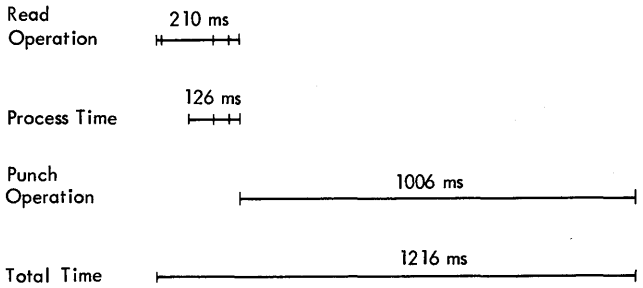


Figure D-48. Read from Columns 1-40; Punch into Columns 41-80

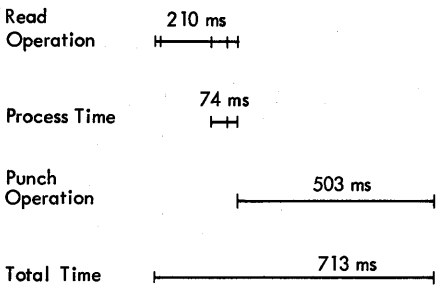


Figure D-49. Punch into Columns 1-40; Read from Columns 41-80

last 126 ms of the operation can be used to process the information.

The punch operation takes 1,000 ms. Because the last 40 columns are being punched, the B-field in core storage must be 80 positions in length (the first 40 positions contain blanks). A 12.5-ms cycle occurs for each one of these 40 positions containing a blank, even though no punching occurs. The data in the last 40 positions of the 80-position field is punched in the last 40 columns of the card. This brings the time for the punch operation to 1,000 ms (80 columns @ 12.5 ms/col plus 6.25 ms for clutch-pickup time), and the total time for the entire operation to 1,216 ms.

The example illustrated in Figure D-49 assumes that card columns 41-80 are read during the read cycle, and card columns 1-40 are punched during the punch operation.

The read operation requires 210 ms as before. The 40 columns read during this operation occur at the end of reading time, and none of the 126-ms reading time can be used for other processing. Because of this, only the last 74 ms of the read cycle can be used for other processing.

In this example the available processing time is less than the time in the first example, because the reading takes place during the entire 126 ms. The data is located in the last 40 columns of the card, but the first 40 columns are also read. Figure D-50 illustrates the relationship between the reading of card columns and the available processing time.

The punch operation takes only 503 ms. Because the first 40 columns are being punched, the B-field in core storage must be only 40 positions in length. This brings the total time for the entire operation to 713 ms.

As can be seen from these examples, the important consideration is not how many columns are punched, but where the punched columns are in the card. Punching 5 columns in the first 5 columns of the card instead of columns 26-30, for example, results in a faster cpm rate (Figure D-51). A cpm rate of 215 results on an IBM 1442, Model 2, when the first 5 columns are punched; a cpm rate of 102 when columns 26-30 are punched.

Last Card Column Read	Available Processing Time	
	Model 1	Model 2
5	171	128
10	165	123
15	158	118
20	152	113
25	146	108
30	139	103
35	133	98
40	126	93
45	119	88
50	113	83
55	106	78
60	100	73
65	93	68
70	87	63
75	80	58
80	74	53

Figure D-50. Relationship Between Card Columns Read and Available Processing Time Left (Approximate)

Last Card Column Read	Available Processing Time	
	Model 1	Model 2
5	215	309
10	176	265
15	149	233
20	127	208
25	113	188
30	102	171
35	92	157
40	84	145
45	77	135
50	71	126
55	66	118
60	62	112
65	58	105
70	55	100
75	52	95
80	49	91

Figure D-51. CPM Rates for Punching into a Card Previously Read (Approximate)

IBM 1443 Printer

The IBM 1443 Printer (Figure D-52) is another output medium for the 1440 system. The number of lines that can be printed per minute depends on the 1443 model and the character set being used. Refer to *IBM 1443 Printer*, Form A24-3120.

IBM 1443 Printer Instruction Format

All printer operations are initiated by either one of two types of printer instructions. If the instruction is two characters long, an operation involving the printer carriage is specified. If the instruction is eight characters long, an operation involving a write operation is specified. The various parts of the printer instruction (Figure D-53) are:

General Mode of Operation

This part of the instruction identifies the operation as either a write operation or a carriage operation. The write operation is performed in the move mode. Any word marks in the specified core-storage area are unaffected by the operation.

d-Modifier Character

If the instruction is two characters long, the second character is the d-modifier character. This character specifies the type of carriage operation that will occur. Refer to Figure D-54 for a list of the d-characters and the carriage operations they initiate.

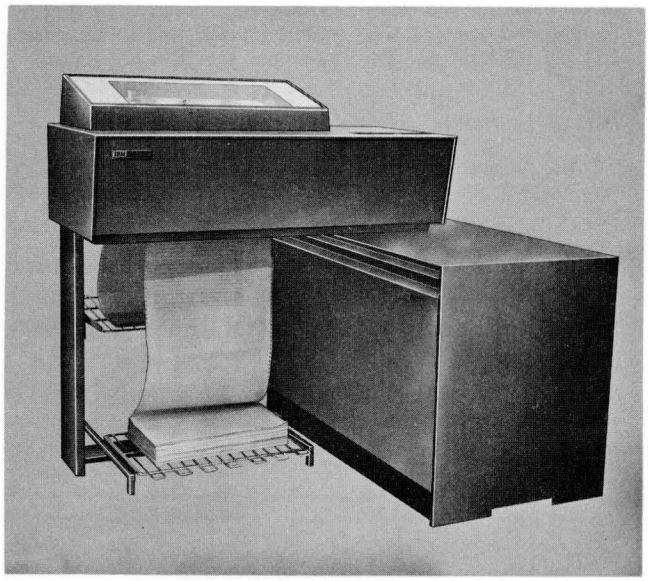


Figure D-52. IBM 1443 Printer

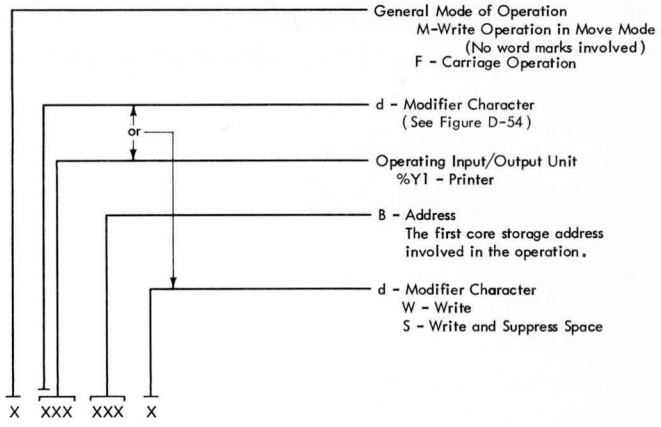


Figure D-53. IBM 1443 Printer Instruction Format

Operating Input/Output Unit

This part of the instruction specifies the printer as the active unit for this operation.

B-Address

This part of the instruction specifies the first core-storage position that will be involved in the operation.

d-Modifier Character

This part of the instruction specifies the type of write operation that will be performed in the printer, when the d-character modifies a write operation code.

d	Immediate skip to	d	Skip after print to
1	Channel 1	A	Channel 1
2	Channel 2	B	Channel 2
3	Channel 3	C	Channel 3
4	Channel 4	D	Channel 4
5	Channel 5	E	Channel 5
6	Channel 6	F	Channel 6
7	Channel 7	G	Channel 7
8	Channel 8	H	Channel 8
9	Channel 9	I	Channel 9
0	Channel 10	?	Channel 10
#	Channel 11	•	Channel 11
@	Channel 12	□	Channel 12
d	**Immediate space**	**d**	**After print-space**
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

Figure D-54. Control Carriage d-Characters

IBM 1443 Printer Instructions

Instructions applying to the 1443 cannot be successfully chained.

Write Line

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
W	M	%Y1	B01	W

Function. This instruction is used to transfer data from core storage to the 1443 printer, where it will be printed.

The high-order position of data in the core-storage position specified by the B-address is transferred and printed in print-position 1. The rest of the data located in the adjacent core-storage positions is transferred, column-by-column, and printed in the adjacent print positions until a group mark with a word mark in core storage is sensed.

The B-address must always specify one of the zero-one positions (x01) in core storage when using an unbuffered printer. The number of characters printed depends on the B-field length established in core storage. The B-field length can be from 1 to either 120 or 144 positions (24 additional print positions are available as a special feature), plus one position for the group mark with a word mark. An automatic single space operation occurs after the actual printing ends unless a different carriage operation is programmed.

Word Marks. Word marks are not affected. A group mark with a word mark is required to end the operation.

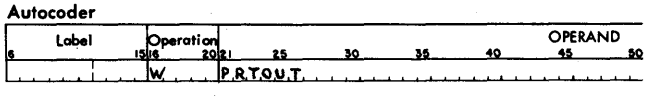
Timing. T = .0999 + 386* ms.
*120 print positions

Note. An address-validity-check condition occurs if the B-address specifies the 01 position of the last 100-position block in core storage as well as any starting position other than 01 (unbuffered printer). The system interlocks with the console I/O printer light ON. The 1443 goes out of ready status.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	$B + L_B + 1$

Example. Print the data beginning in the area labeled PRTOUT (0101) and ending with a group mark with a word mark (Figure D-55).



Assembled Instruction: M %Y1 101 W

Figure D-55. Write Line

Write Line and Suppress Space

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WS	M	%Y1	B01	S

Function. This instruction is used to transfer data from core storage to the 1443 where it will be printed. The automatic single space, normally taken after printing, is suppressed.

Data in the core-storage position specified by the B-address is transferred and printed in print-position 1. The B-address must always specify one of the zero-one positions (x01) in core storage when using an unbuffered printer. The rest of the data located in the adjacent core-storage positions is transferred, character-by-character, and printed in the adjacent print positions until a group mark with a word mark in core storage is sensed. The number of characters printed depends on the B-field established in core storage. The B-field lengths can be from 1 to either 120 or 144 positions (24 additional print positions are available as a special feature), plus one position for the group mark with a word mark.

Word Marks. Word marks are not affected. A group mark with a word mark is needed to end the operation.

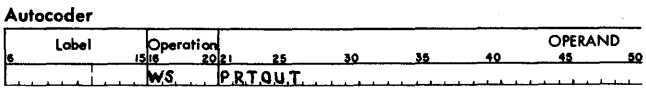
Timing. T = .0999 + 386* ms.
*120 print positions

Note: An address validity-check condition occurs if the B-address specifies the 01 position of the last 100-position block of core storage as well as any starting position other than 01 (unbuffered printer). The system interlocks with the console I/O printer light on. The 1443 goes out of ready status.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	$B + L_B + 1$

Example. Print the data beginning in the area labeled PRTOUT (0101) and ending with a group mark with a word mark, and suppress the automatic single space (Figure D-56).



Assembled Instruction: M %Y1 101 S

Figure D-56. Write Line and Suppress Space

Branch if Printer Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	‡

Function. If an error occurs during a printer operation, this indicator is set ON, and the printer light glows on the console. This indicator can be tested to effect a branch. If the indicator is on, it is reset. The next instruction to be executed is at the location specified by the I-address of the BRANCH IF INDICATOR ON instruction.

Note: This error indicator applies to any 1403, 1443, or 1445 attached to the system, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

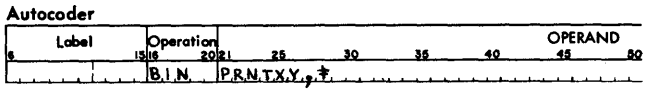
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	‡bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer-error indicator to determine if an error occurred on the immediately preceding print instruction. If an error occurred, branch to the instruction labeled PRNTXY (0661). If no error occurred, continue processing with execution of NSI (Figure D-57).



Assembled Instruction: B 661 ‡

Figure D-57. Branch If Printer Error

Branch if Printer Busy

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BPB	B	III	P

Function. If the printer or printer carriage is currently occupied with another operation, this indicator can be tested to effect a branch to another series of instructions. The indicator is reset as soon as the printer is available for another operation. Using this instruction allows processing to continue while the printer is busy, thus, in effect, allowing temporary overlapping of processing and printer operation.

Note 1: This indicator is effective for any 1403, 1443, or 1445 attached to the system.

Note 2: The *Branch if Printer Busy* instruction should be included in any program where one print and space/skip operation has not been completed before the next print and space operation is initiated. If this instruction is not used, incorrect carriage spacing or skipping can result. The following sample program segment illustrates how these instructions might be coded:

```
PRBUSY    BPBB     PRBUSY,
SPACE     CC       S
WRITE     W        NET PAY
          NSI
```

Frequently, there are other processing steps that can be performed while waiting for the carriage to complete the last print/space/skip function. The *Branch if Printer Busy* instruction could have branched to perform these other functions.

This precaution applies only to systems having buffered printers.

Word Marks. Word marks are not affected.

Timing.

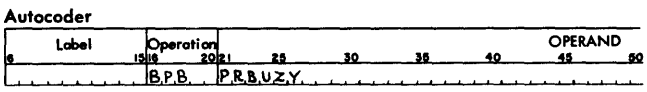
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Pbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer-busy indicator to determine if the printer is occupied with some other operation. If it is, branch to another series of instructions beginning at the instruction labeled PRBUZY (0486) while waiting for the printer to become available. If the printer is *not* busy, execute the NSI (Figure D-58).



Assembled Instruction: B 486 P

Figure D-58. Branch If Printer Busy

Branch if Channel 9

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BC9	B	III	9

Function. If the channel 9 position of the carriage-control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Word Marks. Word marks are not affected.

Timing.

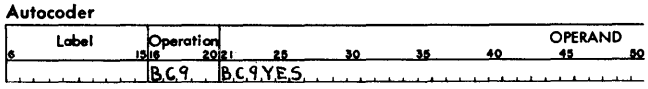
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	9bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test to determine if the carriage-control tape (forms) channel 9 has been sensed. Branch to a subroutine called BC9YES (0784) if the test is positive (Figure D-59).



Assembled Instruction: B 784 9

Figure D-59. Branch on Printer Carriage Channel 9

Branch if Channel 12

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BCV	B	III	@

Function. If the channel 12 (forms overflow) position of the carriage-control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Word Marks. Word marks are not affected.

Timing.

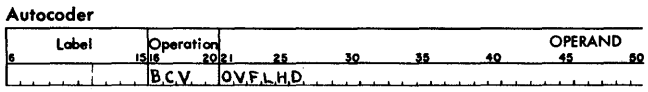
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	@bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to a subroutine labeled OVFLHD (0659) if the hole in channel 12 of the carriage-control tape has been sensed, indicating that the present form is filled; Figure D-60.



Assembled Instruction: B 659 @

Figure D-60. Branch on Printer Carriage Forms Overflow

Control Carriage

Instruction Format.

Mnemonic	*Op Code*	*d-character*
CC	F	d (see text)

Function. This instruction causes the carriage to move as specified by the d-character. If the d-character is:

1. a digit, an immediate skip to the specified channel in the carriage tape occurs.

2. an alphabetic character containing a 12-zone, a skip to the specified channel in the carriage tape occurs after the next line is printed.

3. an alphabetic character containing an 11-zone, an immediate space operation, as specified by the digit portion of the character, occurs.

4. an alphabetic character containing a zero-zone, a space operation, as specified by the digit portion of the character, occurs after the next line is printed.

Refer to Figure D-54 for a list of the d-characters and the carriage operations they specify. If the carriage is already in motion when another CONTROL CARRIAGE instruction is given, the stored program execution is suspended until the carriage operation being performed is completed. At that time, the carriage action specified by the instruction begins, and the program advances to the next instruction.

Note: There is no CARRIAGE CONTROL AND BRANCH (CCB) instruction for the 1440 system.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms + remaining form-movement time, if carriage is already in motion when this instruction is given. The total form movement time depends on the specific carriage operation being performed. Refer to the IBM *1443 Printer Timing* section for more detail.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

Example. Skip to channel 1 after print operation (Figure D-61).

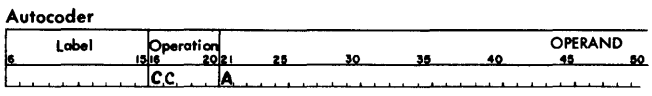


Figure D-61. Control Carriage

IBM 1443 Printer Timing

Model 1 Printing Speed

Model 1 of the IBM 1443 Printer operates at a maximum rated speed of 150 lines per minute when the 52-character typebar is installed.

The print cycle is 400 ms long (Figure D-62). A total of 368 ms is needed during the 400-ms print cycle to transfer the data from core storage and print it. The form movement takes place during the last 32 ms of the print cycle. Up to two lines of form movement can take place during this time if the delayed forms op is programmed prior to the WRITE instruction. Additional lines (beyond 2) extend the print-cycle time by 10 ms per line. To establish the new line-per-minute rate, divide 60,000 by the print-cycle time (400 ms) to have a 150-ms time for one print line. For information on the additional form-movement timing, refer to the *Carriage Speed* section.

No other processing can take place during the data-transfer and print time. The entire form-movement time is available to perform other systems operations.

Model 2 Printing Speed

Model 2 of the IBM 1443 Printer operates at a maximum rated speed of 240 lines per minute when the 52-character typebar is installed. The duration of the print cycle is 250 ms (Figure D-63). A total of 218 ms is needed during the 250-ms print cycle to transfer the data from core storage and print it.

Up to two lines of form movement can take place during the normal print cycle. Additional lines extend the print-cycle time by 10 ms per line. For information on the additional form-movement timing, refer to the *Carriage Speed* section.

No other processing can take place during the data-transfer and print time. The entire form-movement time (32 ms) is available to perform other systems operations.

Carriage Speed

Normal Form-Movement Operation

Form movement is normally accomplished during the last 32 ms of a print cycle. It is possible to space two lines during the normal print cycle, if a DELAYED CONTROL CARRIAGE instruction is programmed before the WRITE instruction. Each additional line requires another 10 ms. This speed is equivalent to approximately 15 inches per second.

Immediate Form-Movement Operation

Figure D-64 shows various timings that result when an immediate form-movement operation is specified by the CONTROL CARRIAGE instruction. If the carriage is already in motion when the instruction is given, the stored program execution is suspended until the carriage operation being performed is completed. At that time, the immediate form-movement operation, specified by the instruction, begins. The time required for spacing the first line is 60 ms, and each additional line requires another 10 ms.

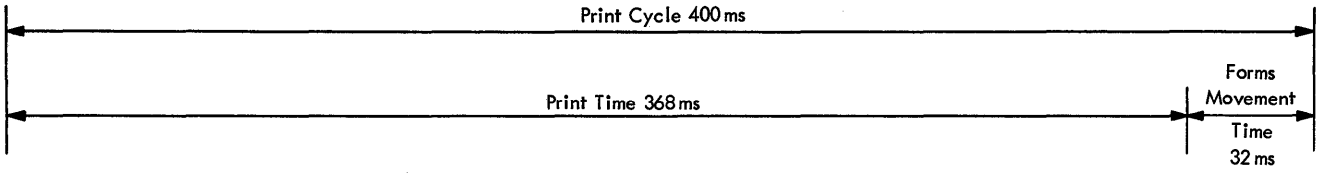


Figure D-62. IBM 1443, Model 1, Print Cycle; 52-Character Typebar

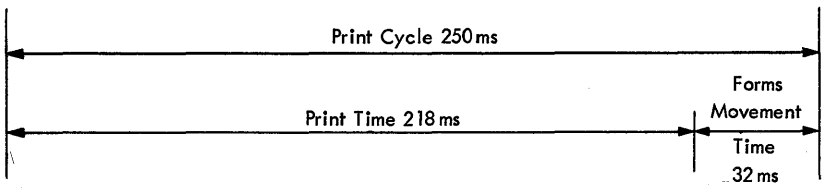


Figure D-63. IBM 1443, Model 2, Print Cycle; 52-Character Typebar

MODEL 1		MODEL 2		Available Process Time (ms)	No. of Lines Spaced/ Skipped	Total No. of Lines Spaced/ Skipped
LPM	Cycle Length (ms)	LPM	Cycle Length (ms)			
133	450	200	300	74	1	2
130	460	196	310	84	2	3
127	470	188	320	94	3	4
125	480	182	330	104	4	5
122	490	176	340	114	5	6

Figure D-64. Immediate Forms Space/Skip Operation Timings

MODEL 1		MODEL 2		Available Process Time (ms)	Total No. of Lines Spaced/ Skipped
LPM	Cycle Length (ms)	LPM	Cycle Length (ms)		
150	400	240	250	24	1
150	400	240	250	24	2
146	410	231	260	34	3
143	420	222	270	44	4
140	430	214	280	54	5

Figure D-65. Delayed Forms Space/Skip Operation Timings

IBM 1444 Card Punch

The IBM 1444 Card Punch (Figure D-66) provides a high-speed card output to the IBM 1440 Data Processing System. This section describes the instruction used with the 1440 system to control the card punch. (Refer to *IBM 1444 Card Punch*, Form A24-3152).

IBM 1444 Card Punch Instructions

Instructions applying to the 1444 cannot be successfully chained.

Punch Card

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
P	M	%G3	B01	G

Function. This instruction is used to transfer data from core storage into the card punch where it is punched in a card. The data transfer from core storage to the punch ends when a group mark with a word mark is sensed.

Note: The first 1442, Model 1, 2, or 4, attached to the system has a unit-select number of 1. The second 1442, Model 1, 2, or 4, has a unit-select number of 2. Only one 1444 can be attached to the system. Its unit-select number is always 3.

Figure D-66. IBM 1444 Card Punch

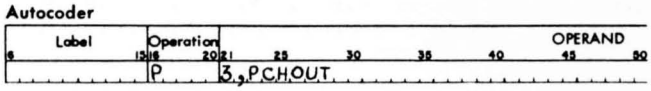
Word Marks. Word marks associated with the data being transferred are neither considered nor affected. The data transfer ends when the group mark with a word mark located in core-storage position B01 + L_B (length of B-field) is sensed.

Timing. T = .0999 ms + I/O. Input/Ouput time equals 240 ms plus punch-access time of 0-60 ms. The processing-unit interlock is released after 217.5 ms of the 240-ms punch cycle.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	B + L_B + 1

Example. Punch the data on card punch, beginning in the area labeled PCHOUT (401) and ending with a group mark with a word mark (Figure D-67).



Assembled Instruction: M %G3 401 G

Figure D-67. Punch Card

Select Stacker

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	#

Function. This instruction causes the card that was just punched to be selected into stacker 2 after the next punch operation takes place. (The card just punched must be checked at the punch-read station before it can be stacked.) If a punch-check condition occurs during the next punch operation, the card is automatically directed to stacker 1.

Note 1: This instruction must be issued prior to the PUNCH AND GO or READ CARD instruction that moves the card on through the feed.

Note 2: The IBM 1444 and the IBM 1442, Models 2 and 4, have two stackers as standard equipment. A second stacker is provided in the IBM 1442, Model 1, as a special feature.

Word Marks. Word marks are neither considered nor affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	#bb	#bb

Example: Place the card, just punched, in stacker 2 (Figure D-68).

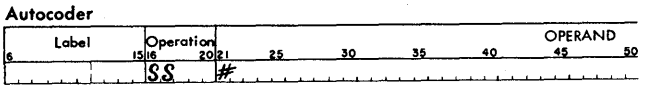


Figure D-68. Select Card in Stacker 2

Branch if Punch Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	! (minus zero)

Function. Same as for *Reader Error,* except that specifics apply to the punch operation.

Note: This error indicator is effective for the 1444 and first or second 1442 Model 1 or 2 punch operations, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	!bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to the instruction labeled PUNCHZ (3775) if a punch operation error occurred on any attached punch device. If no punch error occurred, proceed to the NSI (Figure D-69).

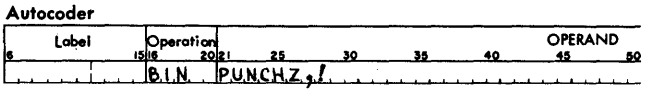


Figure D-69. Branch If Punch Error

IBM 1444 Card Punch Timing

The card punch operates at a rated speed of 250 cycles per minute (240 ms per cycle). Actual card punching, at an optimum rate of 250 cards per minute, is controlled by punch instructions in the program.

There are four points in the cycle occurring at 60-millisecond intervals when the punch feeding mechanism can receive an impulse to start the punch cycle.

The punch cycle is divided into three separate functions (Figure D-70):

1. Punch-start time is 37 ms. After the feed mechanism has been impulsed, the time required for the card to feed and be positioned for punching is called *punch-start time.* The IBM processing unit is interlocked during punch-start time.
2. Card punching time is 181 ms. Actual card punching takes place during this part of the cycle. The processing unit is always interlocked during *card-punching time.*
3. Processing time is 22 ms. This is the remainder of the punch cycle allotted for *processing* in the system.

The next PUNCH CARD instruction must be given during this 22-ms period, or the punch operation will end, and at least 60 ms will elapse before the punch can start again.

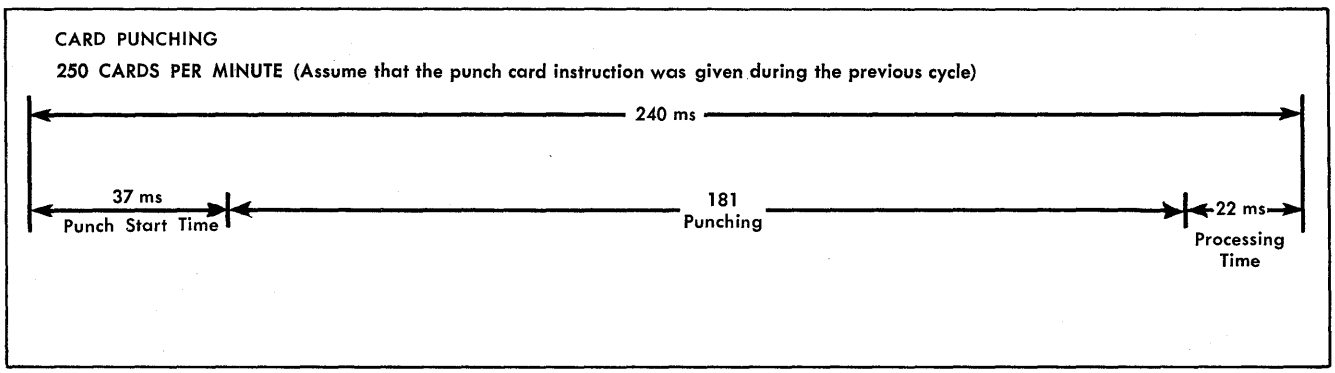


Figure D-70. Punch Cycle

feed and be positioned for punching is called *punch-start time.* The IBM processing unit is interlocked during punch-start time.

2. Card punching time is 181 ms. Actual card punching takes place during this part of the cycle. The processing unit is always interlocked during *card-punching time.*
3. Processing time is 22 ms. This is the remainder of the punch cycle allotted for *processing* in the system.

The next PUNCH CARD instruction must be given during this 22-ms period, or the punch operation will end, and at least 60 ms will elapse before the punch can start again.

Figure D-71 shows card-punching speeds and the processing time available with each.

Cards Punched Per Minute	Length of Cycle (ms)	Processing Time (ms)
250	240	22
200	300	82
166	360	142
143	420	202
125	480	262

Figure D-71. Card Punching Speeds

IBM 1445 Printer

The IBM 1445 Printer provides a means of inscribing in magnetic ink A.B.A. (E-13B) type font (Figure D-72) as well as conventional characters for another medium of printed output. Refer to *IBM 1445 Printer Models 1 and N1*, Form A24-3210.

IBM 1445 Printer Instructions

Instructions applying to the 1445 cannot be successfully chained.

Write Line

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
W	M	%Y1	B01	W

Function. This instruction is used to transfer data from core storage to the 1445 printer to be printed.

The high-order position of data in the core-storage position specified by the B-address is transferred and printed in print-position 1. The rest of the data located in the adjacent core-storage positions is transferred, character-by-character, and printed in the adjacent print positions until a group mark with a word mark in core storage is sensed.

On unbuffered systems the B-address (B01) must specify an address with the units and tens positions being 01, except the last 100-position block of storage. The number of characters printed depends on the B-field length established in core storage. The B-field length is 113 positions plus one for the group mark with a word mark. An automatic single-space operation occurs after the actual printing ends unless a different carriage operation is programmed.

Word Marks. Word marks are not affected. A group mark with a word mark is required to end the operation.

Timing. T = .0999 + 361 ms.

Note: An address-validity-check condition occurs if the B-address specifies x01 position of the last 100-position block of core storage, as well as any starting position other than x01 for unbuffered systems. The system interlocks with the console I/O printer light on. The 1445 goes out of ready status.

Address Registers After Operation.

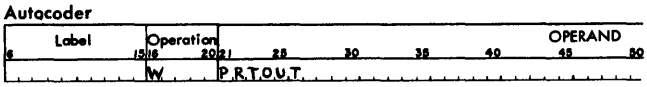
I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	$B + L_B + 1$

Example. Print the data beginning at the core-storage address of the label PRTOUT (0101), and ending

E-13B Character	Card Code	BCD Code							Equivalent Character	Name
0	0-4-8			A	8	4			% or (	Percent or Left Parenthesis
1	0-5-8	C		A	8	4		1	¥	Word Separator
2	0-7-8			A	8	4	2	1	⧺	Tape Segment Mark
3	3-8				8		2	1	# or =	Number Sign or Equal Sign
4	4-8	C			8	4			@ or '	At Sign or Apostrophe
5	5-8				8	4		1	:	Colon
6	7-8	C			8	4	2	1	√‾	Tape Mark
7	12-0	C	B	A	8		2		?	(Plus Zero)
8	11-0		B		8		2		!	(Minus Zero)
9	0-2-8			A	8		2		‡	Record Mark
‡	12-5-8		B	A	8	4		1	[	Left Bracket
'	11-5-8	C	B		8	4		1	]	Right Bracket
⧺	12-4-8	C	B	A	8	4			⌑ or)	Lozenge or Right Parenthesis
¥	12	C	B	A					& or +	Ampersand or Plus Sign

Figure D-72. E-13B Characters and Codes

with the core-storage location containing a group mark with a word mark (Figure D-73).



Assembled Instruction: M %Y1 101 W

Figure D-73 Write Line

Write Line and Suppress Space

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WS	M	%Y1	B01	S

Function. This instruction is used to transfer data from core storage to the 1445 to be printed. The automatic single space, normally taken after printing, is suppressed.

Data in the core-storage position specified by the B-address is transferred and printed in print-position 1. The rest of the data located in the adjacent core-storage positions is transferred, character-by-character and printed in the adjacent print positions until a group mark with a word mark in core storage is sensed. The number of characters printed depends on the B-field established in core storage. The B-field length is 113 positions plus one for the group mark with a word mark.

Word Marks. Word marks are not affected. A group mark with a word mark is needed to end the operation.

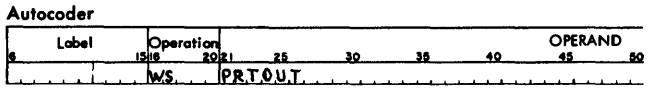
Timing. T = .0999 + 361 ms.

Note: An address-validity-check condition occurs if the B-address specifies x01 position of the last 100-position block of core storage, as well as any starting position other than x01 for unbuffered systems. The system interlocks with the console I/O printer light on. The 1445 goes out of ready status.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	$B + L_B + 1$

Example. Print the data beginning at the core-storage address of the label PRTOUT (0101), and ending with the core-storage location containing a group mark with a word mark, and suppress the automatic single space (Figure D-74).



Assembled Instruction: M %Y1 101 S

Figure D-74. Write Line and Suppress Space

Branch if Printer Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	‡

Function. If an error occurs during a printer operation, this indicator is set ON, and the printer light glows on the console. This indicator can be tested to effect a branch. If the indicator is ON, it is reset. The next instruction to be executed is at the location specified by the I-address of the BRANCH IF INDICATOR ON instruction.

Note: This error indicator applies to any 1403, 1443, or 1445 attached to the system, when the I/O check-stop switch is OFF.

Word Marks. Word marks are not affected.

Timing.

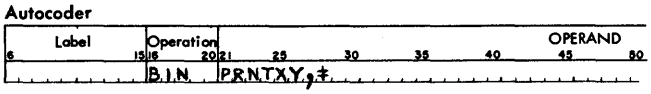
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	‡bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer-error indicator to determine if an error occurred on the immediately preceding print instruction. If an error occurred, branch to the instruction labeled PRNTXY (0661). If no error occurred, continue processing with NSI (Figure D-75).



Assembled Instruction: B 661 ‡

Figure D-75. Branch If Printer Error

Branch if Printer Busy

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BPB	B	III	P

Function. If the printer or printer carriage is occupied with another operation, this indicator can be tested to effect a branch to another series of instruction. The indicator is reset as soon as the printer is available for another operation. Using this instruction allows processing to continue while the printer is busy, thus, in effect, allowing temporary overlapping of processing and printer operation.

Note: This indicator is effective for any 1403, 1443, or 1445 attached to the system.

Word Marks. Word marks are not affected.

Note 1: This indicator is effective for any 1403, 1443, or 1445 attached to the system.

Note 2: The *Branch if Printer Busy* instruction should be included in any program where one print and space/skip operation has not been completed before the next print and space operation is initiated. If this instruction is not used, incorrect carriage spacing or skipping can result. The following sample program segment illustrates how these instructions might be coded:

```
PRBUSY    BPBB    PRBUSY,
SPACE     CC      S
WRITE     W       NET PAY
          NSI
```

Frequently, there are other processing steps that can be performed while waiting for the carriage to complete the last print/space/skip function. The *Branch if Printer Busy* instruction could have branched to perform these other functions.

This precaution applies only to systems having buffered printers.

Word Marks. Word marks are not affected.

Timing.

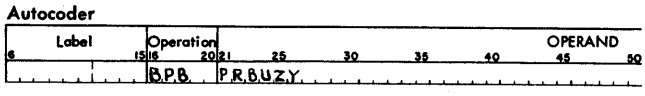
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Pbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the printer-busy indicator to determine if the printer is occupied with some other operation. If it is, branch to another series of instructions beginning at the instruction labeled PRBUZY (0486) while waiting for the printer to become available. If the printer is *not* busy, execute the NSI (Figure D-76).



Assembled Instruction: B 486 P

Figure D-76. Branch If Printer Busy

Branch if Channel 9

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BC9	B	III	9

Function. If the channel-9 position of the carriage-control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Timing.

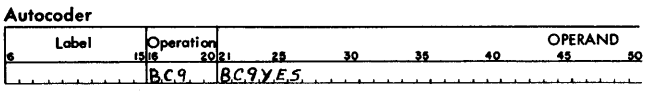
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	9bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test to determine if the carriage-control tape (forms) channel 9 has been sensed. Branch to a subroutine called BC9YES (0784) if the test is positive (Figure D-77).



Assembled Instruction: B 784 9

Figure D-77. Branch on Printer Carriage Channel 9

Branch if Channel 12

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BCV	B	III	@

Function. If the channel 12 (forms overflow) position of the carriage-control tape has been sensed, this instruction will cause a branch to the address specified by the I-address. This indicator is reset by the branch test, or by a channel-1 punch in the carriage-control tape.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	@bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to a subroutine labeled OVFLHD (0659) if the hole in channel 12 of the carriage-control tape has been sensed, indicating that the present form is filled (Figure D-78).

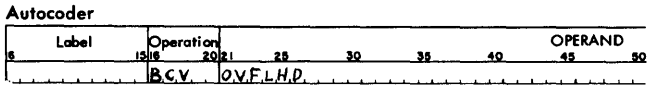


Figure D-78. Branch on Printer Carriage Forms Overflow

Control Carriage

Instruction Format.

Mnemonic	*Op Code*	*d-character*
CC	F	d

Function. This instruction causes the carriage to move as specified by the d-character (Figure D-79). If the d-character is:

1. a digit, an immediate skip to the specified channel in the carriage tape occurs.
2. an alphabetic character containing a 12-zone, a skip to the specified channel in the carriage tape occurs after the next line is printed.
3. an alphabetic character containing an 11-zone, an immediate space operation, as specified by the digit portion of the character, occurs.
4. an alphabetic character containing a zero-zone, a space operation, as specified by the digit portion of the character, occurs after the next line is printed.

Note: There is no CARRIAGE CONTROL AND BRANCH (CCB) instruction for the 1440 system.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms + remaining form-movement time, if carriage is already in motion when this instruction is given. The total form-movement time depends on the specific carriage operation being performed.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

d	Immediate skip to	d	Skip after print to
1	Channel 1	A	Channel 1
2	Channel 2	B	Channel 2
3	Channel 3	C	Channel 3
4	Channel 4	D	Channel 4
5	Channel 5	E	Channel 5
6	Channel 6	F	Channel 6
7	Channel 7	G	Channel 7
8	Channel 8	H	Channel 8
9	Channel 9	I	Channel 9
0	Channel 10	?	Channel 10
#	Channel 11	•	Channel 11
@	Channel 12	□	Channel 12
d	**Immediate space**	**d**	**After print-space**
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

Figure D-79. Control Carriage d-Characters

Example. Skip to channel 1 after a print operation (Figure D-80).

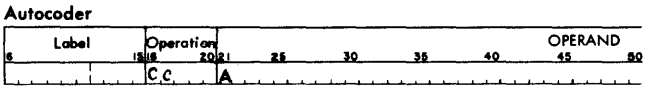


Figure D-80. Control Carriage

IBM 1445 Printing Speeds

The IBM 1445 Printer operates at a maximum speed of 190 lines per minute when the 56-character bar is installed.

Rated speeds for special feature character sets are:

Character Set	*Speed (LPM)*
42-Character Bar (Alphameric Type)	240
14-Character Bar (Standard Numeric Type)	525

IBM 7335 Magnetic Tape Unit

An additional storage medium with the advantage of compact record handling is now available to the 1440 system user by attaching the IBM 7335 Magnetic Tape Unit, Models 1 and 2 (Figure E-1) to his 1440 system. Refer to *IBM 7335 Magnetic Tape Unit*, Form A22-6789.

Figure E-2 shows the 7335 magnetic tape unit characteristics.

Data Flow

The IBM 7335 Magnetic Tape Unit functions as both an input and an output device. The 7335 transports the tape and accomplishes the actual reading and writing of information as directed by outside control from the system's stored program.

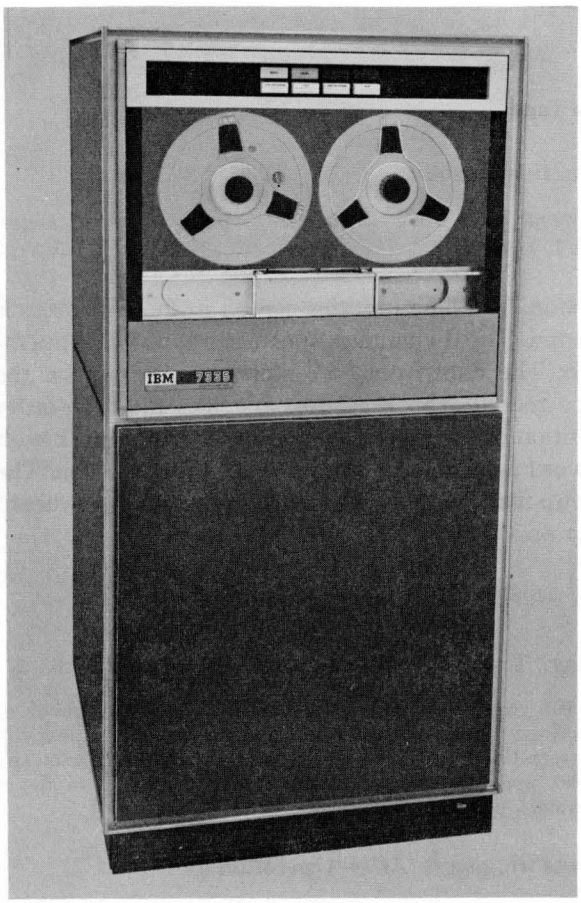


Figure E-1. IBM 7335 Magnetic Tape Unit (Model 1)

Data Rate	20,000 characters per second (CPS)
Bit Density	556 per inch (CPI)
Tape Speed	36 inches per second
Interrecord Gap	3/4 inch
Rewind (High Speed)	2.2 minutes

Figure E-2. IBM 7335 Magnetic Tape Unit Characteristics

Magnetic Tape Instructions

Instructions applying to the IBM 7335 cannot be successfully chained.

Read Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RT	M	%Un	BBB	R

Function. The tape unit specified in the A-address is started. The d-character specifies a tape read operation. The B-address specifies the high-order position of the tape read-in area of storage. The machine begins to read magnetic tape, and continues to read until either an inter-record gap in the tape record or a group mark with a word mark in core storage is sensed. The inter-record gap indicates the end of the tape record, and a group mark (code CBA 8421) is inserted in core storage at this point.

If the group mark with a word mark occurs before the inter-record gap is sensed, the transfer of data from tape stops, but tape movement continues until the inter-record gap is sensed.

Note: When a tape-mark (EOR) record is read, a group mark will be inserted in the second position of the tape read-in area.

Word Marks. Word marks are not affected.

Timing. $T = .0999 \text{ ms} + T_M$. (See *Magnetic-Tape Timing* for T_M time.)

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Group-mark + 1

Example. Read the record from tape unit 2 (labeled 2) into core storage. The high-order tape-record character is moved to INPUT (0419), the next character is moved to the next higher position (0420), etc., until transfer of data is stopped by an inter-record gap in the tape record, or a group mark with a word mark in core storage (Figure E-3).

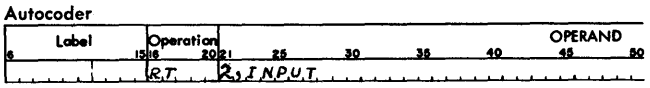


Figure E-3. Read Tape (Move Operation)

Read Tape with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RTW	L	%Un	BBB	R

Function. With the following exceptions this instruction is the same as the *Read Tape* operation. Word-separator characters (written with the WRITE TAPE WITH WORD MARKS instruction) are translated to word marks during the transmission into core-storage. As in all load-mode operations, word marks encountered in the B-field are cleared.

Note: When a tape-mark (EOR) record is read, a group mark is inserted in the second position of the tape read-in area.

Word Marks. A word-separator character (A841) read from tape causes a word mark to be associated with the next tape character transferred into core storage (Figure E-4).

Note. If a record has been written on tape by a WRITE TAPE WITH WORD MARKS instruction, it should be read back by a READ TAPE WITH WORD MARKS instruction so that word-separator characters are translated to word marks.

Tape Positions	A	B	C	D
Tape Code	82	A841	41	C4
1440 Core-Storage Locations	A	B	C	
1440 Meaning	0	5	4	
1440 Core-Storage Code	C82	41W	4	

Figure E-4. Word-Separator Character Handling during Read Tape with Word Marks Operation

Timing. $T = .0999 \text{ ms} + T_M$.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Group-mark + 1

Example. Read the record from tape unit 1 (labeled 1) into core storage, and insert word marks where word-separator characters exist in the tape record. The high-order character is moved to INREC1 (0518), the next character is moved to the next higher position (0519), etc., until the transfer of data is stopped by an inter-record gap in the tape record, or until a group mark with a word mark is sensed in 1440 core storage (Figure E-5).

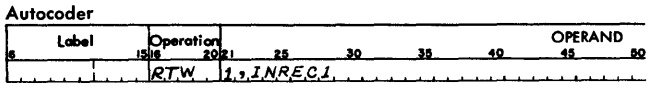


Figure E-5. Read Tape with Word Marks (Load Operation)

Write Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WT	M	%Un	BBB	W

Function. The tape unit designated in the A-address is started. The d-character specifies a tape write operation. The data from core storage is written on the tape record. The B-address specifies the high-order position of the record in storage. A group mark with a word mark in core storage stops the operation. The group mark with a word mark causes an inter-record gap on the tape.

Word Marks. Word marks are not affected.

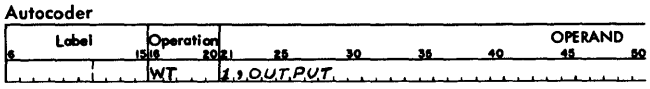
Timing. $T = .0999 \text{ ms} + T_M$.

Note. If a group mark with a word mark is the first character of B-address, the tape-adapter unit and the tape unit hangs up. The condition can be reset by pressing the start-reset key if the tape-select switch on the system console is in the N (normal) position.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Group-mark + 1

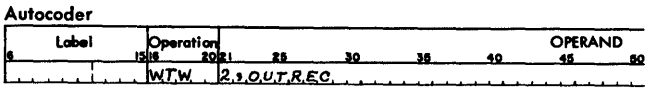
Example. Transfer the contents of core storage to tape unit 1 (labeled 1), starting at the location labeled OUTPUT (0525) and ending at the location of the first group mark with a word mark (Figure E-6).



Assembled Instruction: M %U1 525 W

Figure E-6. Write Tape (Move Operation)

Example. Transfer the contents of core storage to tape unit 2 (labeled 2). Insert a word-separator character where word marks exist in core storage, beginning at OUTREC (0696) and ending at the first group mark with a word mark in core storage (Figure E-8).



Assembled Instruction: L %U2 696 W

Figure E-8. Write Tape with Word Marks

Write Tape with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WTW	L	%Un	BBB	W

Function. This is the same as the write tape operation except that the WRITE TAPE WITH WORD MARKS instruction affects word marks in core storage.

Word Marks. A word mark associated with any position in core storage causes a word-separator character (A841) to be written automatically on tape, one character ahead of that which contained the word mark. Thus, word marks are translated to word-separator characters for tape storage (Figure E-7).

Timing. T = .0999 ms + T_M.

Note. Load operations must be used when word marks are needed for identification in tape storage. If tape is written by a WRITE TAPE WITH WORD MARKS instruction, it must be read back by a READ TAPE WITH WORD MARKS instruction to insure proper translation between the tape and core storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Group-mark + 1

1440 Core-Storage Locations	A	B	C	
1440 Core-Storage Code	C82	41W	4	
1440 Meaning	0	5	4	
Tape Positions	A	B	C	D
Tape Code	82	A841	41	C4
Tape meaning	0	Word Separator	5	4

Figure E-7. Word-Separator Character Handling during Write Tape with Word Marks Operation

Backspace Tape Record

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
BSP	U	%Un	B

Function. The tape unit specified in the A-address backspaces over one tape record. The first interrecord gap (IRG) encountered stops the backspace operation specified by the d-character, B.

Word Marks. Word marks are not affected.

Timing. T = .0666 ms + T_M.*

*Backspace after Read operation: (428 + .050 N) ms

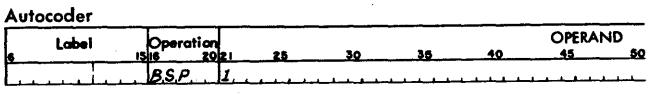
Backspace after Write operation: (435 + .050 N) ms

Note: The system is interlocked for the duration of tape movement for any instructions that have a percent sign (%) in the hundreds position of the A-address. Other functions are not affected.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Bbb

Example. Backspace tape unit 1 (labeled 1) until an IRG is sensed (Figure E-9).



Assembled Instruction: U %U1 B

Figure E-9. Backspace Tape Record

Skip and Blank Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
SKP	U	%Un	E

Function. The tape unit, designated by the A-address, spaces forward and erases approximately 4 inches of tape. The actual skip occurs when the next WRITE TAPE instruction is given. This instruction makes it possible to bypass defective tape areas.

Word Marks. Word marks are not affected.

Timing. T = .0666 ms. Processing can continue immediately after this operation. However, 110 ms must be added to the next WRITE TAPE instruction time.

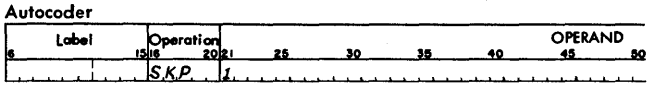
Notes. The SKIP AND BLANK TAPE instruction should be given immediately preceding a WRITE TAPE instruction for the tape unit specified by both instructions.

The system is interlocked for the duration of tape movement for any instructions that have a percent sign (%) in the hundreds position of the A-address. Other functions are not affected.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Ebb

Example. Erase tape on tape unit 1 (labeled 1) when the next write operation is ordered for that unit (Figure E-10).



Assembled Instruction: U %U1 E

Figure E-10. Skip and Blank Tape

Write Tape Mark

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
WTM	U	%Un	M

Function. This instruction causes a tape mark character (C8421) to be recorded immediately following the last record on tape. When the tape mark is read back from a tape, the end-of-reel indicator is turned on. This signals the system program that the end of a major group of records has been reached (end-of-file) or the end of utilized tape has been reached.

Word Marks. Word marks are not affected.

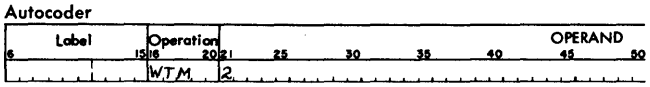
Timing. T = .0666 ms + T_M.

Note: The system is interlocked for the duration of tape movement for any instructions that have a percent sign (%) in the hundreds position of the A-address. Other functions are not affected.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Mbb

Example. Insert a tape mark on the tape in tape unit 2 (labeled 2); Figure E-11.



Assembled Instruction: U %U2 M

Figure E-11. Write Tape Mark

Diagnostic Read

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%Bn	A
CU	U	%Un	A

Function. This instruction causes the tape unit specified in the A-address to reposition its tape to the next inter-record gap (IRG) without transmitting any data to core storage. If the tape record contains a first character tape mark, the end-of-file (EOF) indicator is turned on.

This instruction is useful in skipping records or files on tape. The system is free to proceed with internal processing during the tape movement.

The tape operations are interlocked until the check character of the record being skipped is sensed.

Word Marks. Word marks are not affected.

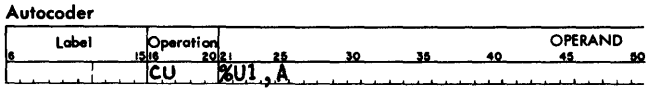
Timing. T = .0666 ms + T_M.

Note: The system is interlocked for the duration of tape movement for any instructions that have a percent sign (%) in the hundred's position of the A-address. Other functions are not affected.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%2n	Abb

Example. Read one tape record from tape drive number 1 but do not enter the tape data into core storage. If the record has a first-character tape mark, turn on the end-of-file indicator (Figure E-12).



Assembled Instruction: U %U1 A

Figure E-12. Diagnostic Tape Read

Rewind Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
RWD	U	%Un	R

Function. This instruction is usually given after an end-of-reel condition, and causes the selected tape unit to rewind its tape. When the operation is initiated, the tape unit is, in effect, disconnected from the system.

Word Marks. Word marks are not affected.

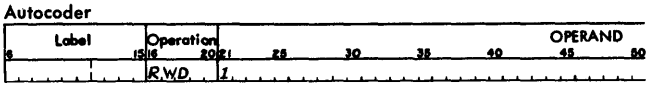
Timing. T = .0666 ms. Rewind time is 13.3 minutes, but it is not calculated with program time. Processing can continue immediately after this instruction is interpreted.

Note. Processing unit not interlocked during tape-movement time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Rbb

Example. Rewind the tape in tape unit 1 (labeled 1); Figure E-13.



Assembled Instruction: U %U1 R

Figure E-13. Rewind Tape

Rewind and Unload

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
RWU	U	%Un	U

Function. This instruction causes the tape unit specified in the A-address to rewind its tape. At the end of the rewind, the tape is out of the vacuum columns, and the reading mechanism is disengaged. The unit is effectively disconnected from the system, and is not available again until the operator restores it to a ready status.

Word Marks. Word marks are not affected.

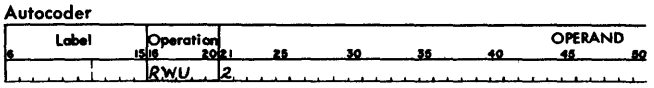
Timing. T = .0666 ms. Rewind time is 2.2 minutes, but it is not calculated with program time. Processing can continue immediately after this instruction is interpreted.

Note. Processing unit not interlocked during tape-movement time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%4n	Ubb

Example. Rewind the tape in tape unit 2 (labeled 2), and make it unavailable to the stored program (Figure E-14).



Assembled Instruction: U %U2 U

Figure E-14. Rewind Tape and Unload

Branch if End of Reel

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BEF	B	III	K

Function. The end-of-reel indicator (EOR) turns on in the system processing unit if a tape mark is read by the system or if a *reflective spot* is sensed during a write tape operation. This instruction tests the indicator and causes an automatic branch to the I-address if the indicator is on. If it is off, the program continues normally.

Word Marks. Word marks are not affected.

Timing.
No branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Note: This instruction must be executed immediately after testing for a tape error, following a tape read or tape write operation (particular tape unit still in select and ready status) to ensure correct results and reset the EOR indicator OFF, if it is on. If another tape unit is selected before a BRANCH IF END-OF-REEL IDNICATOR ON instruction is executed, the indicator remains ON and a false EOR test specifying the wrong tape unit results.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	Kbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the tape unit just used for an end-of-reel condition. If there is an EOR condition, branch to TAPER1 (0685) for the next instruction. If no EOR exists, continue the program with the next sequential instruction (Figure E-15).

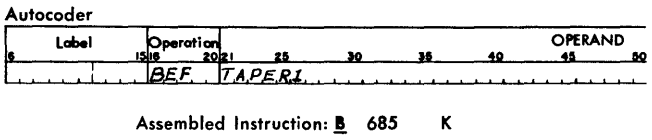


Figure E-15. Branch If End of Reel

system and the tape light on the console glows red. This instruction tests the error indicator, and branches to the I-address for the next instruction if the indicator is on. If it is off, the program continues with the next sequential instruction.

Word Marks. Word marks are not affected.

Timing.
No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Note.
The tape read-in area must be cleared if an error occurs because the error may have created a group mark with a word mark somewhere in the read-in area. This false bit configuration will cause all subsequent tape read operations to terminate too early.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg*
No Branch	NSI	BI	Lbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Read a tape record from the tape unit 1 (labeled 1) into core-storage area labeled TAPEIN (0629) and test for a tape error. If there is an error, branch to TAPER2 (0539) for the next instruction. If there is no error, continue processing with the next sequential instruction (Figure E-16).

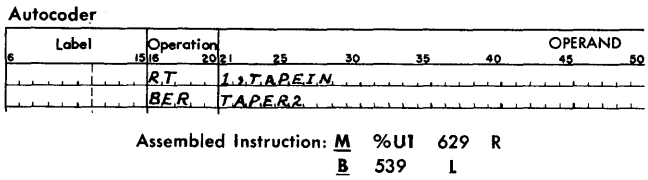


Figure E-16. Branch If Tape Error

Branch if Tape Error

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BER	B	III	L

Function. If an error occurs in transmission between a tape unit and the system during a tape read or tape write operation, an error indicator turns on in the

Read Binary Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RTB	M	%Bn	BBB	R

Function. A tape record written in binary form is read into core storage, beginning at the location specified by the B-address and ending at an inter-record gap between tape records or a group mark with a word mark in core storage. The A-address indicates the tape unit selected, and signals the column-binary tape operation. The d-character (R) specifies a read operation.

Word Marks. Word marks are not affected.

Timing. T = .0999 ms + T_M.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%2n	Group-mark + 1

Example. Read the binary tape record from the tape unit labeled 1 into the area of core storage labeled BTPIN (2080) and ending at the group mark with a word mark sensed in core storage or at the first inter-record gap encountered in the tape record (Figure E-17).

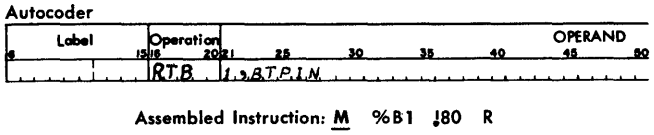


Figure E-17. Read Binary Tape

Write Binary Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WTB	M	%Bn	BBB	W

Function. This instruction writes a tape record in the odd-parity mode. The A-address specifies the tape unit to be selected, and signals that this is a binary-tape operation. The B-address specifies the high-order position of the tape record in core storage. The d-character indicates a tape-write operation. Sensing a group mark with a word mark in core storage stops transmission from the system to the tape unit.

Word Marks. Word marks are not affected.

Timing. T = .0999 ms + T_M.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%2n	Group-mark + 1

Example. Write a tape record in the binary mode on the tape unit labeled 2, with the data stored in the area labeled BTPOUT (2001) and ending when a group mark with a word mark is sensed in core storage (Figure E-18).

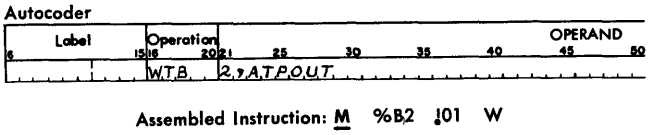


Figure E-18. Write Binary Tape

Magnetic-Tape Timing

The tape units attached to the 1440 system are under the control of a tape-adapter unit (TAU). This unit controls the operation of only one tape unit at a time. If the one tape unit is busy, the other tape unit cannot be used until all operations on the one that is busy has been completed.

The following symbols and figures are used in the 7335 timing formulas:

Character Rate of the 7335 at 556 Characters per Inch: .050 ms.

N: the number of characters in the record.

Start Time: the time necessary for the tape unit to accelerate to operating speed.

Stop Time: the time necessary for the tape unit to decelerate and stop.

Record Check Time: the time it takes to read or write the check character. This time is based on the read-write head gap (the distance that separates the read and write heads) and the time it takes a single character written on tape to travel from the write head to the read head.

Load Point Time. When reading or writing from load point, a skip of 3.5 inches occurs prior to reading or writing a record and the start time is increased about 27 milliseconds.

Read Operation Timing

During a 7335 tape-read operation, the tape-adapter unit is interlocked 20.5 + .050N ms (Figure E-19). This includes:

10.3 ms – start time
9.8 ms – stop time
.4 ms – record check time
.050N ms – record time
20.5 + .050N ms

During the same read operation, the processing unit is interlocked for 10.4 + .050N ms (Figure F-19). This includes:

10.3 ms – start time
.1 ms – part of .4 ms record check time
.050N ms – record time
10.4 + .050N ms

Therefore, in a tape-read operation, processing can take place during 10.1 ms of stop time and record-check time. A tape-transmission-error condition can be recognized .3 ms after the processing interlock is released.

Write Operation Timing

During a 7335 tape-write operation, the tape-adapter unit is interlocked 20.5 + .050N ms (Figure E-19). This includes:

7.2 ms – start time
4.4 ms – stop time
8.7 ms – record check time
.050N ms – record time
20.3 + .050N ms

During the same write operation, the processing unit is interlocked for 7.2 + .050N ms (Figure E-19). This includes:

7.2 ms – start time
.050N ms – record time
7.2 + .050N ms

Therefore, in a tape-write operation, processing can take place during the 13.1 ms record check and stop time. A tape-transmission-error condition can be recognized 8.7 ms after the processing interlock is released. If the tape-transmission-error test is given *during* the 8.7 ms record check time, *the processing unit is interlocked until the error indicator is interrogated.* The difference between the reading record-check time of .4 ms and the writing record-check time of 8.7 ms is due to the read-write head gap time (8.3 ms).

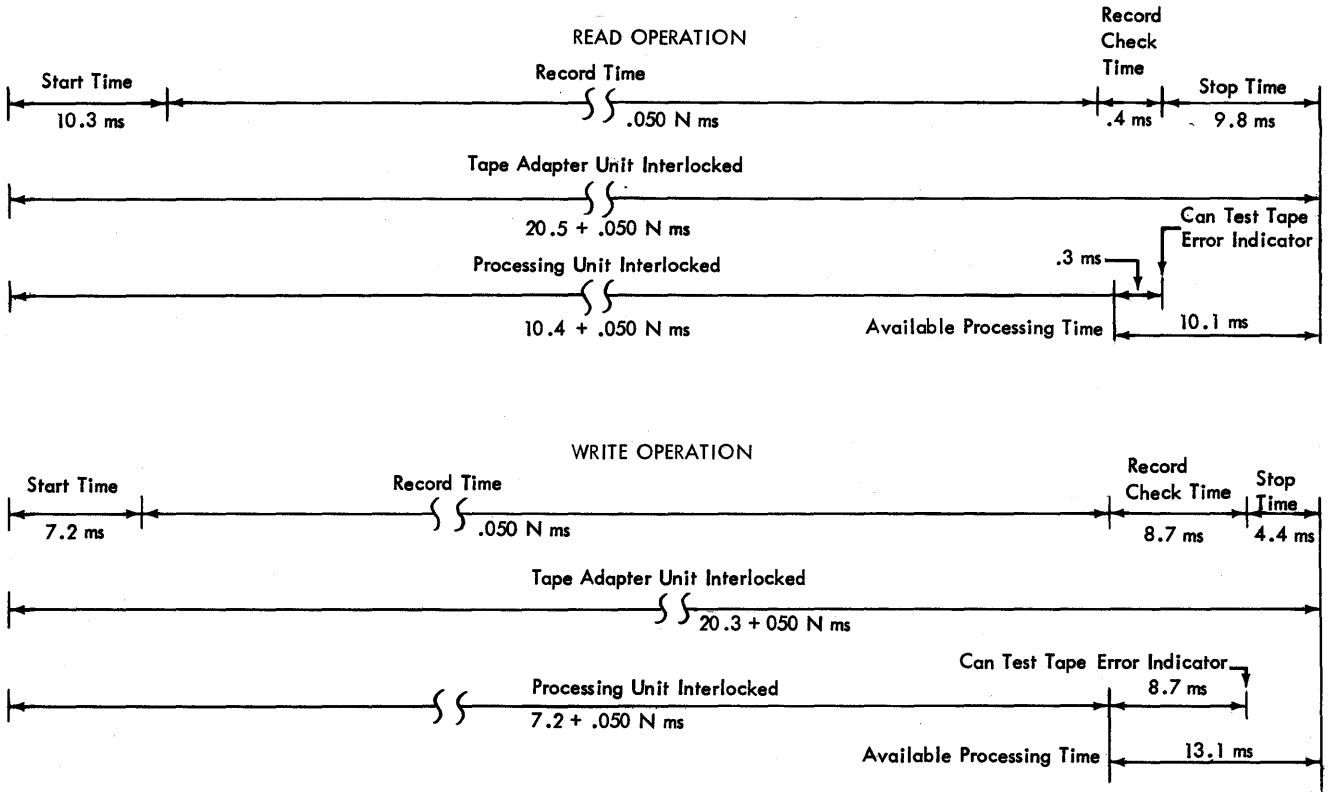


Figure E-19. IBM 7335 Read-Write Operation Timing

IBM 1011 Paper Tape Reader

The IBM 1011 Paper Tape Reader (Figure E-20) for the IBM 1440 Data Processing System is an input device controlled by stored programs in the same manner as other input/output equipment. Refer to *IBM 1011 Paper Tape Reader*, Form A26-5754.

Information punched in paper or Mylar* tape can be read by the IBM 1011 directly into any area of core storage. Any character punched in 5-track telegraphic, 8-track IBM, or many other paper-tape codes can be encoded into any valid 1440 character through the flexibility of control-panel wiring on the tape reader.

IBM 1011 Paper Tape Reader Instructions

Instructions applying to the 1011 cannot be successfully chained.

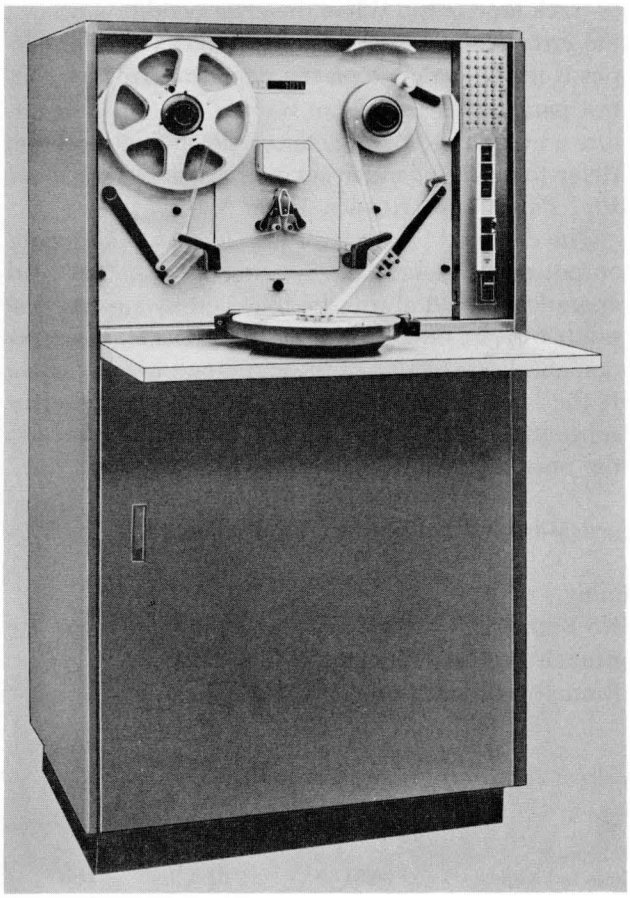


Figure E-20. IBM 1011 Paper Tape Reader

*Trademark of E. I. du Pont de Nemours & Co.

Read from Paper Tape

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%P1	BBB	R

Function. This instruction causes data to be read from the paper tape reader into core storage, beginning at the B-address.

The M op code specifies that the operation is to be performed in the *move* mode. When the M operation code is used, word marks are not transferred into core storage with the data read from the paper tape, and word marks in the core-storage paper-tape read-in areas are undisturbed. The A-address, %P1, is the code assigned to both the IBM 1011 Paper Tape Reader and the IBM 1012 Tape Punch.

The B-address specifies the core-storage position (high-order) that receives the first character of information from the paper-tape reader. The succeeding characters are read into the adjacent higher-numbered core-storage positions.

The d-character R specifies a read operation. The read operation ends either by detection of a group mark with a word mark in core storage (signifies the end of the read-in area), or by reading an EOR (end-of-record character) character punched in the tape.

Any paper-tape character can be used as an end-of-record character. Wiring the assigned end-of-record character decode-exit hub to the end-of-record IN hub terminates the paper-tape read operation and enters a group mark in core storage.

Note. If a group mark with a word mark in core storage is used to terminate the paper-tape-read operation, the character read into the A-register, when the group mark with a word mark is sensed, is lost.

Word Marks. Word marks are not affected.

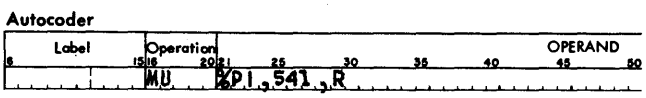
Timing. T = .0999 ms + record transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%71	B + message length + 1

Example. Read a record from paper-tape reader 1 into core storage. The high-order paper-tape character is moved to a storage location labeled READIN (0541); the next character is moved to the next higher position (0542), etc., until the transfer is terminated. The transfer is stopped either by detecting a group mark with a word mark in core-storage (signifying the limit of the read-in area), or by reading an End-

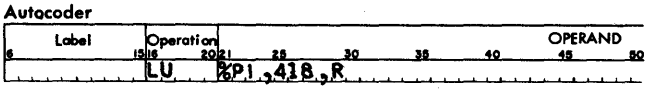
of-Record (EOR) character punched in the tape (Figure E-21). Word marks in the read-in area are not affected.



Assembled Instruction: M %P1 541 R

Figure E-21. Read from Paper Tape

read from the paper-tape reader causes a word mark to be associated with the next character inserted in core-storage from the 1011 (Figure E-22).



Assembled Instruction: L %P1 418 R

Figure E-22. Read from Paper-Tape Reader with Word Marks

Read from Paper Tape with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%P1	BBB	R

Function. This instruction is similar to the READ FROM PAPER TAPE instruction, except that word marks are removed from the paper-tape read-in area in core storage, and word-separator characters read from the paper-tape reader cause the insertion of a word mark in core storage with the next character read from the 1011. The L op code specifies that the operation is to be performed in the load mode.

Note: See *Read from Paper Tape* section.

Word Marks. Word marks are removed from the paper-tape read-in area in core storage, and word-separator characters read from the paper-tape reader cause a word mark to be associated with the next character read from the 1011.

Timing. T = .0999 ms + record transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%71	B + message length + 1

Example. Read a record from paper-tape reader 1 into core-storage. The high-order paper-tape character is moved to a storage location labeled RDPTWM (0418); the next character is moved to the next higher position (0419) until the read operation ends either by detecting a group mark with a word mark in storage, or by reading an End-of-Record (EOR) character punched in the tape. Any existing word marks in the read-in area are cleared. A word-separator character

Branch if Input/Output Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	1

Function. When a parity error is detected during a read operation, the PE hub provides an error output (8-track tape only). When the error condition occurs, the error character is suppressed and a special output is made available on the paper-tape-reader control panel. This signal can be used to either substitute a unique error character, or delete that position. Refer to CONTROL and SPECIAL PURPOSE hubs in *IBM 1011 Paper Tape Reader,* Form A26-5754.

The detected parity error also turns on the input/output error latch in the system. A paper-tape-read operation should always be followed by a BRANCH IF INPUT-OUTPUT INDICATOR ON instruction. This instruction checks the status of the input/output error latch. If the latch is ON, the system branches to the error subroutine. If the latch is OFF, the program goes to the next sequential instruction.

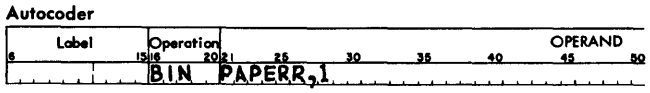
Word Marks. Word marks are not affected.

Timing.
No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch (no indexing)	NSI	BI	1bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to an error subroutine labeled PAPERR (0661) if the input/output indicator signals a paper-tape read error (Figure E-23).



Assembled Instruction: B 661 1

Figure E-23. Branch If Input/Output Indicator On

Branch if Paper Tape Reader Ready

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	2

Function. This instruction checks the status of the tape-reader-ready indicator. If the paper-tape reader is not ready, when tested, the program goes to the next sequential instruction. If the paper-tape reader is ready, when tested, the program branches to the subroutine that begins at the core-storage position specified by the instruction I-address.

Word Marks. Word marks are not affected.

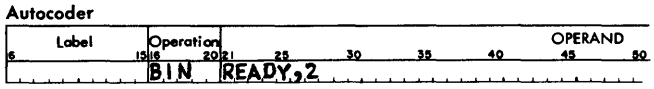
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	2bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a paper-tape read routine labeled READY (0767) if the 1011 is ready to read (Figure E-24).



Assembled Instruction: B 767 2

Figure E-24. Branch If Paper Tape Reader Ready

IBM 1012 Tape Punch

The IBM 1012 Tape Punch (Figure E-25) attached to the IBM 1440 Data Processing System is an output device controlled by stored programs in the same manner as other input/output equipment. Refer to *IBM 1012 Tape Punch*, Form A26-5776.

The IBM 1012 Tape Punch operates at the rate of 150 tape characters per second, using 5-, 6-, 7-, or 8-track paper or Mylar tape, supplied from a reel. Data, stored in the core-storage area of 1440 system and ready to be punched, is converted to the appropriate tape code by using a translation program that includes the appropriate stored code table.

IBM 1012 Tape Punch Instructions

Instructions applying to the 1012 cannot be successfully chained.

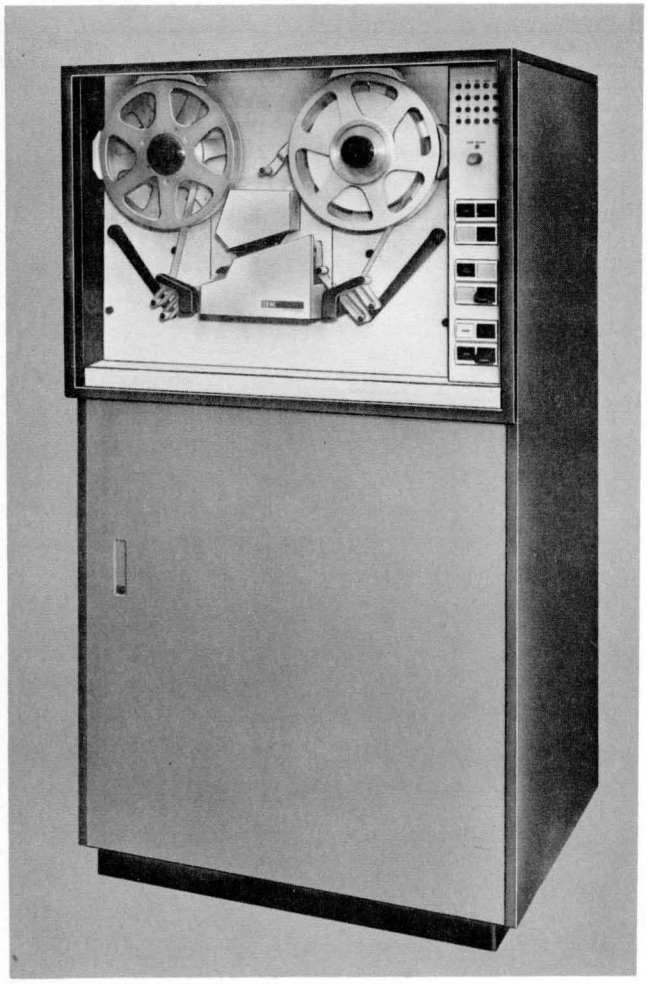


Figure E-25. IBM 1012 Tape Punch

Write on Tape Punch

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%P1	BBB	W

Function. This instruction causes one vertical tape column to be punched. The M op code specifies an operation in the *move* mode. When the M operation code is used, word marks are not transferred from core storage to the tape punch. The A-address, %P1, is the code assigned to both the IBM 1011 Paper Tape Reader and the IBM 1012 Tape Punch.

The B-address specifies the first core-storage position (high-order) of the three-position field. The three-position field contains the total bit configuration that will be punched in one vertical column in the tape. The d-character W specifies a write operation.

Word Marks. Word marks are not affected.

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%71	B + 3

Example. Read a record from core storage identified by the label WRITE (0551), and punch into paper tape. (Figure E-26).

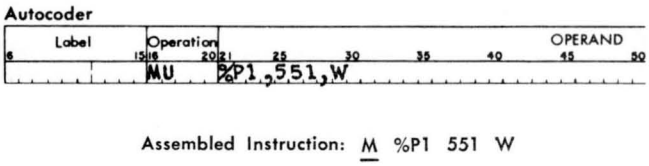


Figure E-26. Write on Paper Tape

Tape-Punch Read-Back Check

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%P1	BBB	R

Function. This instruction reads one vertical tape column when a read-back check is desired on the characters punched in the tape. The M op code specifies an operation in the move mode. When the M operation code is used, word marks in core storage are not removed or affected. The A-address, %P1, is the code assigned to both the IBM 1011 Paper Tape Reader and the IBM 1012 Tape Punch.

The B-address specifies the first core-storage position (high-order) of the three-position field. The three-position field contains the total bit configuration of the character being read from the tape at the reading station. The d-character R specifies a read operation.

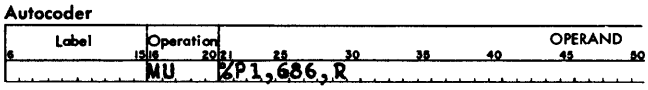
Word Marks. Word marks are not affected.

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%71	B + 3

Example. Read one vertical paper-tape column to check the character punched into the tape. The label RDBKCK (0686) specifies the first core-storage (high-order) of the 3-position field (Figure E-27).



Assembled Instruction: M %P1 686 R

Figure E-27. Tape-Punch Read-Back Check

Backspace Tape

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	A

Function. This instruction moves the tape backward one vertical column. The program then goes to the next sequential instruction.

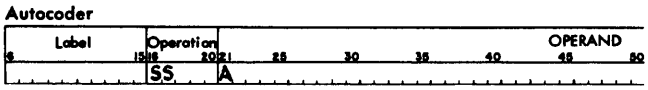
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Abb	Abb

Example. Move paper tape backwards one vertical column (Figure E-28).



Assembled Instruction: K A

Figure E-28. Backspace (Paper) Tape

Backspace Tape and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	A

Function. This instruction is similar to the BACKSPACE TAPE instruction, except that the location of the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

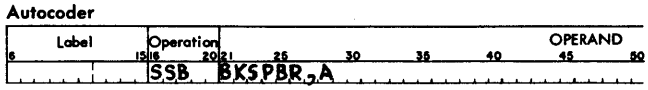
Timing.

Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Move the paper tape backwards one vertical column, and branch to a core-storage location labeled BKSPBR (0777) for the next instruction to be executed (Figure E-29).



Assembled Instruction: K 777 A

Figure E-29. Backspace (Paper) Tape and Branch

Branch if in Backspace Operation

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	1

Function. This instruction checks to see whether the tape punch is executing a backspace operation. The backspace operation condition is present from the time the tape punch is signaled to execute the backspace operation until the operation is completed.

If a backspace operation is in progress, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. If a backspace operation is not in progress, when checked, the program goes to the next sequential instruction.

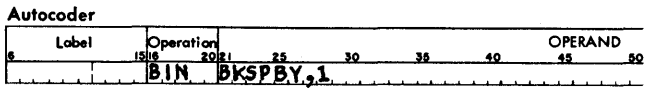
Word Marks. Word marks are not affected.

Timing.
No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	1bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test an indicator to determine if the 1012 punch is presently performing a tape-backspace operation. If it is, branch to a core storage location labeled BKSPBY (0525) for the next instruction (Figure E-30).



Assembled Instruction: B 525 1

Figure 30. Branch If in Backspace Operation

Branch if Tape Punch Ready

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	2

Function. This instruction checks to see whether the tape punch is in a ready condition. The tape punch is considered in a ready condition when each of the following conditions is satisfied:

1. Tape properly loaded
2. Tape tension is normal
3. Electrical power is supplied
4. Start switch has been pressed.

If the tape punch is in a ready condition, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. If the tape punch is not in a ready condition, when checked, the program goes to the next sequential instruction.

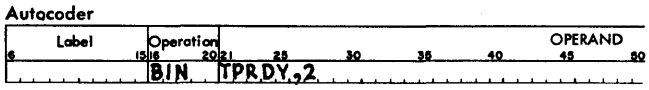
Word Marks. Word marks are not affected.

Timing.
No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	2bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a subroutine labeled TPRDY (0818) if the 1012 tape punch is ready (Figure E-31).



Assembled Instruction: B 818 2

Figure E-31. Branch If Tape Punch Ready

Branch if Tape Punch Not Ready to Accept Data

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	3

Function. This instruction checks to see whether the tape punch is in the correct mechanical position to accept data. The IBM 1012 Tape Punch punches tape at the speed of 150 characters per second (6.6 milliseconds between characters). The data can be accepted during a 1.5-millisecond (ms) portion of the 6.6 ms time interval between characters.

If the tape punch is not in correct mechanical position to accept data, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. If the tape punch is in correct mechanical position to accept data, the program goes to the next sequential instruction.

Word Marks. Word marks are not affected.

Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	3bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a core-storage location labeled NOPCH (0343) if the 1012 tape punch is not ready to accept data (Figure E-32).

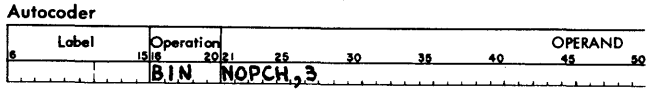


Figure E-32. Branch If the Tape Punch Is Not Ready to Accept Data

Branch if Tape Punch Not Ready to Read

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	4

Function. This instruction checks to see whether the reading portion of a punch cycle has been reached.

The IBM 1012 Tape Punch punches tape at the speed of 150 characters per second (6.6 milliseconds between characters). A 1.8-millisecond (ms) portion of the 6.6 ms time interval between characters is reserved for the actual read operation.

If the reading portion of a punch cycle has not been reached, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. If the reading portion of a punch cycle has been reached, the program goes to the next sequential instruction.

Word Marks. Word marks are not affected.

Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	4bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a subroutine at a core-storage location labeled NORD (0381) if the 1012 tape punch is not ready to read (Figure E-33).

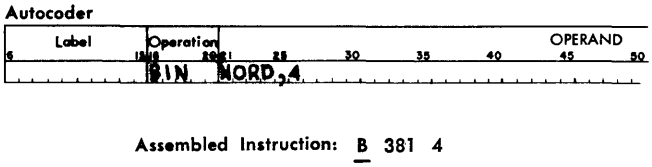


Figure E-33. Branch If the Tape Punch Is Not Ready to Read

Branch if Tape Punch Overextended

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	5

Function. This instruction checks to see whether a previous punch (or read) operation occurred within the allotted time during the last possible punch (or read) cycle.

If a punch (or read) operation did not occur within the allotted time, it may indicate a machine malfunction, and the tape punch can be stopped through programming. This condition:

1. Always exists when the tape punch is idle
2. Exists until a punch (or read) operation starts
3. Never exists when the tape punch is punching (or reading) at its maximum speed.

If a punch (or read) operation did not occur within the allotted time, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. If a punch (or read) operation did occur within the allotted time, when checked, the program goes to the next sequential instruction.

Word Marks. Word marks are not affected.

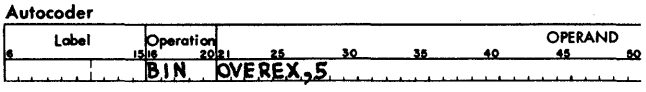
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	5bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a subroutine designed to handle a case of tape-punch overextension. This subroutine might be located at core-storage location 0820, and have the label OVEREX (Figure E-34).



Assembled Instruction: B 820 5

Figure E-34. Branch If Tape Punch Overextension

Branch if Supply Reel Low or Chad Box Full

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	6

Function. This instruction checks to see whether either an end-of-reel (supply reel low) condition exists or the chad box is full or not in position. Additional punching can occur after the supply-reel-low condition occurs, but the amount of additional punching depends upon the length of the records being punched.

If the supply reel is low, or the chad box is full, or not in position, when checked, the program branches to the subroutine that starts at the I-address specified in the instruction. Otherwise, the program goes to the next sequential instruction.

Word Marks. Word marks are not affected.

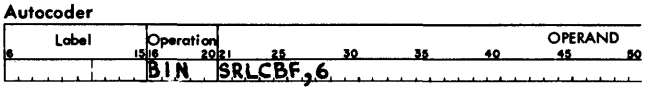
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	6bb
Branch (no indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Branch to a subroutine which can set up conditions to allow limited additional punching, if the chad-box is nearly full or the supply reel has approached its end. If either of these conditions arise, branch to a core-storage location labeled SRLCBF (0904). (Figure E-35).



Assembled Instruction: B 904 6

Figure E-35. Branch If Supply Reel Low Or Chad Box Full

IBM 1012 Programming Concepts

The basic logic of programmed translation is based on a programming device known as *character selection.* Character selection uses a stored table to translate the system characters to the pattern required by the IBM 1012 Tape Punch. This method of translation makes a control panel unnecessary.

A stored-program routine controls the over-all operation of the 1012. PTAPE (punch tape), a macro instruction, will be provided in the Autocoder library of routines. The operand of the PTAPE command is the symbolic name of the output area to be punched. A second operand is used to designate whether the routine is to punch standard Teletype* (5-track) code or IBM standard (8-track) code. Therefore, when punching is desired, the programmer need merely write PTAPE with the appropriate operand(s). The autocoder Processor then generates the necessary instructions to punch the tape.

The programs used for 6-and 7-track operation are basically the same as those used for 5- and 8-track operation. The major difference is in the stored table used for translating the system BCD codes into punch codes.

Once the theory of operation and the 5-and 8-track programs are understood, altering the program and the stored table is a simple task.

The decoding routine is within the 6.6-millisecond time interval between characters in the record that are to be punched. No attempt is made to overlap this time with the user's program. A special test is made for the 5-track tape to automatically generate mode-change characters where appropriate.

A brief description of the theory of operation is given to aid the user in understanding the over-all operation of the IBM 1012 Tape Punch. The examples used are not necessarily the only methods of programming the various operations. The examples are used primarily as an aid in simplifying the explanations. The methods, constants, and stored code tables used in the following explanations do not necessarily represent these items as they would appear in the macro instruction PTAPE.

The theory of operations (8-track operation) is explained in the following order:

1. Move character from record into test location
2. Decode to a table address
3. Get table argument for punching
4. Punch character in tape
5. Read-check
6. End-of-Record routine
7. Automatic error correction
8. End-of-Reel routine
9. Five-track tape operation.

*Trademark of Teletype Corporation.

1. Move Character from Record into Test Location

The record to be punched in tape can be any length (within the limits of available core storage) and may be stored in any system core-storage location. When a new record is ready to be punched into tape, a three-position field (the record address), is initialized to the first (high-order) postion of the record. This field is moved to the A-address of a move instruction which, when executed, moves the first record character to a location in core storage where it can be analyzed. A character compare instruction then checks the character for an end-of-record indication. An equal compare causes a branch to the end-of-record routine; otherwise the program advances to the decode routine.

2. Decode to a Table Address

A table address (Figure E-36) is developed for every character in the record. After the character is moved from the record into another core-storage location, it is analyzed to develop the table address. This is accomplished in the following manner.

A constant is moved to the A-address of a MOVE instruction. The constant would be the starting location

	Table Address	B	A	8	4	2	1	Char	Card Code
	0- 2	X	X					&	12
*	3- 5								
*	6- 8								
	9-11	X	X	X		X	X	·	12-3-8
	12-14	X	X	X	X			¤	12-4-8
	15-17	X	X	X	X		X	[	12-5-8
	18-20	X	X	X	X	X		<	12-6-8
	21-23	X	X	X	X	X	X	‡	12-7-8
*	24-26								
*	27-29								
	30-32	X						–	11
*	33-35								
*	36-38								
	39-41	X		X		X	X	$	11-3-8
	42-44	X		X	X			*	11-4-8
	45-47	X		X	X		X	]	11-5-8
	48-50	X		X	X	X		;	11-6-8
	51-53	X		X	X	X	X	Δ	11-7-8
*	54-56								
*	57-59								
	60-62		X					⅋	2-8
*	63-65								
*	66-68								
	69-71		X	X		X	X	,	0-3-8
	72-74		X	X	X			%	0-4-8
	75-77		X	X	X		X	v	0-5-8
	78-80		X	X	X	X		\	0-6-8
	81-83		X	X	X	X	X	⇻	0-7-8
*	84-86								
*	87-89								
	90-92							BL	No Punch
*	93-95								
*	96-98								
	99-101			X		X	X	#	3-8
	102-104			X	X			@	4-8
	105-107			X	X		X	:	5-8
	108-110			X	X	X		>	6-8
	111-113			X	X	X	X	√‾	7-8
*	114-116								
*	117-119								

	Table Address	B	A	8	4	2	1	Char	Card Code
	120-122	X	X	X		X		?	12-0
	123-125	X	X				X	A	12-1
	126-128	X	X			X		B	12-2
	129-131	X	X			X	X	C	12-3
	132-134	X	X		X			D	12-4
	135-137	X	X		X		X	E	12-5
	138-140	X	X		X	X		F	12-6
	141-143	X	X		X	X	X	G	12-7
	144-146	X	X	X				H	12-8
	147-149	X	X	X			X	I	12-9
	150-152	X		X		X		!	11-0
	153-155	X					X	J	11-1
	156-158	X				X		K	11-2
	159-161	X				X	X	L	11-3
	162-164	X			X			M	11-4
	165-167	X			X		X	N	11-5
	168-170	X			X	X		O	11-6
	171-173	X			X	X	X	P	11-7
	174-176	X		X				Q	11-8
	177-179	X		X			X	R	11-9
	180-182		X	X		X		‡	0-2-8
	183-185		X				X	/	0-1
	186-188		X			X		S	0-2
	189-191		X			X	X	T	0-3
	192-194		X		X			U	0-4
	195-197		X		X		X	V	0-5
	198-200		X		X	X		W	0-6
	201-203		X		X	X	X	X	0-7
	204-206		X	X				Y	0-8
	207-209		X	X			X	Z	0-9
	210-212			X		X		0	0
	213-215						X	1	1
	216-218					X		2	2
	219-221					X	X	3	3
	222-224				X			4	4
	225-227				X		X	5	5
	228-230				X	X		6	6
	231-233				X	X	X	7	7
	234-236			X				8	8
	237-239			X			X	9	9

*Not Used

Figure E-36. Table Address Chart

of the table plus two. Assume the stored table starts in location 400. The constant used would be 402. The character to be punched is now analyzed to develop a table address. If the character contains a B-bit only, the constant 30 is added to the number (402) already in the A-address of the MOVE instruction. An A-bit, only, adds the constant 60, no zone-bits adds the constant 90; and if the character contains A- and B-bits, nothing is added. A further test determines whether the character is a special character or blank. The constant 120 is also added to the A-address of the MOVE instruction if the character is *not* a blank or special character. Furthermore, the sum of the digit bits in the character being analyzed is tripled and also added to the number already developed. As an example, assume the character B (BA2) is being analyzed. The presence of an A- and B-bit would add nothing; however, 120 would be added because B is not a special character. The 2 bit is tripled, adding 6 to the number. The table address developed would therefore be 402 plus 120, plus 6, or a total of 528.

Figure E-37 illustrates the arrangement of the characters in the table, and the bit pattern for punching that is contained in each 3-position character location. Assuming the starting location of the table is 400, the address 528 would direct the system to the low-order position of the 3-position location for the character B.

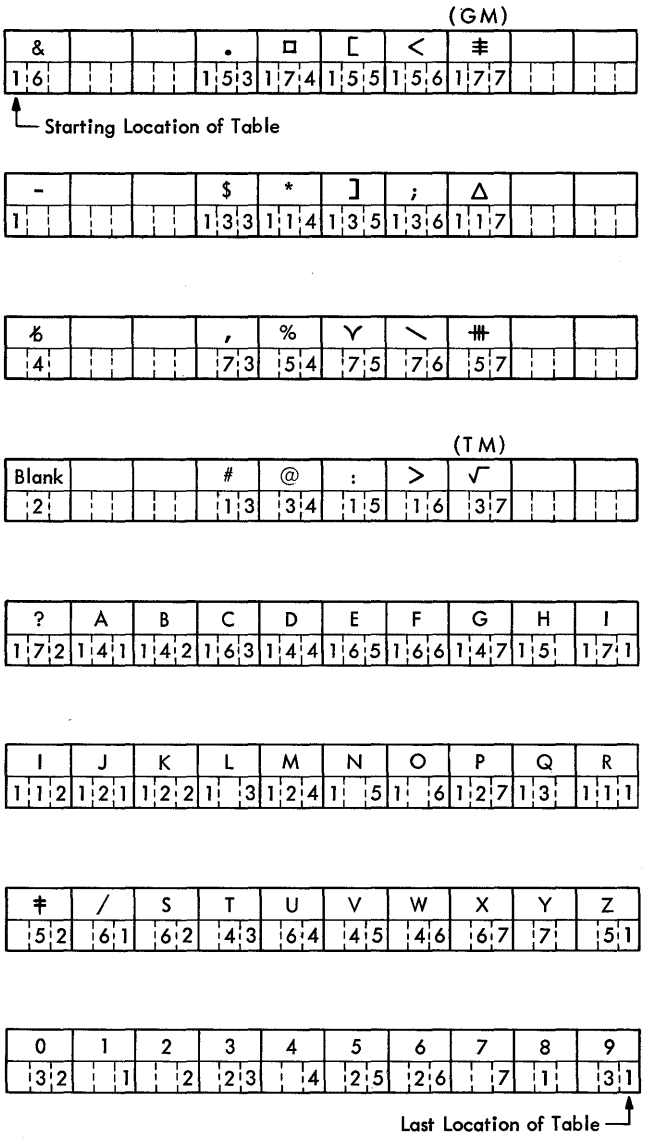


Figure E-37. IBM Eight-Track Code Table

3. Get Table Argument for Punching

The table address for the character to be punched is developed in the A-address location of a MOVE instruction. The MOVE instruction, when executed, moves the proper field from the table and places it in an unused 3-position storage location referred to as QD. The character is now ready for punching using the bit pattern in location QD.

4. Punch Character in Tape

The punch instruction M(%P1)(BBB)W is executed next, which causes data to be transferred to the 1012 for punching. The address in the punch instruction refers to the high-order position of the three-position field (QD) that contains the bit pattern to punch one vertical column in the tape.

Figure E-38 illustrates the bit pattern to tape-punching translation.

5. Read-Check

Because of the delay between punching and reading, the punching bit pattern for four characters must be retained for checking. Four consecutive 3-position fields are set aside for this purpose.

As noted in the punch routine, the bit pattern to be punched is moved from the stored table into location QD. By a method described later, the character bit pattern for the column of the tape that can be read during this punch cycle is put in location QA.

The read instruction M(%P1)(BBB)R causes data to be transferred from the reading station into three consecutive core-storage locations beginning in (BBB). The data transfer is in accordance to the pattern shown in Figure E-39. This is similar to the punch transfer but with direction reversed. The 3-position field into which data is read is compared to the contents of location QA. Thus, the punched tape is given a bit-by-bit

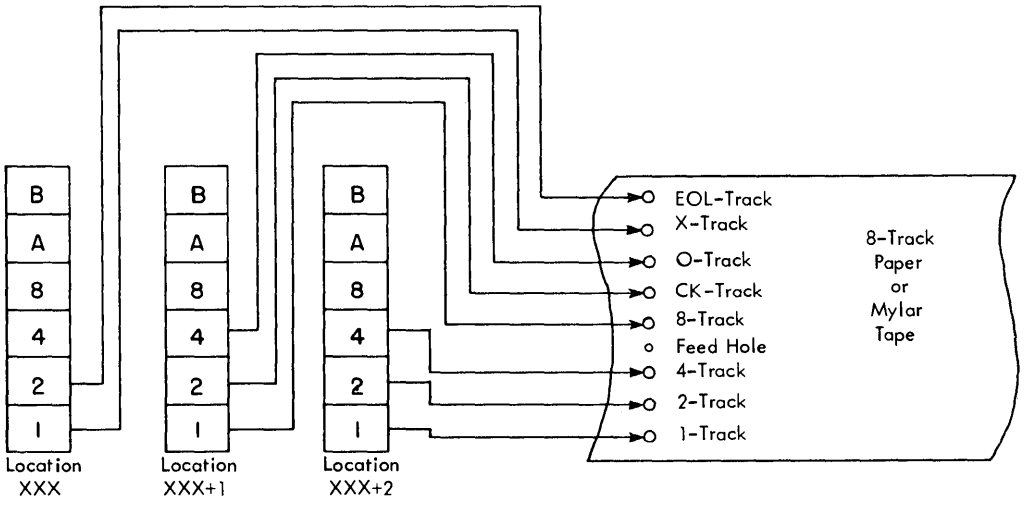


Figure E-38. Eight-Track Bit Pattern to Tape Punching Translation

comparison check. If punching and reading were correct, the BRANCH COMPARE instruction would not cause a branch to the error routine, but would continue to the next sequential instruction. After a valid compare, the contents of the four fields (QD, QC, QB, and QA) are shifted in preparation for the next read-check instruction. The program is then directed to the beginning to process the next character in the record.

6. End-of-Record Routine

Any one of many characters can be assigned as an EOR (End-of-Record) character. However, to simplify the explanation of this routine, assume that a group mark with a word mark is used. The main program routine checks every character to determine whether or not it is a group mark. If a group mark is detected, a further check determines whether a word mark is present. If a word mark is detected, this signifies an end-of-record character, and the system branches to the end-of-record routine. Figure E-40 illustrates the cycle-by-cycle operation of the punch and the relationship of the characters in locations QD, QC, QB, and QA dur-

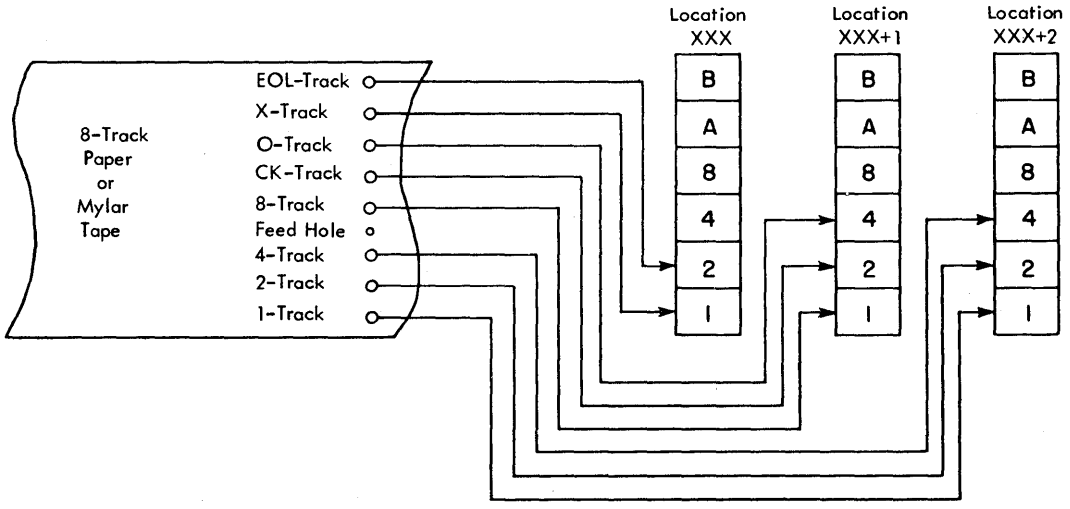


Figure E-39. Eight-Track Tape Punching to Bit-Pattern Translation

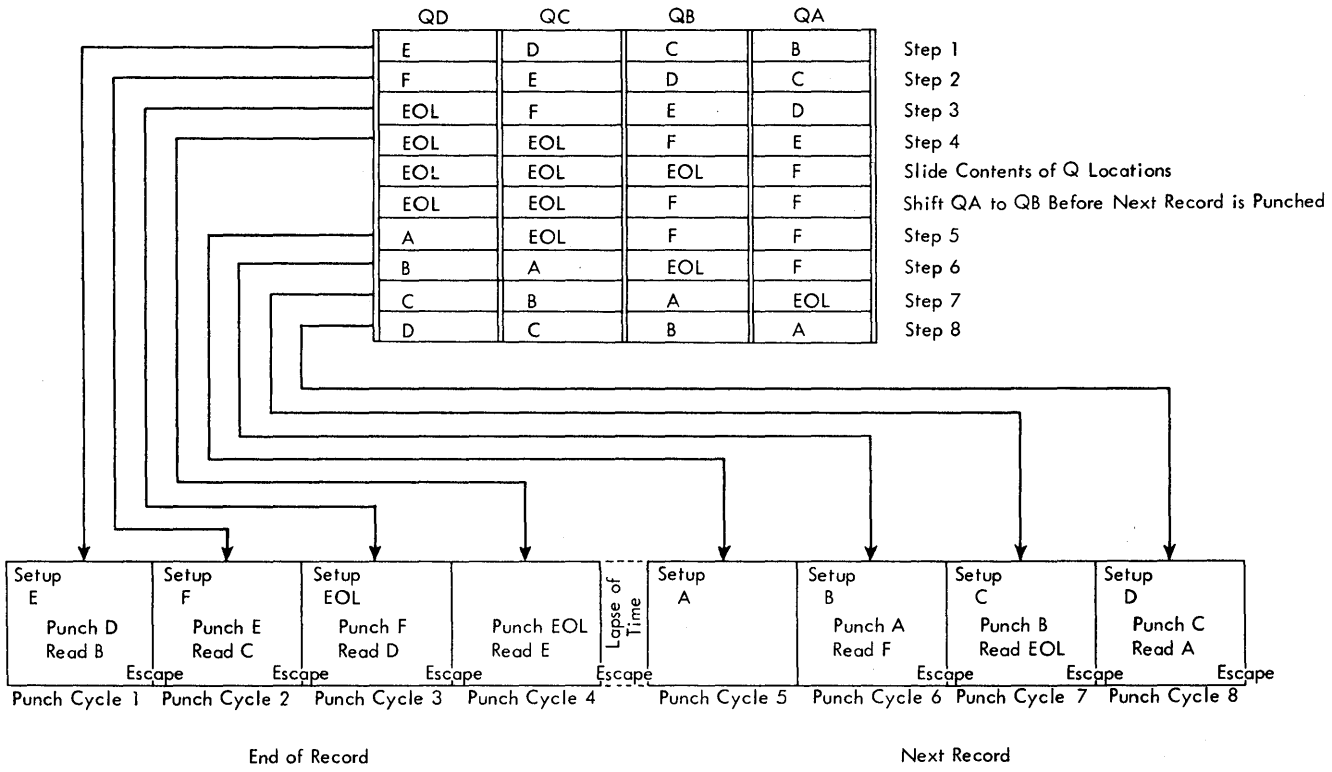


Figure E-40. End-of-Record Punch and Read Routine

ing an end-of-record routine. Assume that EOL (End-of-Line) is the tape representation for the end-of-record character and that the record to be punched consists of A, B, C, D, E, F, and end-of-record character.

The main program routine is in effect until an end-of-record character is detected. At the beginning of step 3 (Figure E-40), an EOR character is detected and the EOL constant (2, blank, blank) is moved into location QD. During punch cycle 3 (Figure E-40), the EOL code is set up in the punch magnets, the character F is punched, and the character D is read. Because EOL is the last character in the record to be punched, nothing is set up in the punch magnets during the punch cycle 4. Because EOL was set up during the previous cycle, it is punched in punch cycle 4. The tape is always advanced after punching takes place, which makes a read operation necessary during punch cycle 4. Without the extra read operation, character E would not have been read and checked. After EOL is punched and an escapement moves the tape, the contents of the Q locations are shifted to the right so that locations QD, QC, and QB contain EOL codes, and location QA contains the code for character F. At the completion of cycle 4, all characters in the record have been punched, including EOL. However, the character F and EOL remain to be read and checked. At this time, a new record is ready to be punched. The contents of QA is shifted into location QB before the first character in the next record is processed.

The first character of the next record (assume A) is moved into location QD. During punch cycle 5, character A is set up in the punch magnets. A read operation does not take place during cycle 5 because the character F is read and checked in the next punch cycle. During punch cycle 6, character B is set up, character A is punched, and character F is read and checked. During punch cycle 7, character C is set up, character B is punched, and EOL is read and checked. The main program routine continues until the next end-of-record character is detected in the record.

7. Automatic Error Correction

The system program branches to the automatic error-correction-program routine when an unequal compare condition is detected following a read-check instruction. When an error is detected, the tape is backspaced until the character in error is under the punch station. The tape is then moved in a forward direction and four delete codes (all tracks punched except EOL) are punched. The constant 177 is the bit pattern for punch-

ing all tracks in the tape, except EOL. The delete codes that are punched are also read back and checked. A valid EOL may be encountered when the delete codes are checked (a hole punched in all tracks). This condition is recognized by the program routine, but no action is taken.

When the tape is read later as an input to another system, this condition (even parity) signals an error. However, control-panel wiring or a program subroutine can be used to circumvent this condition. If an unequal compare (except EOL) is detected when the delete codes are checked, the error is corrected by branching to the normal error-correction routine.

After the delete codes have been punched, the characters that were deleted are now repunched. These characters are still in locations QD, QC, QB, and QA. These characters are punched starting with QA through QD.

8. End-of-Reel Routine

An end-of-reel test is made after every end-of-record program routine has been completed. After EOL has been punched (EOR), an appropriate end-of-file character (if required) can be punched in the tape. The last character in the record, EOL, and the end-of-file character (if required), are read back and checked before the end-of-reel program routine is completed.

After a new reel of tape has been installed, pressing the feed switch on the 1012 causes delete codes to be punched in the leader portion of the tape. Pressing the start key on the 1012 places the 1012 in a ready status. The last two delete codes punched in the leader portion of the tape are read and checked when the first record is punched in the new reel of tape.

9. Five-Track Tape Operation

Basically 5-track tape operation is similar to 8-track tape operation, with a limited number of exceptions. Only fifty-eight characters are punched when using 5-track tape, which necessitates the use of a different code table (Figure E-41). To properly identify a char-

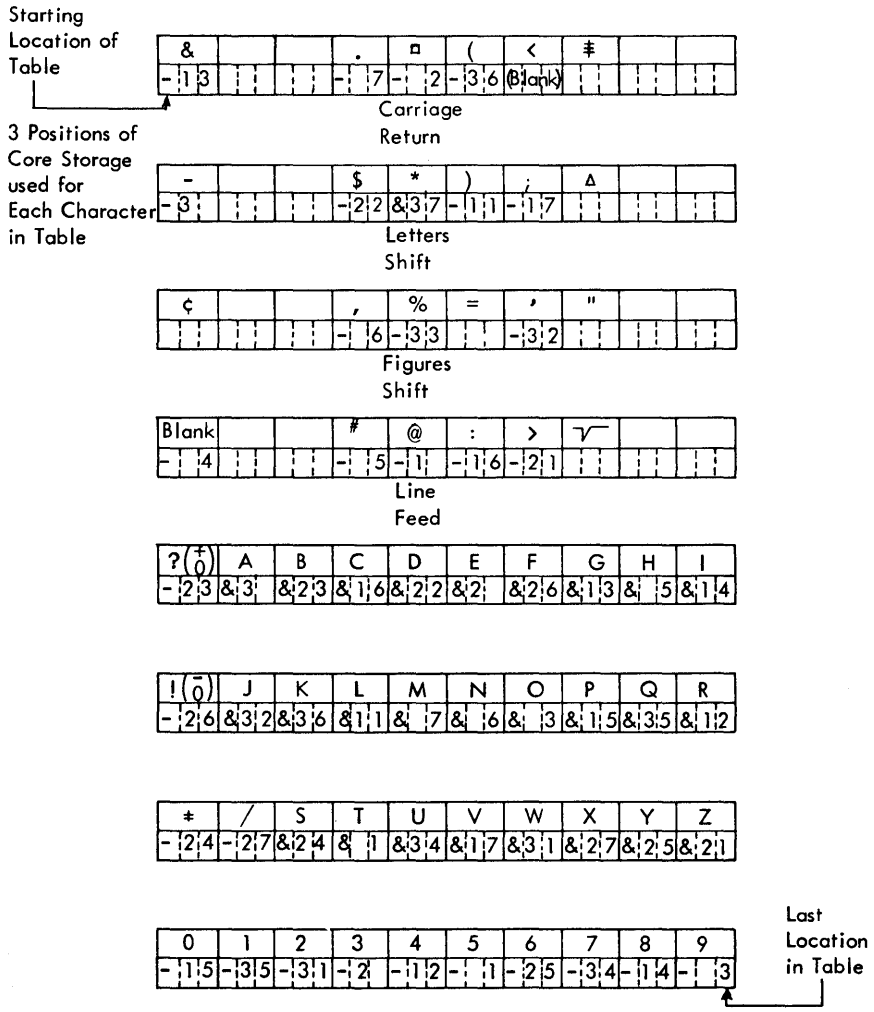


Figure E-41. Five-Track Code Table

acter punched in the tape, a figures-shift or letters-shift code must precede the punched character, if a mode change takes place.

Development of the stored-table address (see Figure E-36) remains the same as for 8-track operation (described in *Decode to a Table Address* section). The code number assigned to each character in the stored table differs from the code assigned to each character in the 8-track code table.

Figure E-41 illustrates the code assigned to each of the fifty-eight characters in the stored table. Only two positions of each 3-position field are used for the bit pattern of the code number. The high-order position of the 3-position field contains a Dash (B-bit), or Ampersand (A- and B-bits). The dash signifies a figures-shift character, and the ampersand signifies a letters-shift character. After the table address has been developed for a character to be punched, the contents of the 3-position table location are moved into location QD. The high-order position of QD is analyzed to determine whether the character is a figures-shift, or letters-shift character. After the mode of the character has been established, it is compared with the mode the 1012 is presently in. If a figures-shift character is to be punched and the 1012 is in a letters-shift mode, a change in mode is required. If a letters-shift character is to be punched and the 1012 is presently in figures-shift, a change in mode is required.

If a change in mode is required, one of two constants is moved into location QD. The constant is 33 if a change to figures-shift is required, or 37 if a change to letters-shift is required. These constants, when decoded (Figure E-42), punch either a figures-shift or letters-shift code in the tape. The character to be punched is then moved back into location QD and punched. If a mode change is not required, punching takes place from QD without moving the constants. When a mode change is executed, it is retained to identify what mode the 1012 is presently in.

When a READ-CHECK instruction is executed, the high-order position of location QA is not involved in the comparison. However, if an error is detected, the bits in the high-order position of QA indicate the shift of the character when it is repunched. Figure E-43 illustrates the 5-track tape punching to bit-pattern translation.

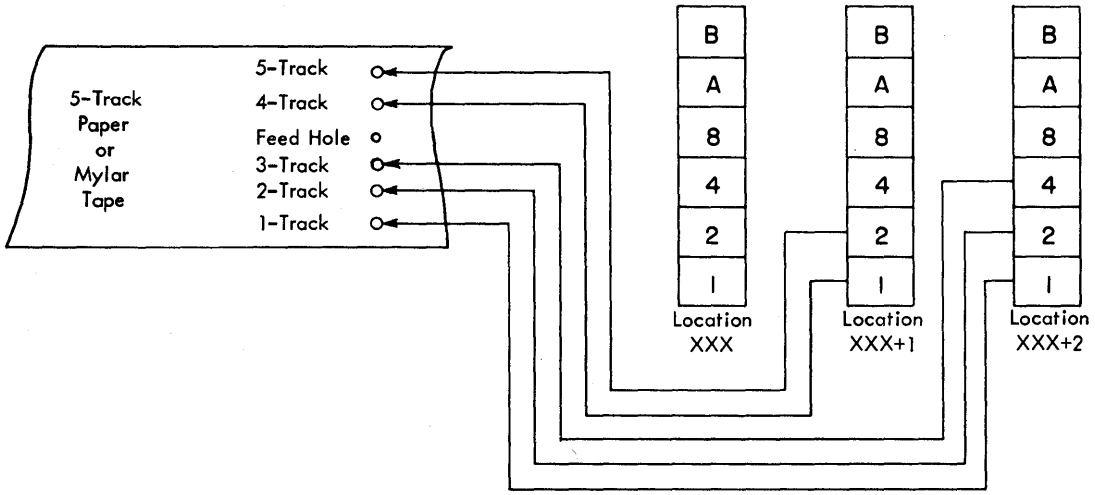


Figure E-42. Five-Track Bit Pattern to Tape Punching Translation

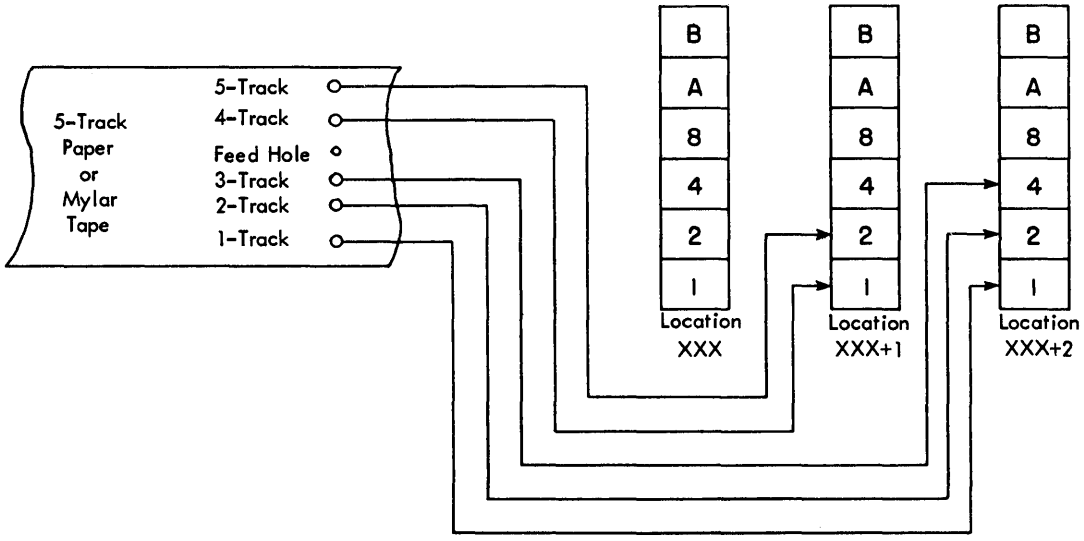


Figure E-43. Five-Track Tape Punching to Bit-Pattern Translation

IBM 1301 Disk Storage, Models 11, 12, 21, 22

The IBM 1301 Disk Storage, Models 11, 12, 21, 22 (Figure F-1), provides the 1440 system user with the advantages of large capacity random access storage. As many as five IBM 1301 modules can be attached to a 1440 system. Refer to *IBM 1301 Disk Storage*, Form A24-3157.

Disk-Control Field

A 10-digit disk-control field specifies the disk-storage area that is involved in the data transfer. This disk-control field is located in core storage, and begins at the core-storage address specified by the disk-storage instruction B-address. The data involved in the transfer follows the disk-control field (no data area is required for a seek-disk operation).

The various parts of the disk-control field are: alternate code, core sector address, and sector count (Figure F-2).

Alternate Code

If a lozenge (⌑) is used in this position, the core sector address specifies the disk drive that is to be selected.

A record mark (+), S, U, W, or Y character in the alternate-code position is used to select a drive other than the drive specified by the sector address. The +, S, U, W, and Y characters select the first, second, third, fourth, and fifth disk modules respectively.

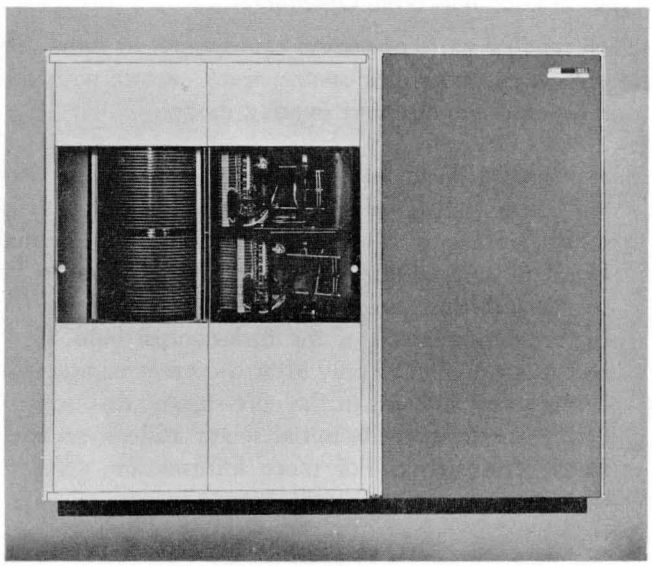


Figure F-1. IBM 1301 Disk Storage

Alternate Code	Core-Sector Address	Sector Count
x ⌑ or +,S,U,W,Y	xxxxxx 000000 - 999, 999	xxx 000 - 999

Figure F-2. Disk-Control Field

A word mark can be placed in the alternate-code position. The word mark does not affect the operation and is not lost. A 1-bit should never appear in the alternate-code position.

Core-Sector Address

The core-sector address contains the 6-digit address of the first sector to be operated upon. Before any disk operation is performed, an automatic comparison is made of the sector address in core storage with the disk-sector addresses on the specific track. If an equal comparison is made, the operation proceeds. If no equal comparison is made, the unequal-address compare indicator turns ON, and the disk operation is not performed. (When a multiple-sector operation is executed, only the address of the first-specified sector on each track involved in the operation is compared.)

When sector operations are performed, the core sector address is automatically increased by 1 immediately following the data transfer of each sector, except under these conditions:

1. Track operation being performed.
2. Sector-count field reaches the value of 000.
3. Wrong-length record.

When any of these conditions occurs, the core-sector address is not increased by 1.

Notes:

1. The six positions of the 6-digit core-sector address may contain any valid character that has a numeric-bit value of zero through nine.
2. Zone bits over the core-sector address positions are lost through the adder if any address modification takes place.
3. Word marks in the core-sector address positions do not affect the operation, but are lost during any operation performed in the load mode that involves address modification.

Sector Count

This field indicates the number of sectors to be operated upon during the disk operation. The sector-count field is not used during a seek operation, but the positions must be there because the disk-control field must be 10 positions long.

During the transfer of data to or from disk storage, the sector-count field is automatically decreased by 1 immediately following a successful address comparison, and before each additional sector is transferred. This operation results in the sector-count field reflecting the number of sectors transferred.

If a sector count of 000 is used when initiating a disk sector read or write operation, an error condition occurs. Before the first sector is transferred, a 1 is subtracted from the sector-count field. In this case, the result would be 999. Therefore, data would be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong-length record and any-disk condition indicators would be turned ON.

Notes:

1. Word marks cannot be placed over the sector-count field units position. Word marks in any other position do not affect the operation, but are lost during any operation performed in the load mode that affects sector-count modification.
2. Zone bits are always removed from all three positions of the sector-count field.

Basic Disk Operations

The four basic operations performed by the 1301 are seek, read, write, and write disk check.

Seek Operation

The seek operation is initiated by a seek-disk instruction, which directs the read/write heads to the proper cylinder on the disk drive. This instruction is followed by a read or write operation.

The data on the disk records is not acted on during this seek operation.

The seek operation positions the access arms over the specified cylinder. The B-address position of the instruction contains the core-storage address of the disk-control field and it is this field that specifies the proper cylinder plus other pertinent information.

Read Operation

The read operation is initiated by one of the three different types of read-disk instructions, and transfers data from disk storage to a specified area in core storage. (The three types of instructions are explained following the write-operation description.) The specified disk-storage area involved in the transfer is partially identified by the previous seek operation, and the rest of the area is fully identified before the data transfer takes place. The identification is accomplished by comparing the sector addresses on the disk with the sector address in core storage. The sector address in core storage is part of the disk-control field, and the B-address position of the read-disk instruction contains the core-storage address of the disk-control field. The data from the disk is placed in a core-storage area located immediately to the right of the disk-control field.

Write Operation

The write operation is initiated by one of the three different types of write-disk instructions, and transfers data from a specified core-storage area into disk storage. (The three types of instructions are explained following this operation description.) The specific disk-storage area involved in the transfer is partially identified by the previous seek operation, and the rest of the area is fully identified before the data transfer takes place. The identification is accomplished by comparing the sector addresses on the disk with the sector address in core storage, The sector address in core storage is part of the disk-control field, and the B-address portion of the write-disk instruction contains the core-storage address of the disk-control field. The data that is to be transfered to the disk is stored in a core-storage area located immediately to the right of the disk-control field.

Types of Read and Write Operations

Each read or write operation can operate in three different ways, or modes: sector, track sectors with addresses, and sector-count overlay modes.

Sector Mode. Read and write operations in the sector mode transfer data, but do not transfer disk-sector addresses. The sector mode is the normal mode of operation. The number of sectors to be handled during one operation is specified by the sector-count portion of the disk-control field. Each sector is transferred only after a correct comparison of the sector address in the core-storage disk-control field is made with the initial sector address on each track of the disk. For more information, refer to the specific instruction.

Track-Sectors with Addresses Mode. This mode of operation transfers both the data and the disk-sector

addresses to and from the disk, one complete track at a time. The mode of operation makes it possible to change the previously recorded sector addresses. The operation requires that the sector-address portion of the disk-control field contain the address of one of the sectors within the specified track, and the sector-count portion of the disk-control field must contain 020 (20 sectors will be transferred). The transfer can occur only after a correct comparison of the sector address in the core storage disk-control field with a sector address on the specified track. For more information, refer to the specific instruction.

Sector-Count Overlay Mode. This mode of operation allows a portion of the data record itself to specify the number of sectors to be involved in the data transfer. The disk-sector addresses are not part of the transfer. This mode of operation permits better disk-storage utilization for sequential applications involving variable-size records. For more information, refer to the specific instruction.

Reading and Writing with Word Marks Mode. Word-marks can be transferred with the data during all reading and writing operations by an L Op code instead of an M Op code. When word marks are written on the disk, the data is written in an 8-bit BCD coding.

Write Disk Check

The write-disk-check operation causes the data in the specified disk area to be compared against the comparable data in the specified core-storage area. When the disk data does not compare, bit-by-bit and character-by-character, with the core-storage data, a disk-error indicator is set ON. This operation takes the form of a write-disk-check instruction, which normally must follow each write operation. The write-disk check operation compares the data written in disk storage with the original source data in core storage.

IBM 1301 Instruction Format and Instructions

Instructions applying to the 1301 cannot be successfully chained.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
xx	M/L	%Fn	BBB	R/W

Op Code

This is always a single character that defines the basic operation to be performed. Either the M or L operation code can be used with IBM 1301 instructions.

When the M Op code is used, characters are written or read in 7-bit mode (CBA 8421). The sector character capacity in the 7-bit mode is 100 characters. The L Op code causes characters to be read or written in 8-bit mode (CBA 8421M). The 8-bit mode provides for a possible word mark with the character being written on, or read from, the disk record. The sector character capacity in the 8-bit mode is 90 characters.

A-Address

%Fn signals that the disk unit is to be selected; n represents the digit used to perform various operations.

n-Position	*Operation*
0	Seek a disk record.
1	Sector—Reading or writing characters from the number of sectors specified by the sector-count field is stopped when a group mark with a word mark, or the end-of-sector, is sensed. If a group mark with a word mark is sensed before the reading of the sector(s) is completed, reading stops and the wrong-length record and any-disk condition indicators turn ON. If the group mark with a word mark is sensed before the writing of a record on a disk is completed and it is before the end of a record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk condition and wrong-length-record indicators are turned. on.
6	Disk Track-Sector with Addresses—Allows the reading or writing of a full track (20 sectors) including sector addresses.
3	Write Disk Check—Data written on a disk in a preceding write operation is read from the disk and compared, character-by-character, with the data in core storage. A write-disk-check instruction must be given following a write operation, unless an error occurred *during* the write operation. A write-disk-check operation can be executed after a read operation if a check on the information read is desired. The operation is performed exactly the same as a write-disk-check operation following a write operation.
5	Sector-Count Overlay—Allows for records of a variable number of sectors (more than one) to be read or written with a single instruction. The number of sectors to be read/written is controlled by the multiple sector-count field. This control field is in the first three data positions of the first sector of the disk record. This technique permits better disk storage utilization for sequential applications involving variable-size records. The record itself specifies the number of sectors involved.

B-Address

The B-address specifies the high-order postion in core storage of the 10-digit disk-control field. The disk-control field is followed by the area of core storage that is to have data read into or out of by a group mark with a word mark.

d-Character

The d-character is used to specify the operation to be performed. The d-character R specifies a read operation; the d-character W specifies a write operation.

Seek Operation

Seek Disk

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
SD	M	%F0	BBB	R
LU	L	%F0	BBB	R

Note: Because word marks have no meaning for the seek-disk operation, no load-mode Autocoder mnemonic is provided. However, the "general" mnemonic (LU) can be used if an L op code is desired.

Function. The A-address specifies that a seek operation is to be performed by the access assembly. The B-address specifies the high-order position in core storage of the disk-control field. Only the alternate-code position and the six-position core-sector address are used during a seek-disk operation, but the disk-control field must be 10 positions long.

The selected access assembly moves from the old setting directly to the new setting. The functions, associated with the direct-seek special feature are standard in the 1301.

Word Marks. Word marks are not affected.

Timing. T = .1665 ms + access time.*
*180 ms is maximum access time for a seek.

160 ms is average access time for a seek.

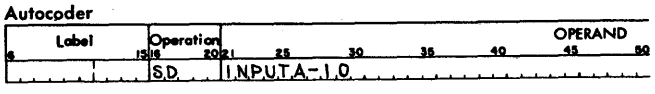
0 ms if access mechanism is at track (seek-disk instruction not given).

Note: If the access mechanism is already at the disk track that is to be used, a seek-disk instruction need not be given.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B+6	B+7

Example. Seek record 015734 with the access assembly. Storage locations 0590-0599 (labeled INPUTA) contains ⌖ or ⌑ 015734001 (Figure F-3).



Assembled Instruction: M %F0 590 R

Figure F-3. Seek Disk

Sector Operations

If only the data portion of a disk record is to be affected, the operation is classified as a *sector operation* (addresses are not affected). Disk records can be read, written, or scanned during *sector operation*. The term *sector operation* does not mean that a disk record is confined to a 100-character sector. The data needed for a record can be written in as many sectors as needed.

Read Disk Sector(s)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RD	M	%F1	BBB	R

Function. This instruction causes data to be read from disk storage into core storage. The digit 1 in the A-address (%F1) specifies that a sector operation is to be performed. The number of sectors to be read is specified by the sector-count field. The reading of the disk is stopped by a group mark with a word mark in core storage and by the end of the sector.

Reading begins at the address contained in the core-sector address field and continues for the number of sectors specified by the sector-count field.

The core-sector address field is *increased* by one for each sector read, and the sector-count field is *reduced* by one as a sector is read.

When the sector-count field reaches 000, an end-of-operation is indicated to the system. An error condition results from any disk-sector read or write operation that begins the operation with a sector count of 000. Before the first sector is transferred, a one (1) is subtracted from the sector-count field, resulting in a sector count of 999. Data would then be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong-length record and any-disk condition indicators are turned on.

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage reserved for the data read from the disk.

The R in the d-character position signifies a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the disk record. If a group

mark with a word mark is detected before reading of the record is completed, the wrong-length-record indicator turns on and reading stops. The position of the group mark with a word mark can be determined by using the formula:

$GMWM = B + N_S(L_S) + 10$

B = Address of high-order position of disk address in core storage.

N_S = Number of sectors read.

L_S = Number of characters per sector.

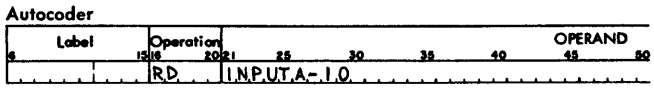
Timing. T = .0999 ms + 1.7N_S + disk rotation.*
*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Note: Before reading starts, an automatic comparison is made of the core-sector address with the sector address on the disk. This check is made for the first sector on each track involved in the operation. If they are not the same, the unequal-address compare indicator turns on, and the data on the disk cannot be read into storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + N_SL_S (no overlay)

Example. Read one sector from disk storage into core storage beginning at location 0600 (labeled INPUTA.) In Figure F-4, the disk-control field is in the ten positions preceding the label (0590-0599).



Assembled Instruction: M %F1 590 R

Figure F-4. Read Disk Sector

Read Disk Sector(s) with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDW	L	%F1	BBB	R

Function. This is similar to the read-disk-sector instruction except that (1) word marks in the record area of core storage are removed, and (2) word marks from the disk record are written in core storage. The length of the sector read from disk storage into core storage is 90 positions.

Word Marks. A group mark with a word mark in core storage terminates the read operation. If the group mark with a word mark is not in the position to the right of the last character read from the disk into core storage, the wrong-length-record and any-disk condition indicators turn ON.

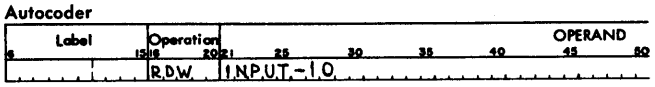
Timing. T = .0999 ms + 1.7N_S + disk rotation.*
*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Note: If a disk is read in a mode different from the one in which it was written (M or L operation code), a parity error occurs. The disk-error indicator turns ON.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + N_SL_S (no overlay)

Example. Read a record, with its associated word marks, from disk storage into the area labeled INPUT (first position of data is at 0600). The disk-control field is located in the ten positions preceding the label (0590-0599); Figure F-5.



Assembled Instruction: L %F1 590 R

Figure F-5. Read Disk Sector with Word Marks

Read Disk with Sector-Count Overlay

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDCO	M	%F5	BBB	R
RDCOW (with word marks)	L	%F5	BBB	R

Function. This operation is similar to the read-disk-sector(s) instruction except that the number of sectors to be read is controlled by the first three positions in the first record read. The digit 5 in the A-address specifies that an overlay operation is to be performed.

As the first sector is read from disk storage, the first three digits of the record being read are placed in the sector-count field of the disk-control field in core storage. Therefore, if a variable number of sec-

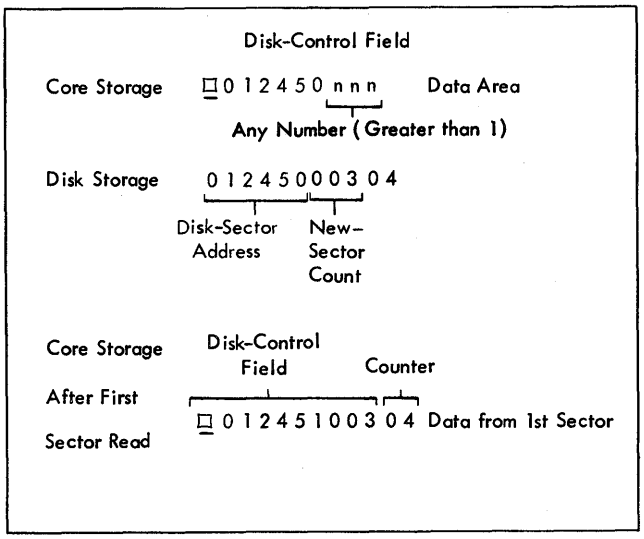


Figure F-6. Read Disk – Sector-Count Overlay Operation

tors is to be read from disk storage, the sector-count field must contain a value greater than 001 to cause the first sector to be read. The first three positions of the first sector read contain the number of additional sectors to be read. Figure F-6 illustrates the operation of an overlay instruction, which causes four sectors of data to be read from disk storage into core storage.

The operation proceeds as a normal read operation with appropriate changes to the core-sector address and sector-count fields.

Word Marks. If the exact number of positions of data to be read from disk storage is not known when this operation is initiated, place the group mark with a word mark (signalling the end-of-operation) one position to the right of the last possible character to be read using this instruction. If the maximum number of records is not read, the read-into-storage stops because the end-of-sector is reached and the sector-count field is all zeros before the group mark with a word mark is sensed. The wrong-length-record indicator also turns on. The programmer can check core storage in this case to see if the correct number of sectors has been read.

This can be accomplished by setting up a counter in the fourth and, if necessary, fifth position of the first sector of the record. This counter, when the read operation is completed, is located in the first and/or second position of the data record in core storage. These positions can be used to check the number of sectors in the record. These counter positions should equal the number of sectors read. Therefore, data reading should have stopped at $B + 6 + N_S L_S$. If it did not, then an error did occur and appropriate action should be taken. If a correct read has occurred, the error indication can be disgarded.

Timing. $T = .0999 \text{ ms} + 1.7N_S + \text{disk rotation.}^*$

*35 ms is maximum time for disk rotation.

18.4 ms is average time for disk rotation.

1.7 ms is minimum time for disk rotation.

Notes:

1. Before reading starts, an automatic comparison is made of the sector address in core storage with the sector address on the disk. This check is made for the first sector on each track involved in the operation. If the addresses are not the same, the unequal-address compare indicator is turned on, and the data on the disk cannot be read into storage.

2. Special consideration must be given to single-sector records read in the sector-count overlay mode. When the read operation begins, the first three data characters of the record overlay the sector-count portion of the disk-control field. When a single-sector operation is specified, 000 is read in and overlaid in the sector-count positions. The system-detection circuits only detect a zero sector count when it is produced by automatically decreasing the sector-count field, however. After reading the single-sector record, the read operation does not end. The sector address is increased by one, and the sector-count field is decreased by one, resulting in a sector count of 999. Because the sector-count field does not contain all zeros at the end of the operation, the wrong-length-record and any-disk indicators are turned ON.

   When an initial sector count of 003 is used and the first three digits of the first sector read are 000 (the three digits to be overlaid), the following occurs:

   1. The operation does *not* stop because the sector count has not been decremented to 000.
   2. The sector address has been incremented and the second sector is read.
   3. The special-add operation (used to keep account of the sector count) decrements the sector count (000) to 999.

   Because the last step (item 3) does not produce a carry to increment the sector address, an unequal-address compare occurs on the attempt to read the third sector. The unequal-address compare does not occur when the initial sector count is 001, but the read continues until a group mark with a word mark is sensed in core storage.

   Single-sector and *multiple*-sector read operations cannot be interspersed (using the M/L %F5 BBB R instruction) without prior knowledge of exactly *when* each read occurs.

   If a disk drive includes single-sector records, a special routine must be included to verify the validity of the record read. Before execution of a read operation, a special character that would never be found in the last position of a record can be moved to the 100th position of the core-storage input area. The wrong-length-record routine can then check to see whether the counter in the first position of the record contains a one (1). If it does, the routine looks to see that the special character has been overlaid in core storage. If it has been, the record was read in its entirety.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 3 + $N_S L_S$

Example. Read into core storage a variable number of sectors that contain the data for a record beginning at location 0900 (labeled INPUTB). In Figure F-7, the disk-control field address is located in the ten positions preceding the label (0890-0899).

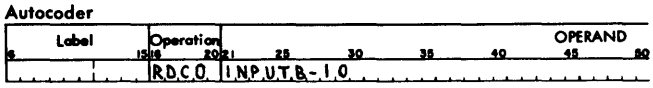


Figure F-7. Read Disk with Sector-Count Overlay

Write Disk Sector(s)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WD	M	%F1	BBB	W

Function. This instruction causes record data in core storage to be written on a disk record. The digit 1 in the A-address (%F1) specifies that a sector operation is to be performed. The number of sectors to be written is specified by the sector-count field. The writing of the disk record is stopped by a group mark with a word mark in core storage and by the end-of-sector.

Writing begins at the address contained in the core-sector address field and continues for the number of sectors specified by the sector-count field.

The core-sector address field is *increased* by one for every sector written. The sector-count field is *reduced* by one as a sector is written.

When the sector-count field reaches 000, an end-of-operation is indicated to the system. An error condition results from any disk sector read or write operation that begins the operation with a sector count of 000. Before the first sector is transferred, a one (1) is subtracted from the sector-count field, resulting in a sector count of 999. Data would then be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong-length-record and any-disk-condition indicators are turned ON.

The B-address specifies the high-order position in core storage of the disk-control field, and is followed by the data to be written on the disk.

The W in the d-character position signifies a write operation.

Word Marks. A group mark with a word mark must be *one* position to the right of the last character of the record in core storage. The writing of data stops when the end-of-record is reached on the disk and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end of a record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk-condition and the wrong-length-record indicators are turned ON. The position of the group mark with a word mark can be determined by using the formula:

$$GMWM = B + N_S(L_S) + 10$$

Timing. T = .0999 ms + 1.7N_S + disk rotation.*
*35 ms is maximum time for disk rotation.

18.4 ms is average time for disk rotation.

1.7 ms is minimum time for disk rotation.

Notes: Before writing starts, an automatic comparison is made of the core-sector address with the record address on the disk, This check is made for the first sector on each track involved in the operation. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data in storage cannot be written on the disk.

If the data in core storage contains characters with word marks only, the CBA8421 portion of the character is written on the disk (the word mark is ignored).

A write-disk-check instruction must be performed following a write-disk operation unless an error occurred during the write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + $N_S L_S$ (no overlay)

Example. Write a disk record (one sector) from the data in the area labeled INPUTA (first position if data is at 0600). In Figure F-8, the disk-control field is located in the ten positions preceding the label (0590-0599).

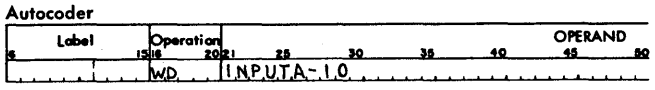


Figure F-8. Write Disk Sector

Write Disk Sector(s) with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDW	L	%F1	BBB	W

Function. This instruction is similar to the write-disk-sector instruction, except that word marks set with the data in core storage are recorded on the disk record. This mode of operation permits writing programs on disk records for system use. Ninety positions of data with word marks are recorded on each sector during the write operation.

Word Marks. A group mark with a word mark one position to the right of the last character of the record in core storage terminates the write operation. If the group mark with a word mark is sensed before the end of a record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk-condition and the wrong-length-record indicators are turned ON.

Timing. T = .0999 ms + 1.7N_S + disk rotation.*

*35 ms is maximum time for disk rotation.

18.4 ms is average time for disk rotation.

1.7 ms is minimum time for disk rotation.

Notes: The programmer should be certain that all records on a specific track are written in the same mode (M or L operation code). Otherwise, track operations are not possible.

Before writing starts, an automatic comparison is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data in storage cannot be written on the disk. A write-disk-check operation must be performed following this instruction.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + N_SL_S (no overlay)

Example. Write a disk record, with word marks, from the data in the area labeled OUTPUT (first position of data is 0600). In Figure F-9, the disk-control field is located in the ten positions preceding the label (0590-0599).

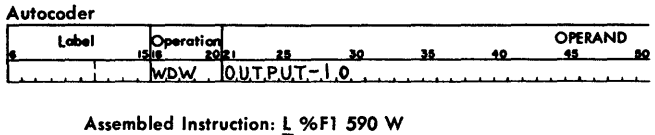


Figure F-9. Write Disk Sector with Word Marks

Write Disk with Sector-Count Overlay

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDCO	M	%F5	BBB	W
WDCOW (with word marks)	L	%F5	BBB	W

Function. This operation is similar to the write-disk sector instruction except that the sector-count field of the disk-control field is automatically decreased by one and then written in the first three data positions of the first sector written. The digit 5 in the A-address specifies that an overlay operation is to be performed.

Therefore, the sector-count field in core storage should contain the number of sectors to be written. The first three data positions of the first sector written contain the number of additional sectors that were written. Figure F-10 illustrates the operation of an overlay instruction, which causes four sectors of data to be written from core storage onto disk storage.

The operation proceeds as a normal write operation with appropriate changes to the core-sector address and sector-count fields.

Word Marks. A group mark with a word mark should be placed one position to the right of the last sector to be written. The group mark with a word mark must be placed at B + 7 + N_SL_S to avoid a false wrong-length-record indication.

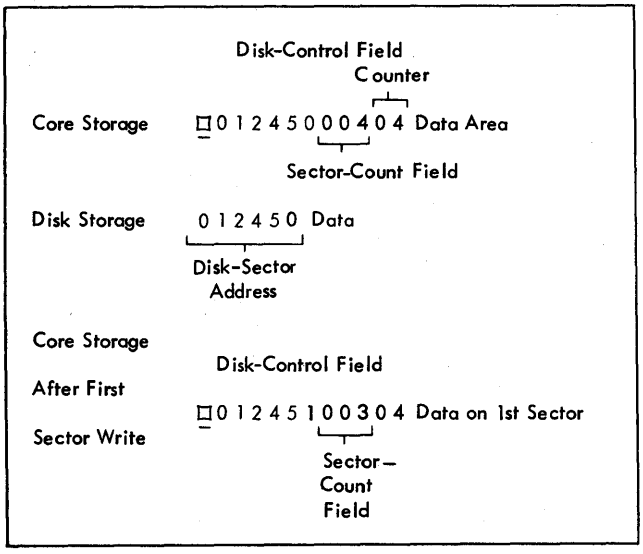


Figure F-10. Write Disk – Sector-Count Overlay Operation

Timing. T = .0999 ms + 1.7N_S + disk rotation.*

*35 ms is maximum time for disk rotation.

18.4 ms is average time for disk rotation.

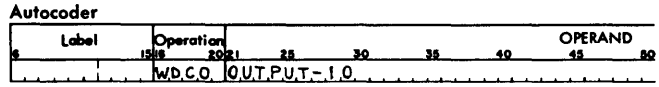
1.7 ms is minimum time for disk rotation.

Note: Before writing starts, an automatic comparison is made of the record address in core storage with the record address on the disk. This check is made before the first sector on each track involved in the operation. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data in core storage cannot be written in disk storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 8 + N_SL_S

Example. Write a number of sectors for a record on disk storage that contains data beginning at location 0900 (labeled OUTPUT). In Figure F-11, the disk-control field is located in the ten positions preceding the label (0890-0899).



Assembled Instruction: M %F5 890 W

Figure F-11. Write Disk with Sector-Count Overlay

Write Disk Check

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDC	M	%F3	BBB	W
WDCW (wih word marks)	L	%F3	BBB	W

Function. This instruction causes a comparison, character-by-character, of the data in core storage with the data just written on the disk. The instruction must be executed after a write operation and before any other disk-storage operation is initiated.

The digit 3 in the A-address specifies that a WRITE DISK CHECK is to be performed. Either an L or M operation code is used, depending on how the data was recorded in disk storage.

The B-address specifies the area in core storage that contains the disk-control field and the data recorded on the disk.

The sector-address and sector-count fields of the sector-control word must be restored to the values present at the beginning of the write operation.

Word Marks. A group mark with a word mark must appear one position to the right of the disk data in core storage.

Timing. T = .0999 ms + 1.7N_S + disk rotation time.*

*35 ms is maximum time for disk rotation.

18.4 ms is average time for disk rotation.

1.7 ms is minimum time for disk rotation.

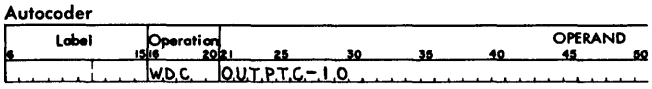
Notes: If the disk address in core storage is not the same as the address on the record, the unequal-address compare indicator turns on. If any of the characters on the disk record do not agree with the characters in core storage, the disk-error indicator turns on.

A write-disk-check operation can be executed after a read operation if a check on the information read is desired. The operation is performed exactly the same as a write-disk-check-operation following a write operation.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Depends on previous operation	

Example. Compare the disk record with a record in core-storage area labeled OUTPTC (beginning at 0700). In Figure F-12, the disk-control field is located in the ten positions preceding the label (0690-0699).



Assembled Instruction: M %F3 690 W

Figure F-12. Write Disk Check

Address Operations

The ability to read and/or alter disk addresses is conditioned by certain IBM 1440-1301 instructions. These instructions contain the term *Address* in their description.

If the proper instruction is not used when trying to perform an address operation, the system stops and the RAMAC light on the system console turns on.

Read Disk Track Sector with Addresses

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDT	M	%F6	BBB	R
RDTW (with word marks)	L	%F6	BBB	R

Function. This instruction causes the contents of an entire disk track (addresses and data) to be read in the mode specified by the operation code M or L. If the L operation code is used, the track is read into storage with its associated word marks.

The core-sector address must correspond to any one of the sector addresses on the track. The disk track is scanned for an address equal to the sector address in core storage. The disk-track reading begins when the first track-index pulse following a successful address-compare operation is sensed. If the result of the address-compare operation is unequal, the unequal-address-compare indicator turns ON. All twenty sectors on the track, including the disk-sector addresses, are read into core storage. The sector-count field of the disk address must be set at 020 before the operation begins.

The core-sector address field is not modified by plus-one during this operation. To keep track of the number of sectors read, however, the sector-count field is modified by minus-one for each sector read.

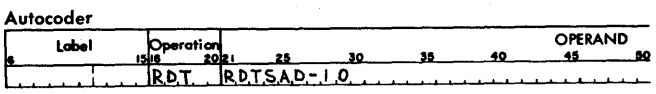
Word Marks. A group mark with a word mark must be placed one position to the right of the last character read into core storage. This position can be found by adding 2130 to the B-address for operations performed with the M operation code, and 1930 for operations performed with the L operation code.

Timing. T = 33.3999 ms + disk rotation.*
*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg*
NSI	B + 9	B + 11 + 2120 (M Op code) or B + 11 + 1920 (L Op code)

Example. Read disk track 17550, with its associated word marks, into the core-storage area labeled RDTSAD (first position of data is at 0800). In Figure F-13, the disk-control field is located in the ten positions preceding the label (0790-0799).



Assembled Instruction: M %F6 790 R

Figure F-13. Read Disk Track Sectors with Addresses

Write Disk Track Sectors with Addresses

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDT	M	%F6	BBB	W
WDTW (with word marks)	L	%F6	BBB	W

Function. This instruction causes the record data and addresses in core storage to be written on a disk track in the mode specified by the operation code (M or L). If the L operation code is used to write the track, word marks in the record area of core storage are written on the track.

The core-sector address must correspond to any one of the sector addresses on the track. The disk track is scanned for an address equal to the sector address in core storage. Writing the disk track begins when the track-index pulse is sensed (signaling first sector on track), if the address-compare operation results in an equal condition. If the result of the address-compare operation is unequal, the unequal-address-compare indicator turns on. All 20 sectors on the track, including the disk sector addresses, are written on disk storage. The sector-count field of the disk-control field must be set at 020 before the operation begins.

The core-sector address field is not modified by plus-one during this operation. To keep track of the number of sectors written, however, the sector-count field is modified by a minus for each sector read.

Word Marks. A group mark with a word mark must be one position to the right of the last character written on disk storage. This position can be found by adding 2130 to the B-address for operations performed with the M operation code, and 1930 for operations performed with the L operation code.

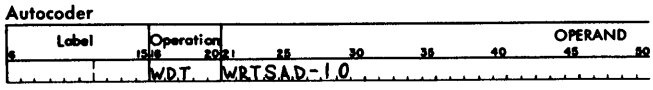
Timing. T = 33.3999 ms + disk rotation.*
*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2120 (M Op code) or B + 11 + 1920 (L Op code)

Example. Write a disk track (with address) from the data in the area labeled WRTSAD (first position of

data is at 1000). In Figure F-14, the disk-control field is located in the ten positions preceding the label (0990-0999).



Assembled Instruction: M %F6 990 W

Figure F-14. Write Disk Track Sectors with Addresses

Branch if Indicator On

The BRANCH IF INDICATOR ON instruction tests the indicators that might be set on during a disk-storage operation. When a disk-storage instruction occurs in the program, it turns off all disk-storage indicators that were turned on by a previous disk-storage operation. The execution of a disk-storage instruction can result in a disk-storage indicator being turned on.

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. The d-character specifies the indicator tested. If the indicator is on, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken. Figure F-15 shows symbols that are valid d-characters, and the indicators they test. More than one indicator can be turned on as the result of a disk-storage operation.

Because the program continues in sequence, a BRANCH IF INDICATOR ON instruction should immediately follow any disk instruction.

Indicators

Access Inoperable. This indicator turns on if either an invalid (not installed) arm or disk-storage unit is addressed.

d - Character	Indicator
N	Access Inoperable
V	Validity Error
W	Wrong - Length Record
X	Unequal - Address Compare
Y	Any Disk Condition
\	Access Busy

Figure F-15. d-Characters for BRANCH IF INDICATOR ON Instruction

An access arm becomes inoperable if the logic safety circuit detects improper operation. A customer engineer can also render an arm inoperable. In either instance, this indicator turns on, at which time the operation is terminated and the next sequential instruction is started. At the same time, the RAMAC light turns on.

This indicator also turns on if power is not supplied to the disk-storage unit addressed, or if certain data-safety conditions occur.

The indicator is turned off during the I-phase portion of the next disk-storage operation.

Disk Error. This indicator turns on if even-bit parity occurs during reading or writing on a disk. Another condition that turns the indicator on is a data unequal-compare condition during a write-check operation. In this case the operation is completed.

The indicator is turned off during the I-phase portion of the next disk-storage operation.

Wrong-Length Record. This indicator turns on if the following conditions are not satisfied: a group mark with a word mark in core storage is sensed at the same time as an end of sector and an all-zero condition in the sector-control field occurs. It also turns on during a scan operation if the search argument is longer than, or equal to, a sector length. Detection of a wrong-length record terminates the operation and starts the next sequential instruction.

The indicator is turned off during the I-phase portion of the next disk-storage operation.

Unequal-Address Compare. An unequal-address-compare condition occurs during the automatic comparison of the sector address in storage with the sector address on the disk. This unequal condition turns the unequal-address-compare indicator on after the disk track is searched and the track-index pulse is sensed twice. This is an automatic check and does not have to be programmed. During multiple-sector operations, the indicator also turns on after the data transfer begins when the next sector address to be compared does not compare.

The internal circuitry is the same as that used by the COMPARE instruction. In programming, be careful that a normal-compare operation and the address-compare operation do not interfere with the settings of the equal-, low-, and high-compare indicators set by a previous instruction. Detection of an unequal-address-compare terminates the operation and starts the next sequential instruction.

The indicator is turned OFF during the I-phase portion of the next disk-storage operation.

Access Busy. This indicator is turned off by every disk storage command. If the access assembly is in motion, the indicator is turned on and the instruction is not executed.

A branch-access-busy command tests the indicator. If on, this indicates that the previous disk-storage instruction was not executed and should be repeated.

An indicator is provided for each drive attached to the system.

The following examples illustrate the access-busy operation.

Operation	*Remarks*
Start	Begins program execution.
Seek Access 0	Turns indicator (0) off Test for access motion – no motion Indicator (0) remains off Instruction executed.
Seek Access 2	Turns indicator (2) off Test for access motion – no motion Indicator (2) remains off Instruction executed.
Branch Access Busy	No branch

Any-Disk Condition. This indicator turns on if any of the other disk-storage indicators are on. It can be tested by the program, and if it is off, the program can proceed. If this indicator is on, check the other indicators to determine where corrective measures should be taken.

This indicator is turned off during the I-phase portion of the next disk-storage operation.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Note: After each disk-unit read or write operation, the program must test for error indications to prevent processing of unusable data.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. At the completion of a disk-read operation, test the any-disk-unit error-condition indicator. If it is OFF, continue in the main program. If it is ON, branch to the routine labeled DISKER (0690) to determine the type of error condition. This tests all disk-unit indicators and branches to the error routine of the respective indicator that is ON. In Figure F-16, the routines are labeled: ACINOP (0690), UNADCL (0695) WRLENR (0700), RWPARC (0705).

Autocoder

```
Label      Operation      OPERAND
6      15  16    20 21  25    30    35    40    45    50
           B.I.N.        D.I.S.K.E.R.,Y.
D.I.S.K.E.R  B.I.N.      A.C.I.N.O.P.,N.
           B.I.N.        U.N.A.D.C.L.,X.
           B.I.N.        W.R.L.E.N.R.,W.
           B.I.N.        R.W.P.A.R.C.,V.
```

Assembled Instruction:

```
480 B 690 Y
690 B 740 N
695 B 790 X
700 B 890 W
705 B 990 V
```

Figure F-16. Branch If Indicator On Testing Routine

IBM 1301 Disk-Storage Timing

The organization of data in disk storage and the method of processing data affect the seek time for a given operation and also affect the total system throughput.

Access Motion Time

The access mechanism requires time to move from one cylinder to another. The time required is related to how far the mechanism moves within certain machine-defined limits. To calculate how much time is required, consider the 250 cylinders of a module as being organized into five areas of 50 cylinders per area (Figure F-17). Also consider each area of cylinders further divided into six sections (Figure F-17). Access motion time for any one access can be determined by one of the following statements:

1. To move the access mechanism within a section of any one area requires 50 milliseconds.
2. To move the access mechanism from one section to another section of an area requires 120 milliseconds.
3. To move the access mechanism from one area to another area (crossing an area boundary) requires 180 milliseconds.

For example, to move the access mechanism from track 000000 to track 039999 requires 120 milliseconds of access motion time. To move the access mechanism from track 039999 to track 040000 requires 180 milliseconds of time.

Rotational-Delay Time

A disk-storage read or write operation includes a timing factor called *rotational-delay time*. An index point for each circular disk track denotes the beginning and end of a track. After a cylinder of tracks has been accessed and the proper read/write head for a specific track of the cylinder is conditioned, actual reading or writing must wait until the specific data or data area of the track is located. *Rotational-delay time* is the time required for the disk to position the desired record at the selected read/write head after an instruction has been initiated.

Maximum *machine* rotational-delay time is 33.3 milliseconds; average rotational-delay time is 16.7 milliseconds. Data-access time includes the combination of access motion time and rotational-delay time. Figure F-18 is a complete chart of access motion time.

Access time from one sector address to another can be determined from Figure F-19. The point of intersection of two lines on a coded area of the figure, one drawn horizontally from a FROM sector address and one drawn vertically from a TO sector address, indicates access time in milliseconds.

Sector Processing Time

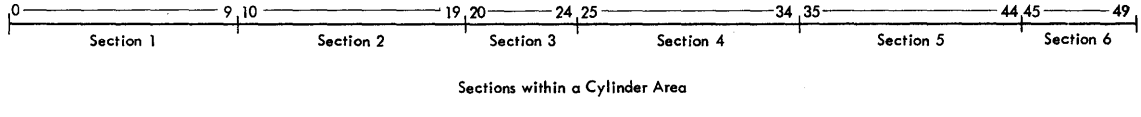
The times required to execute a 1-sector and a 3-sector operation are:

	1-Sector Operation	3-Sector Operation
Seek	160.0 ms	160.0 ms
Average Rotational Delay	16.7 ms	16.7 ms
Read (Includes Module Select Time—1.66 ms)	3.4 ms	6.7 ms
Rotational Delay	30.0 ms	26.7 ms
Write (Includes Module Select Time—1.66 ms)	3.4 ms	6.7 ms
Rotational Delay	30.0 ms	26.7 ms
Write Disk Check (Includes Module Select Time—1.66 ms)	3.4 ms	6.7 ms
	246.9 ms	250.2 ms

If possible, keep processing within the available rotational time. If not, the cycle is increased by one 33.3 ms revolution for each extension of available processing time.

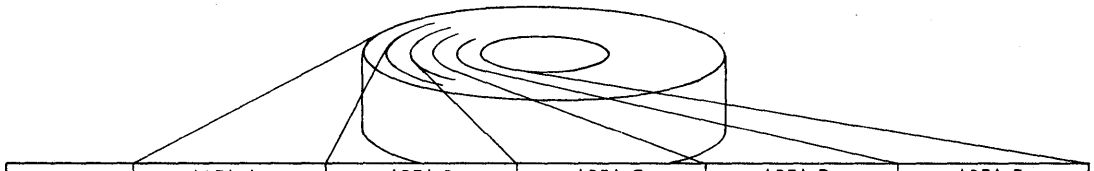
Processing time between a write operation and a write-disk-check operation can be used for such processing as updating control totals and/or arranging fields of printing.

AREA	CYLINDER
A	0-49
B	50-99
C	100-149
D	150-199
E	200-249



Sections within a Cylinder Area

Figure F-17. Access Motion Areas and Sections



SECTIONS	AREA A Cylinders 0 - 49 Sector Addresses 000000 - 039999	AREA B Cylinders 50 - 99 Sector Addresses 040000 - 079999	AREA C Cylinders 100 - 149 Sector Addresses 080000 - 119999	AREA D Cylinders 150 - 199 Sector Addresses 120000 - 159999	AREA E Cylinders 200 - 249 Sector Addresses 160000 - 199999
1	000000 - 007999	040000 - 047999	080000 - 087999	120000 - 127999	160000 - 167999
2	008000 - 015999	048000 - 055999	088000 - 095999	128000 - 135999	168000 - 175999
3	016000 - 019999	056000 - 059999	096000 - 099999	136000 - 139999	176000 - 179999
4	020000 - 027999	060000 - 067999	100000 - 107999	140000 - 147999	180000 - 187999
5	028000 - 035999	068000 - 075999	108000 - 115999	148000 - 155999	188000 - 195999
6	036000 - 039999	076000 - 079999	116000 - 119999	156000 - 159999	196000 - 199999

NOTE: Numbers shown above are the sector addresses of the lowest - and highest - numbered sector addresses in each section of each area.

Examples: Area A, section 1 contains 8000 sector addresses (000000 - 007999)
Area A, section 3 contains 4000 sector addresses (016000 - 019999)

Access Motion Time is calculated as follows:

1. Movement between sector addresses in the same section of an area is 50 milliseconds.
2. Movement between sector addresses not in the same section of an area requires:
   120 milliseconds when movement is within the same area
   (Between 000000 and 015000; 020000 and 036000)
   180 milliseconds when movement is between two of the five areas
   (Between 015000 and 055000; 108000 and 168000)

Figure F-18. Access Motion Time

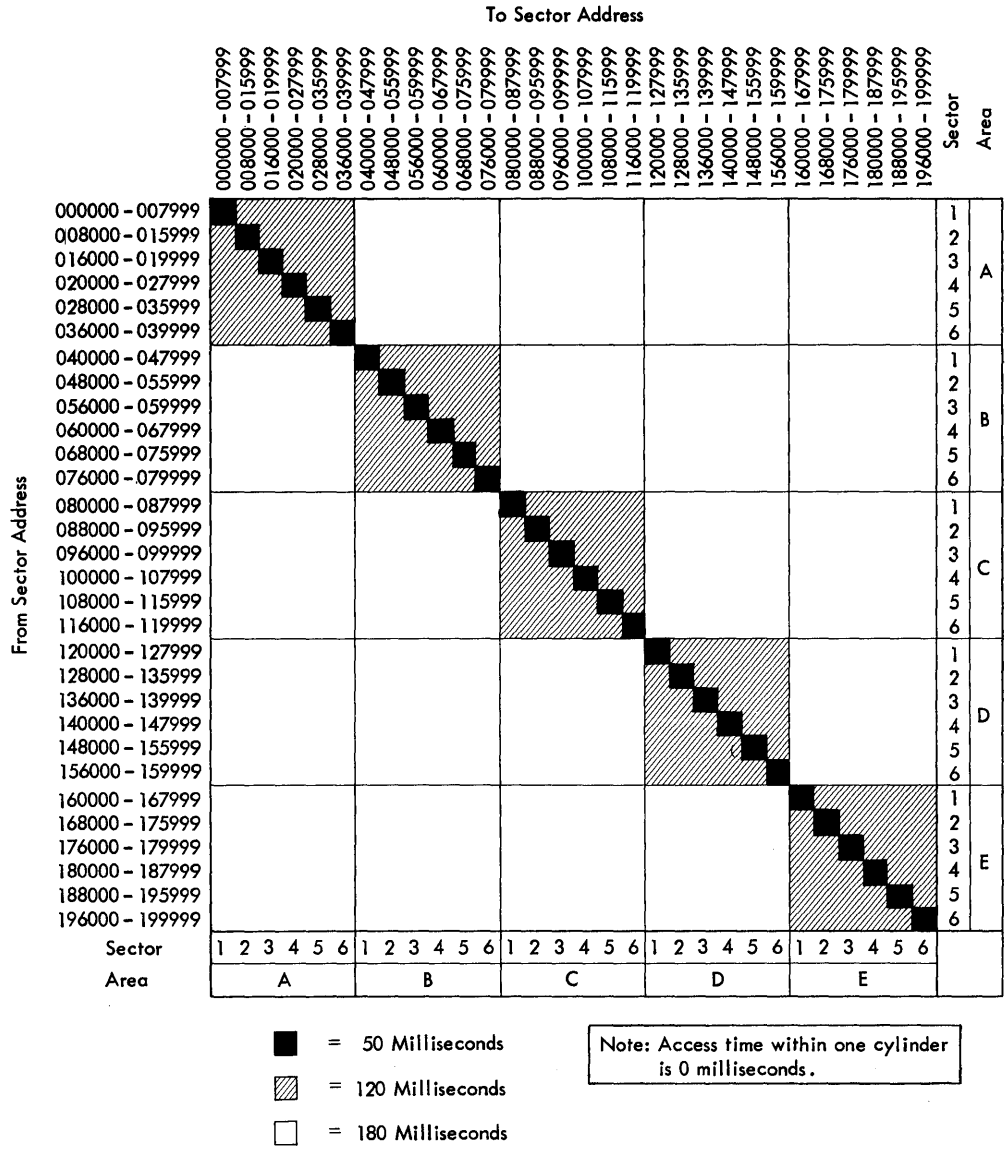


Figure F-19. Sector-Address-to-Sector-Address Access Time

IBM 1311 Disk Storage Drive

The IBM 1311 Disk Storage Drive (Figure F-20) provides the 1440 user with fast, efficient disk storage. As many as five IBM 1311 drives can be attached to a 1440 system, and each drive is equipped with an interchangeable disk pack capable of storing from 2 to 2.9 million alphameric characters. Refer to *IBM 1311 Disk Storage Drive*, Form A24-3086.

The first disk-storage drive attached to the 1440 system must be a 1311 Model 1; additional drives are 1311 Model 2.

Disk-Control Field

A 10-digit disk-control field specifies the disk-storage area that is involved in the data transfer. This disk-control field is located in core storage, and begins at the core-storage address specified by the disk-storage instruction B-address. The data involved in the transfer follows the disk-control field (no data area is required for a seek-disk operation).

Figure F-20. IBM 1311 Disk Storage Drive, Model 1

Alternate Code	Core-Sector Address	Sector Count
x * or 0 - 8 (even)	xxxxxx 000000 - 099, 999	xxx 000 - 999

Figure F-21. Disk-Control Field

The various parts of the disk-control field are: alternate code, disk-sector address, and sector count (Figure F-21).

Alternate Code

If an asterisk (*) is used in this position, the disk-sector addresses of the disk pack correspond to the address range for the disk drive on which the disk pack is placed.

A digit in the alternate-code position can be used to select the disk drive by the instruction. It allows drives with the same range of sector addresses to be used by the program during the same run.

When all disk drives have different sector addresses, an asterisk (*) instead of a numeric code can be placed in the alternate-code position if the address range of the disk packs and disk drive are the same.

Both word marks and zone bits can be placed in the alternate code position. The word marks and zone bits do not affect the operation and are not lost. If an A-bit is present, a 1301 operation is specified. If a B-bit is present, it is considered an *. If 1301 disk storage units are attached, refer to *IBM 1301 Disk Storage* section.

Disk-Sector Address

The disk-sector address contains the 6-digit address of the first sector to be operated upon. Before any disk operation is performed, an automatic comparison is made of the sector address in core storage with the disk-sector addresses on the specific track. If an equal comparison is made, the operation continues. If no equal comparison is made, the unequal-address compare indicator turns on, and the disk operation is not performed.

When sector operations are performed, the disk-sector address is automatically increased by 1 immedi-

ately following the data transfer of each sector, except under these conditions:

1. track operation being performed.
2. sector-count field reaches the value of 000
3. wrong length record.

When any of these conditions occurs, the disk-sector address is not increased by 1.

Notes:

1. The high-order position of the 6-digit disk-sector address must contain a zero.
2. The other five position of the 6-digit disk-sector address may contain any valid character that has a numeric-bit value of zero through nine.
3. Zone bits over the disk-sector address positions are lost through the adder if any address modification takes place.
4. Word marks over the disk-sector address positions do not affect the operation, but are lost during any operation that involves address modification.

Sector Count

This field indicates the number of sectors to be operated upon during the disk operation. The sector-count field is not used during seek operations. During the transfer of data to or from disk storage, the sector-count field is automatically decreased by 1 immediately following a successful address comparison so that the sector-count field reflects the number of successful address comparisons.

If a sector count of 000 is used when initiating a disk-sector read or write operation, an error condition occurs. Before the first sector is transferred, a 1 is subtracted from the sector-count field. In this case, the result would be 999. Therefore, data would be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong length record and any-disk condition indicators would be turned on.

Notes:

1. Word marks cannot be placed over the sector-count field units position. Word marks in any other position do not affect the operation, but are lost during any operation that affects sector-count modification.
2. Zone bits are always removed from all three positions of the sector-count field.

Basic Disk Operations

The four basic operations performed by the 1311 are seek, read, write, and write disk check.

Seek Operation

The seek operation is initiated by a SEEK DISK instruction, which directs the read/write heads to the proper cylinder on the disk pack. This instruction is followed by a read or write operation.

The data on the disk records is not acted on during this seek operation.

The seek operation positions the access arms over the specified cylinder. The B-address position of the instruction contains the core-storage address of the disk-control field and it is this field that specifies the proper cylinder plus other pertinent information.

Read Operation

The read operation is initiated by one of the three different types of READ DISK instructions, and transfers data from disk storage to a specified area in core storage. (The three types of instructions are discussed following the write-operation description.) The specified disk-storage area involved in the transfer is partially identified by the previous seek operation, and the rest of the area is fully identified before the data transfer takes place. The identification is accomplished by comparing the sector addresses on the disk with the sector address in core storage. The sector address in core storage is part of the disk-control field, and the B-address position of the READ DISK instruction contains the core-storage address of the disk-control field. The data from the disk is placed in a core-storage area located immediately to the right of the disk-control field.

Write Operation

The write operation is initiated by one of the three different types of WRITE DISK instructions, and transfers data from a specified core-storage area into disk storage. (The three types of instructions are discussed following this operation description.) The specific disk-storage area involved in the transfer is partially identified by the previous seek operation, and the rest of the area is fully identified before the data transfer takes place. The identification is accomplished by comparing the sector addresses on the disk with the sector address in core storage. The sector address in core storage is part of the disk-control field, and the B-address portion of the WRITE DISK instruction contains the core-storage address of the disk-control field. The data that is to be transferred to the disk is stored in a core-storage area located immediately to the right of the disk-control field.

Types of Read and Write Operations

Each read or write operation can operate in three different ways, or modes: sector, track sectors with addresses, and sector-overlay modes.

Sector Mode. Read and write operations in the sector mode transfer data, but do not transfer disk-sector addresses. The sector mode is the normal mode of operation. The number of sectors to be handled during one operation is specified by the sector-count portion of the disk-control field. Each sector is transferred only after a correct comparison of the sector address in the core-storage disk-control field is made with the sector address on the disk. For more detailed information, refer to the specific instruction.

Track-Sectors with Addresses Mode. This mode of operation transfers both the data and the disk-sector addresses to and from the disk, one complete track at a time. The mode of operation makes it possible to change the previously recorded sector addresses. The operation requires that the sector-address portion of the disk-control field contain the address of one of the sectors within the specified track, and the sector-count portion of the disk-control field must contain 020 (20 sectors will be transferred). The transfer can only occur after a correct comparison of the sector address in the core-storage disk-control field with a sector address on the specified track. For more detailed information, refer to the specific instruction.

Sector-Count Overlay Mode. This mode of operation allows a portion of the data record itself to specify the number of sectors to be involved in the data transfer. The disk-sector addresses are not involved in the transfer. This mode of operation permits better disk storage utilization for sequential applications involving variable-size records. For more detailed information, refer to the specific instruction.

Reading and Writing with Word Marks Mode. Word mark can be transferred with the data during all reading and writing operations by an L Op code instead of an M Op code. When word marks are written on the disk, the data is written in an 8-bit BCD coding.

Write Disk Check

The write-disk-check operation causes the data in the specified disk area to be compared against the comparable data in the specified core-storage area. When the disk data does not compare, bit-by-bit and character-by-character, with the code-storage data, a disk-error indicator is set ON. This operation normally takes the form of a WRITE DISK CHECK instruction, which must follow each write operation. The write-disk-check operation compares the data written in disk storage with the original source data in core storage.

IBM 1311 Instruction Format and Instructions

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
xx	M/L	%Fn	BBB	R/W

Instructions applying to the 1311 cannot be successfully chained.

Op Code

This is always a single character that defines the basic operation to be performed. Either the M or L operation code can be used with IBM 1311 instructions.

When the M Op-code is used, characters are written or read in the 7-bit mode (CBA 8421). The L Op-code causes characters to be read or written in 8-bit mode (CBA 8421 M). The 8-bit mode provides for a possible word mark with the character being written on, or read from, the disk record.

A-Address

%Fn signals that the disk unit is to be selected; *n* represents the digit used to perform various operations.

n-Position	*Operation*
0	*Seek* a disk record.
1	*Sector*—Reading or writing characters from the number of sectors specified by the sector-count field is stopped when a group mark with a word mark, or the end-of-sector is sensed. If a group mark with a word mark is sensed before the reading of the sector(s) of the track is completed, reading stops and the wrong length record and any-disk condition indicators turn ON. If the group mark with a word mark is sensed before the writing of a record on a disk is completed and it is before the end of a record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk condition and wrong length record indicators are turned ON.
6	*Disk Track-Sector with Addresses*—Allows the reading or writing of a full track (20 sectors) including sector addresses. To perform this operation, the write-address key-light on disk-storage unit 0 must be on. When the write-address light is ON, write-sector operations cannot be performed.
3	*Write Disk-Check*—Data written on a disk in a preceding write operation is read from the disk and compared, character-by-character, with the data in core storage. A WRITE DISK CHECK must be given following a write operation, unless an error occurred *during* the write operation. A write-disk-check operation can be executed after a read operation if a check on the information read is desired. The operation is performed exactly the same as a write-disk-check operation following a write operation.
5	*Sector-Count Overlay*—Allows for records of a variable number of sectors (more than one) to be read or written with a single instruction. The number of sectors to be read/written is controlled by the multiple sector-count field. This control field is in the first three data positions of the first sector of the disk record. This technique permits better disk-storage utilization for sequential applications involving variable-size records. The record itself specifies the number of sectors involved.

B-Address

The B-address specifies the high-order position in core storage of the 10-digit disk-control field. The disk-control field is followed by the area of core storage that is to have data read into or out of by the disk-storage drive. The data area must be followed by a group mark with a word mark.

d-Character

The d-character is used to specify the operation to be performed.

Seek Operation

Seek Disk

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
SD	M	%F0	BBB	R
LU (with word marks)	L	%F0	BBB	R

Note: Because word marks have no meaning for the seek-disk operation, no load-mode Autocoder mnemonic is provided. However, the "general" mnemonic (LU) can be used if an L Op-code is desired.

Function. The A-address specifies that a seek operation is to be performed by the access assembly. The B-address specifies the high-order position in core storage of at least the first six positions of the disk-control field. Only the alternate-code position and the first five positions of the core-sector address are used during a seek-disk operation.

The selected access assembly is first withdrawn from the disks to the home position, and then is moved toward the center of the disk pack. Movement of the mechanism stops when the correct cylinder is reached.

Figure F-22 shows a functional schematic of a seek-disk operation.

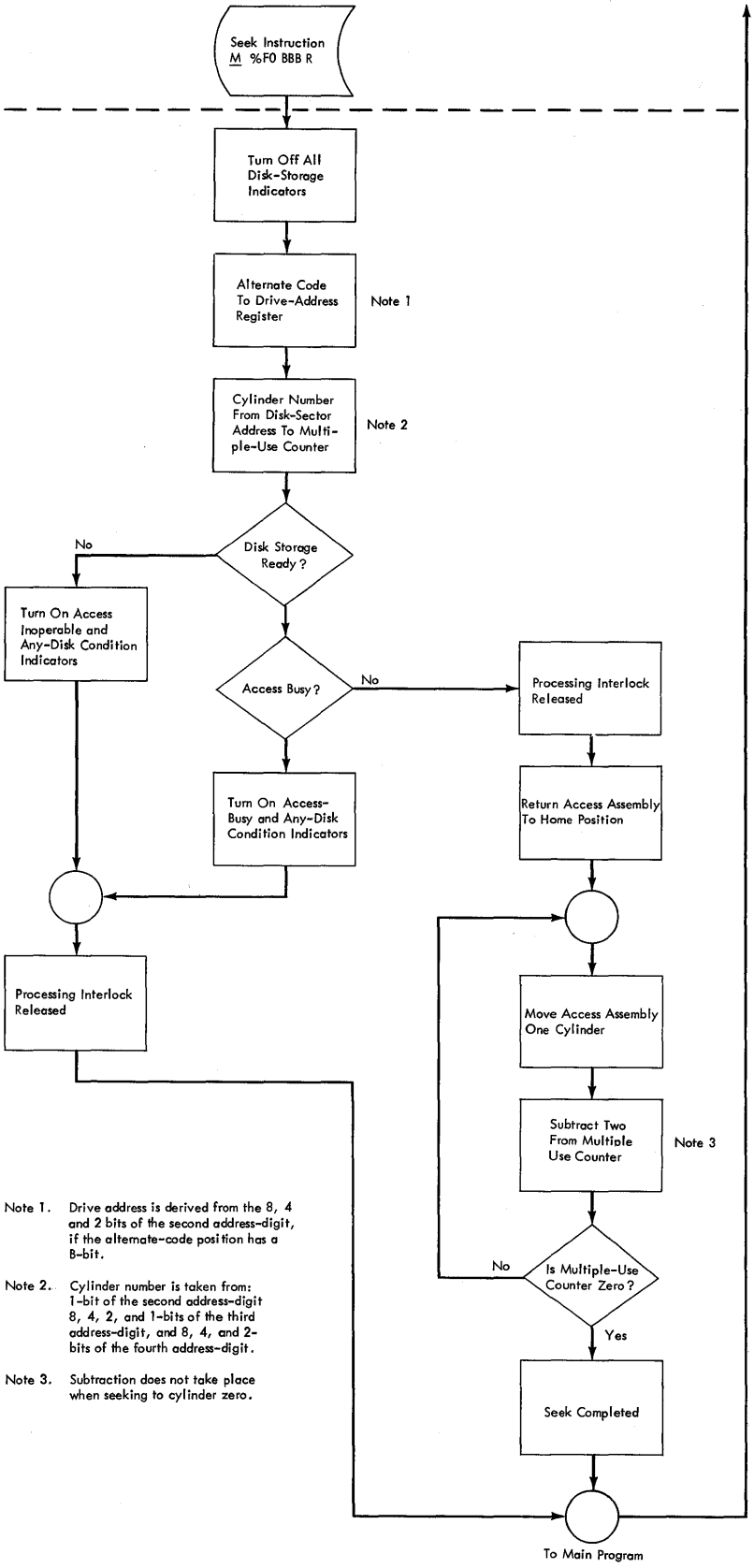
Word Marks. Word marks are not affected.

Timing. T = .1887 ms + access time.*

*400 ms is maximum access time for a seek.

250 ms is average access time for a seek.

Note: If the access mechanism is already at the disk track that is to be used, a SEEK DISK instruction need not be given.



Note 1. Drive address is derived from the 8, 4 and 2 bits of the second address-digit, if the alternate-code position has a B-bit.

Note 2. Cylinder number is taken from: 1-bit of the second address-digit 8, 4, 2, and 1-bits of the third address-digit, and 8, 4, and 2-bits of the fourth address-digit.

Note 3. Subtraction does not take place when seeking to cylinder zero.

Figure F-22. Seek-Disk Functional Schematic

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B+6	B+7

Example. Seek record 015734 with the access assembly. Storage locations 0590-0599 (labeled INPUTA) contains 0015734001 (Figure F-23).

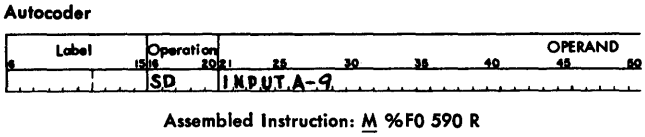


Figure F-23. Seek Disk

Sector Operations

If only the data portion of a disk record is to be affected, the operation is classified as a *sector operation* (addresses are not affected). Disk records can be read, written, or scanned during sector operation. The term *sector operation* does not mean that a disk record is confined to a 100-character sector. The data needed for a record can be written in as many sectors as needed.

Read Disk Sector(s)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RD	M	%F1	BBB	R

Function. This instruction causes data to be read from disk storage into core storage. The digit 1 in the A-address (%F1) specifies that a sector operation is to be performed. The number of sectors to be read is specified by the sector-count field. The reading of the disk is stopped by a group mark with a word mark in core storage and by the end of the sector.

Reading begins at the address contained in the core-sector address field and continues for the number of sectors specified by the sector-count field.

The core-sector address field is *increased* by one for each sector read, and the sector-count field is *reduced* by one as a sector is read.

When the sector-count field reaches 000, an end of operation is indicated to the system. An error condition results from any disk sector read or write operation that begins the operation with a sector count of 000. Before the first sector is transferred, a one (1) is subtracted from the sector-count field, resulting in a sector count of 999. Data would then be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong length record and any-disk condition indicators are turned ON.

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage reserved for the data read from the disk.

The R in the d-character position signifies a read operation.

Refer to Figure F-24 for a functional schematic of a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the disk record. If a group mark with a word mark is detected before reading of the record is completed, the wrong-length-record indicator turns on and reading stops. The position of the group mark with a work mark can be determined by using the formula:

$$GMWM = B + N_S(L_S) + 10$$

B = Address of high-order position of disk address in core storage

N_S = Number of sectors read

L_S = Number of characters per sector

Timing. T = .0999 ms + $2N_S$ + disk rotation.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

Note: Before reading starts, an automatic comparison is made of the core-sector address with the sector address on the disk. This check is made for each sector read. If they are not the same, the unequal-address compare indicator turns ON, and the data on the disk cannot be read into storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B+6	B+11+N_SL_S

Example. Read one sector from disk storage into core storage beginning at location 0600 (labeled

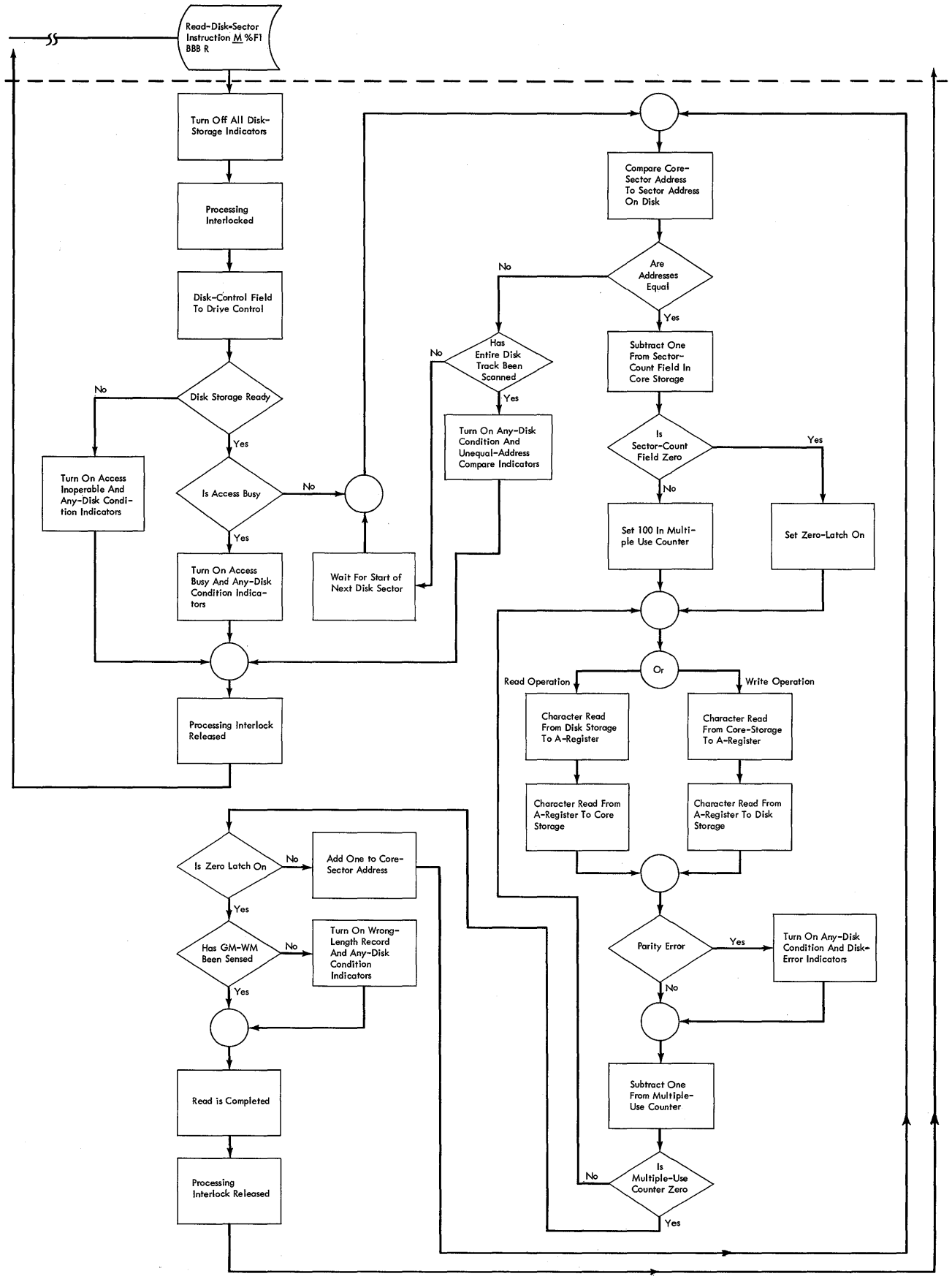
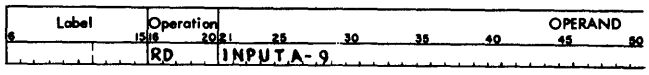


Figure F-24. Read/Write Disk Functional Schematic

INPUTA). In Figure F-25, the disk-control field is located in the ten positions preceding the label (0590-0599).

Autocoder



Assembled Instruction: M %F1 590 R

Figure F-25. Read Disk Sector

Read Disk Sector(s) with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDW	L	%F1	BBB	R

Function. This is similar to the READ DISK SECTOR instruction except that

1. word marks in the record area of core storage are removed, and
2. word marks from the disk record are written in core storage. The length of the sector read from disk storage into core storage is 90 positions.

Word Marks. A group mark with a word mark in core storage terminates the read operation. If the group mark with a word mark is not in the position to the right of the last character read from the disk into core storage, the wrong length record and any-disk condition indicators turn on.

Timing. $T = .0999 \text{ ms} + 2N_S + \text{disk rotation.}$*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

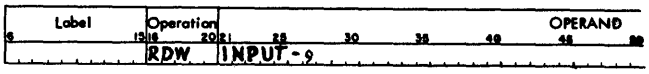
Note: If a disk is read in a mode different from the one in which it was written (M or L operation code), a parity error occurs. The disk-error indicator turns ON.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	$B + 11 + N_S L_S$

Example. Read a record, with its associated word marks, from disk storage into the area labeled INPUT (first position of data is at 0600). In Figure F-26, the disk-control field is located in the ten positions preceding the label (0590-0599).

Autocoder



Assembled Instruction: L %F1 590 R

Figure F-26. Read Disk Sector with Word Marks

Read Disk with Sector-Count Overlay

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDCO	M	%F5	BBB	R
RDCOW (with word marks)	L	%F5	BBB	R

Function. This operation is similar to the READ DISK SECTOR(s) instruction except that the number of sectors to be read is controlled by the first three positions in the first record read. The digit 5 in the A-address specifies that an overlay operation is to be performed.

As the first sector is read from disk storage, the first three digits of the record being read are placed in the sector-count field of the disk-control field in core storage. Therefore, if a variable number of sectors are to be read from disk storage, the sector-count field must contain a value greater than 001 to cause the first sector to be read. The first three positions of the first sector read contain the number of additional sectors to be read. Figure F-27 illustrates the operation of an overlay instruction which causes four sectors of data to be read from disk storage into core storage.

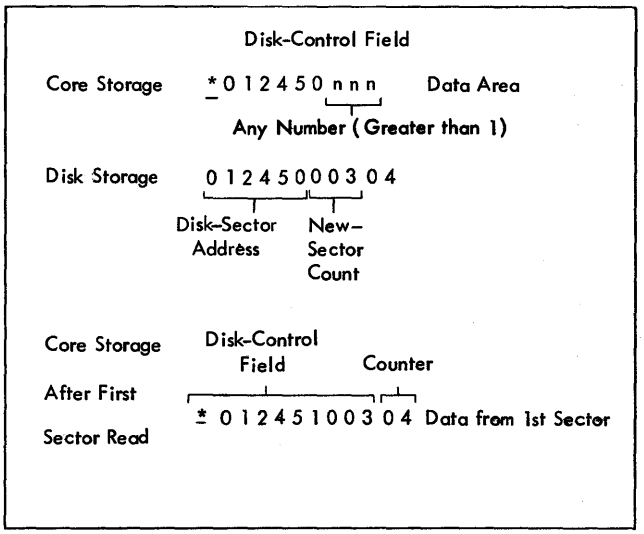


Figure F-27. Read Disk — Sector-Count Overlay Operation

The operation proceeds as a normal read operation with appropriate changes to the core-sector address and sector-count fields.

Word Marks. Because the exact number of positions of data to be read from disk storage may not be known when this operation is initiated, place the group mark with a word mark (signaling the end-of-operation) one position to the right of the last possible character to be read using this instruction. If the maximum number of records is not read, the read into storage stops because the end of sector is reached and the sector-count field is all zeros before the group mark with a word mark is sensed. The wrong-length-record indicator also turns ON. The programmer can check core storage in this case to see if the correct number of sectors has been read.

This can be accomplished by setting up a counter in the fourth and, if necessary, fifth position of the first sector of the record. This counter, when the read operation is completed, is located in the first and/or second position of the data record in core storage. These positions can be used to check the number of sectors in the record. These counter positions should equal the actual number of sectors in the record. For any record length other than single-sector records, reading data from disk should have stopped at $B + 6 + N_S L_S$. If it did not, an error did occur and appropriate action should be taken. If a correct read has occurred, the error indication can be disregarded.

Special consideration must be given to single-sector records when read in the sector-count overlay mode. When the read operation begins, the first three characters of the record overlay the sector count. In this case, 000 is read in and overlaid. However, the machine does not detect a zero sector count except when produced by automatically decreasing the sector-count field. After reading the single-sector record, the address is increased by one and an equal compare is sought on the next sector. When found, the sector-count field is decreased by one again, resulting in a count of 999. Because the sector-count field is not all zeros when this occurs, the wrong-length-record indicator is turned on if a group mark with a word mark is placed in the 101st position of the data field.

When an initial sector count of 002 or greater is used and the first three digits of the first sector being read are 000 (the three digits to be overlaid), the following occurs:

1. The operation does *not* stop because the sector count has not been decremented to 000.
2. The sector address has been incremented and the second sector is read.
3. The special-add operation (used to keep track of the sector count) decrements the sector count (000) to 999.

Because the last step (item 3) does not produce a carry to increment the sector address, an unequal-address compare occurs on the attempt to read the third sector. The unequal-address compare will not occur if the initial sector count was 001, and only one sector will be read into core storage because the sector count has been decremented to 000.

Single-sector and *multiple* sector-read operations should not be interspersed (using M/L %F5 BBB R instruction) without prior knowledge of exactly *when* each read will occur.

When a file includes single-sector records, a special routine must be included to verify the validity of the record read. Before executing a read, a special character that would never be found in the last position of a record can be moved to the 100th position of the input area. The wrong-length-record routine can then check to see whether the counter in the first position of the record contains a one (1). If so, it would check to see that the special character has been overlaid. If it has, the record was read in its entirety.

Timing. T = .0999 ms + $2N_S$ + disk rotation.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

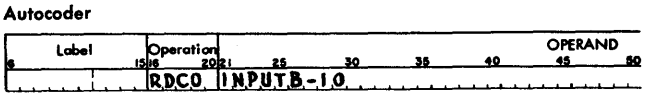
Note: Before reading starts, an automatic comparison is made of the record address in core storage with the record address on the disk. This check is made as each sector is read. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data on the disk cannot be read into storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	$B + 8 + N_S L_S$

Example. Read into core storage a variable number of sectors that contain the data for a record beginning at location 0900 (labeled INPUTB). In Figure F-28, the disk-control field is located in the ten positions preceding the label (0890-0899).

Autocoder



Assembled Instruction: M %F5 890 R

Figure F-28. Read Disk with Sector-Count Overlay

Write Disk Sector(s)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WD	M	%F1	BBB	W

Function. This instruction causes record data in core storage to be written on a disk record. The digit 1 in the A-address (%F1) specifies that a sector operation is to be performed. The number of sectors to be written is specified by the sector-count field. The writing of the disk record is stopped by a group mark with a word mark in core storage and by the end of sector.

Writing begins at the address contained in the core-sector address field and continues for the number of sectors specified by the sector-count field.

The core-sector address field is *increased* by one for every sector written. The sector-count field is *reduced* by one as a sector is written.

When the sector-count field reaches 000, an end-of-operation is indicated to the system. An error condition results from any disk sector read or write operation that begins the operation with a sector count of 000. Before the first sector is transferred, a one (1) is subtracted from the sector-count field, resulting in a sector count of 999. Data would then be transferred until a group mark with a word mark is encountered in core storage. Because the sector count is not zero at this time, the wrong length record and any-disk condition indicators are turned on.

The B-address specifies the high-order position in core storage of the disk-control field, and is followed by the data to be written on the disk.

The W in the d-character position signifies a write operation.

Refer to Figure F-24 for a functional schematic of a write operation.

Word Marks. A group mark with a word mark must be *one* position to the right of the last character of the record in core storage. The writing of data stops when the end-of-record is reached on the disk and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end of a record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk condition and wrong-length-record indicators are turned ON.

Timing. T = .0999 ms + $2N_S$ + disk rotation.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

Notes: Before writing starts, an automatic comparison is made of the core-sector address with the record address on the disk. This check is made for each sector written. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data in storage cannot be written on the disk.

If the data in core storage contains characters with word marks, only the CBA8421 portion of the character is written on the disk (the word mark is ignored).

A WRITE DISK CHECK instruction must be performed following a write disk operation unless an error occurred during the write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + N_SL_S

Example. Write a disk record (one sector) from the data in the area labeled INPUTA (first position of data is at 0600). In Figure F-29, the disk-control field is located in the ten positions preceding the label (0590-0599).

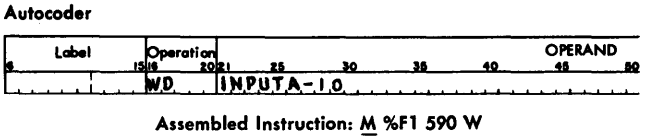


Figure F-29. Write Disk Sector

Write Disk Sector(s) with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDW	L	%F1	BBB	W

Function. This instruction is similar to the WRITE DISK SECTOR instruction, except that word marks set with the data in core storage are recorded on the disk record. This mode of operation permits writing programs on disk records for system use. Ninety positions of data with word marks are recorded on each sector during the write operation.

Word Marks. A group mark with a word mark one position to the right of the last character of the record in core storage terminates the write operation. If the group mark with a word mark is not sensed at the same time as the end-of-a-record, the remainder of the disk record is filled with valid blanks (C-bit), and the any-disk condition and the wrong-length-record indicators are turned on.

Timing. T = .0999 ms + $2N_S$ + disk rotation.*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

Notes: The programmer should be certain that all records on a specific track are written in the same mode (M or L operation code). Otherwise, track operations are not possible.

Before writing starts, an automatic comparison is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare indicator is turned on, and the data in storage cannot be written on the disk. A write-disk-check operation must be performed following this instruction.

Address Registers After Operation.

I-Add. Rg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + N_SL_S

Example. Write a disk record, with word marks, from the data in the area labeled OUTPUT (first position of data is 0600). In Figure F-30, the disk-control field is located in the ten positions preceding the label (0590-0599).

Autocoder

Label	Operation	OPERAND
	WDW	OUTPUT-10

Assembled Instruction: L %F1 590 W

Figure F-30. Write Disk Sector with Word Marks

Write Disk with Sector-Count Overlay

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDCO	M	%F5	BBB	W
WDCOW (with word marks)	L	%F5	BBB	W

Function. This operation is similar to the WRITE DISK SECTOR instruction except that the sector-count field of the disk-control field is automatically decreased by one and then written in the first three data positions of the first sector written. The digit 5 in the A-address specifies that an overlay operation is to be performed.

Therefore, if a variable number of sectors are to be written on disk storage, the sector-count field in core storage should contain the number of sectors to be written. The first three data positions of the first sector written contain the number of additional sectors that were written. Figure F-31 illustrates the operation of an overlay instruction, which causes four sectors of data to be written from core storage onto disk storage.

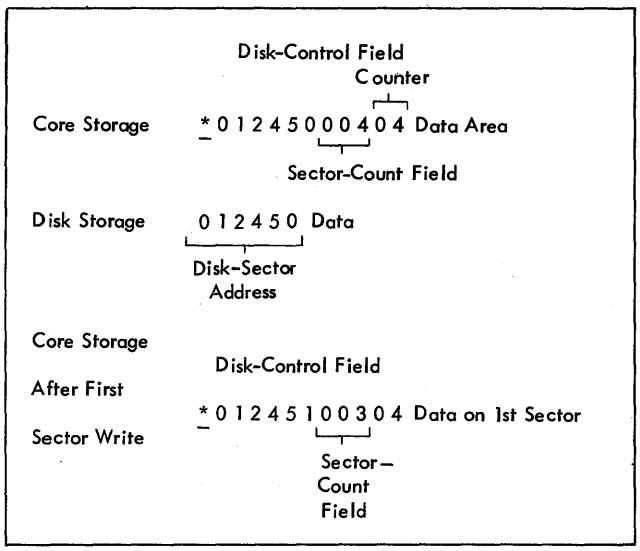


Figure F-31. Write Disk – Sector-Count Overlay Operation

The operation proceeds as a normal write operation with appropriate changes to the core-sector address and sector-count fields.

Word Marks. A group mark with a word mark should be placed one position to the right of the last sector to be written. The group mark with a word mark must be placed at B + 7 + N_SL_S to avoid a false wrong-length-record indication.

Timing. T = .0999 ms + $2N_S$ + disk rotation.*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

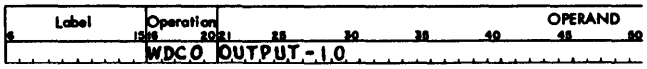
Note: Before writing starts, an automatic comparison is made of the record address in core storage with the record address on the disk. This check is made before each sector is written. If the addresses are not the same, the unequal-address-compare indicator is turned on, and the data in core storage cannot be written in disk storage.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 8 + N_SL_S

Example. Write a number of sectors for a record on disk storage that contains data beginning at location 0900 (labeled OUTPUT). In Figure F-32, the disk-control field is located in the ten positions preceding the label (0890-0899).

Autocoder



Assembled Instruction: M %F5 890 W

Figure F-32. Write Disk with Sector-Count Overlay

Write Disk Check

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDC	M	%F3	BBB	W
WDCW (with word marks)	L	%F3	BBB	W

Function. This instruction causes a comparison, character-by-character, of the data in core storage with the data just written on the disk. This instruction must be executed after a write operation and before any other disk-storage operation is initiated.

The digit 3 in the A-address specifies that a WRITE DISK CHECK is to be performed. Either an L or M operation code is used, depending on how the data was recorded in disk storage.

The B-address specifies the area in core storage that contains the disk-control field and the data recorded on the disk.

The sector-address and sector-count fields of the sector-control word must be restored to the values present at the beginning of the write operation.

Word Marks. A group mark with a word mark must appear one position to the right of the disk data in core storage.

Timing. T = .0999 ms + $2N_S$ + rotation time.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

Notes: If the disk address in core storage is not the same as the address on the record, the unequal-address compare indicator turns ON. If any of the characters on the disk record do not agree with the characters in core storage, the disk-error indicator turns on.

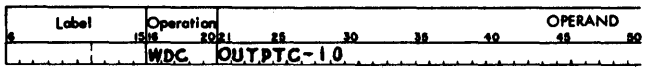
A write-disk-check operation can be executed after a read operation if a check on the information read is desired. The operation is performed exactly the same as a write-disk-check operation following a write operation.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Depends on previous operation	

Example. Compare the disk record with a record in core-storage area labeled OUTPTC (beginning at 0700). In Figure F-33, this disk-control field is located in the ten positions preceding the label (0690-0699).

Autocoder



Assembled Instruction: M %F3 690 W

Figure F-33. Write Disk Check

Address Operations

The disk sector addresses written on the disk pack are protected from improper systems operation by the write-address key-light on disk-drive 0. When the light is off, addresses on the disk pack cannot be altered or read into core storage. If the light is on, disk-pack addresses can be read into core storage and new addresses can be written on the disk pack.

The ability to read and/or alter disk addresses is conditioned by the setting of the write-address key-light and the stored program instructions. Certain IBM 1311 instructions are used when reading or writing disk addresses. These instructions contain the term *Address* in their description and a note on the setting of the write-address key.

If the proper instruction and key setting are not used when trying to perform an address operation, the system stops and the RAMAC light on the system console turns ON.

Read Disk Track Sectors with Addresses

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDT	M	%F6	BBB	R
RDTW (with word marks)	L	%F6	BBB	R

Function. This instruction causes the contents of an entire disk track (addresses and data) to be read in

the mode specified by the operation code (M or L). If the L operation code is used, the track is read into storage with its associated word marks.

The core-sector address must correspond to any one of the sector addresses on the track. The disk track is scanned for an address equal to the sector address in core storage. The disk-track reading begins when the first track-index pulse following a successful address-compare operation is sensed. If the result of the address-compare operation is unequal, the unequal-address compare indicator turns on. All twenty sectors on the track, including the disk-sector addresses, are read into core storage. The sector-count field of the disk address must be set at 020 before the operation begins.

The core-sector address field is not modified by plus-one during this operation. To keep track of the number of sectors read, however, the sector-count field is modified by minus-one for each sector read.

Word Marks. A group mark with a word mark must be placed one position to the right of the last character read into core storage. This position can be found by adding 2130 to the B-address for operations performed with the M operation code, and 1930 for operations performed with the L operation code.

Timing. T = .0999 + 42 ms + disk rotation.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

Note: The write-address key-light on disk-storage-drive zero must be on to perform the operation. When the key-light is off, disk-sector operations cannot be performed.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2120 (M Op code) or B + 11 + 1920 (L Op code)

Example. Read disk track 17550, with its associated word marks, into the core-storage area labeled RDTSAD (first position of data is at 0800). In Figure F-34, the disk-control field is located in the ten positions preceding the label (0790-0799).

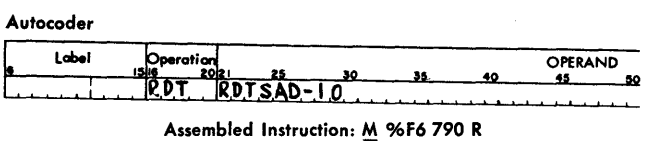


Figure F-34. Read Disk Track Sectors with Addresses

Write Disk Track Sectors with Addresses

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-Address*	*d-character*
WDT	M	%F6	BBB	W
WDTW (with word marks)	L	%F6	BBB	W

Function. This instruction causes the record data and addresses in core storage to be written on a disk track in the mode specified by the operation code (M or L). If the L operation code is used to write the track, word marks in the record area of core storage are written on the track.

The core-sector address must correspond to any one of the sector addresses on the track. The disk track is scanned for an address equal to the sector address in core storage. Writing the disk track begins when the track-index pulse is sensed (signaling first sector on track), if the address-compare operation results in an equal condition. If the result of the address-compare operation is unequal, the unequal-address compare indicator turns ON. All 20 sectors on the track, including the disk-sector addresses, are written on disk storage. The sector-count field of the disk-control field must be set at 020 before the operation begins.

The core-sector address field is not modified by plus-one during this operation. To keep track of the number of sectors written, however, the sector-count field is modified by a minus for each sector read.

Word Marks. A group mark with a word mark must be placed one position to the right of the last character written on disk storage. This position can be found by adding 2130 to the B-address for operations performed with the M operation code, and 1930 for operations performed with the L operation code.

Timing. T = .0999 + 42 ms + disk rotation.*

*42 ms is maximum time for disk rotation.

22 ms is average time for disk rotation.

2 ms is minimum time for disk rotation.

Note: The write-address key-light on disk-storage-drive zero must be on to perform this operation. When the key-light is off, disk-sector operations cannot be performed.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2120 (M Op code) or B + 11 + 1920 (L Op code)

Example. Write a disk track (with address) from the data in the area labeled WRTSAD (first position of

data is at 1000). In Figure F-35, the disk-control field is located in the ten positions preceding the label (0990-0999).

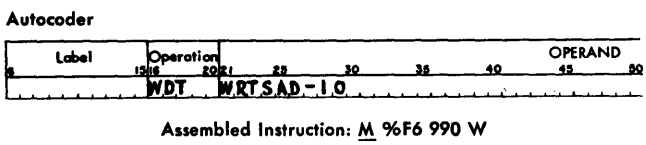


Figure F-35. Write Disk Track Sectors with Addresses

Branch if Indicator On

The BRANCH IF INDICATOR ON instruction tests the indicators that might be set ON during a disk-storage operation. When a disk-storage instruction occurs in the program, it turns off all disk-storage indicators that were turned on by a previous disk-storage operation. The execution of a disk-storage instruction can result in a disk-storage indicator being turned on.

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. The d-character specifies the indicator tested. If the indicator is ON, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken. Figure F-36 shows symbols that are valid d-characters, and the indicators they test. More than one indicator can be turned ON as the result of a disk-storage operation.

Indicators

Access Inoperable. An access arm becomes inoperable if the logic safety circuit detects improper operation. A customer engineer can also render an arm inoperable. In either instance, this indicator turns on when the program addresses the inoperable arm, at which time the operation is terminated and the next sequential instruction is started. At the same time, the RAMAC light turns on.

d-Character	Indicator
N	Access Inoperable
V	Validity Error
W	Wrong-Length Record
X	Unequal-Address Compare
Y	Any Disk Condition
\	Access Busy

Figure F-36. d-Characters for Branch If Indicator On Instruction

The indicator also turns on if an invalid (not installed) arm or disk-storage unit is addressed. Because the program continues in sequence even when an inoperable arm is addressed, a BRANCH IF INDICATOR ON instruction must immediately follow a seek instruction.

Disk Error. This indicator turns on if even-bit parity occurs during reading or writing on a disk. Another condition that turns the indicator on is an UNEQUAL COMPARE during a write-check operation. In this case the operation is completed.

Wrong Length Record. This indicator turns on if the following conditions are not satisfied: a group mark with a word mark in core storage is sensed at the same time as an end-of-sector and an all-zero condition in the sector-control field occur. It also turns on during a scan operation if the search argument is longer than, or equal to, a sector length. Detection of a wrong length record terminates the operation and starts the next sequential instruction.

Unequal-Address Compare. An unequal-address-compare condition occurs during the automatic comparison of the sector address in storage with the sector address on the disk. This unequal condition turns the unequal-address-compare indicator on after the disk track is scanned and the track-index pulse is sensed twice. Each sector operated on by a disk-storage read-write instruction is checked for ADDRESS COMPARE. This is an automatic check and does not have to be programmed. During multiple-sector operations, the indicator also turns on after the data transfer begins when the sector address following a correct address comparison does not compare.

The internal circuitry is the same as that used by the COMPARE instruction. In programming, be careful that a normal-compare operation and the address-compare operation do not interfere with the settings of the equal-, low-, and high-compare indicators set by a previous instruction. Detection of an unequal-address compare terminates the operation and starts the next sequential instruction.

Any-Disk Condition. This indicator turns on if any of the other disk-storage indicators are on. It can be tested by the program, and, if it is off, the program can proceed. If this indicator is on, check the other indicators to determine where corrective measures should be taken.

Access Busy. This indicator is turned off by every disk-storage command. If the access assembly is in motion, the indicator is turned on and the instruction is not executed.

A branch-access-busy command tests the indicator. If on, the previous disk-storage instruction was not executed and should be repeated.

With the seek-overlap feature, an indicator is provided for each drive attached to the system. Without the feature there is only one indicator.

The following examples illustrate the access-busy-operation.

With Seek Overlap

Operation	*Remarks*
Start	Begins program execution.
Seek Access 0	Turns indicator (0) off Test for access motion – no motion Indicator (0) remains off Instruction executed.
Seek Access 2	Turns indicator (2) off Test for access motion – no motion Indicator (2) remains off Instruction executed.
Branch Access Busy	No branch

Without Seek Overlap

Operation	*Remarks*
Start	Begins program execution.
Seek Access 0	Turns indicator off Test for access motion – no motion Indicator remains off Instruction executed.
Seek Access 2	Turn indicator off Test for access motion – motion Indicator is turned on Instruction is not executed.
Branch Access Busy	Branch

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Note: After each disk unit read or write operation, the program must test for error indications to prevent processing of unusable data.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. At the completion of a disk-read operation, test the any-disk-unit error condition indicator. If it is OFF, continue in the main program. If it is ON, branch to the routine labeled DISKER (0690) to determine the type of error condition. This tests all disk-unit indicators and branches to the error routine of the respective indicator that is on. In Figure F-37, the routines are labeled: ACINOP (0690), UNADCL (0695), WRLENR (0700), RWPARC (705).

Autocoder

Label	Operation	OPERAND
	B.I.N.	D.I.S.K.E.R.,Y.
D.I.S.K.E.R.	B.I.N.	A.C.I.N.O.P.,N.
	B.I.N.	U.N.A.D.C.L.,X.
	B.I.N.	W.R.L.E.N.R.,W.
	B.I.N.	R.W.P.A.R.C.,V.

Assembled Instruction:
480 B 690 Y
690 B 740 N
695 B 790 X
700 B 890 W
705 B 990 V

Figure F-37. Branch If Indicator On Testing Routine

IBM 1311 Disk Storage Drive Timing

The organization of data in disk storage and the method of processing data affect the seek time for a given operation and also affect the total systems' throughput. Some methods of seeking records and the sequence of disk storage and input/output instructions are considered here as an aid to program development.

Seeking Disk-Storage Records

Two modes of operation for seek instructions are: *Return-to-Home* and *Direct Seek,* a special feature. The return-to-home mode is the standard mode of operation. In this mode, all seeks are achieved by first moving the access arms to a *home* position outside cylinder 00 and then counting into the desired cylinder. This function is automatically performed by the system. The direct-seek special feature enables the programmer to write the program so that the system can seek from one track to another track without requiring the access arms to return to home position.

Another factor to be considered in systems planning is that the access arms move at both a low speed and a high speed. Access-arm movement within ten cylinders is at low-speed rate of 2 inches per second. If more than ten cylinders are searched, the access arms move at the high-speed rate of 16 inches per second for all cylinders in excess of ten. These two speeds (2 inches and 16 inches per second) are not used by the programmer in timing disk-storage operations because the timing charts incorporate these variations in speed. Variation in speed is covered here so it can be considered when data is being organized in disk storage.

TO ↓ / FROM	00	09	19	29	39	49	59	69	79	89	99
00	75	88	101	114	127	140	153	167	179	192	204
09	175	188	201	214	227	240	253	267	279	292	304
19	143	156	169	182	195	208	221	235	247	260	272
29	153	166	179	192	205	218	231	245	257	270	282
39	168	181	194	207	220	233	246	260	272	285	297
49	184	197	210	223	236	249	262	276	288	301	313
59	200	213	226	239	252	265	278	292	304	317	329
69	215	228	241	254	267	280	293	307	319	332	344
79	232	245	258	271	284	297	310	324	336	349	361
89	248	261	274	287	300	313	326	340	352	365	377
99	263	276	289	302	315	328	345	355	367	380	392

Figure F-38. Cylinder Seek Time without Direct Seek

After a SEEK DISK instruction in either mode has been issued, processing can continue until another disk-storage instruction is issued. The length of the seek depends on the total number of cylinders that must be passed during the seek operation. Figure F-38 provides actual seek time for cylinder-to-cylinder movement in increments of ten cylinders.

In the return-to-home mode, the total throughput time can be reduced by using a technique known as *dummy seek to cylinder 00.*

The total time for this operation is 106 ms, for approximately 2⅝ disk revolutions. The available processing time is 68 ms.

Processing time is reduced as more sectors are read or written. The timing for a 4-sector operation illustrates this point:

Read	2 ms head select delay time
	20 ms average rotational time
	8 ms to read four sectors
Process	30 ms processing
Write	2 ms head select delay time
	8 ms to write four sectors
Process	30 ms processing
Write Check	2 ms head select delay time
	8 ms write check
	110 ms Total

A summary of the disk-storage times follows:

Rotational Delay	40 ms
Average Rotational Delay	20 ms
Head Select Delay	2 ms
Read One Sector	2 ms
Write One Sector	2 ms
Write Check One Sector	2 ms

Seek time — without direct access

Maximum	400 ms
Mean Seek Time	250 ms

Seek time — with direct access

Maximum	250 ms
Mean Seek Time	150 ms

The total time in the preceding example is 110 ms (2¾ revolutions), only 4 milliseconds longer than the 2-sector operation. However, total processing time is 60 ms as opposed to 68 ms in the earlier example.

If possible, processing should be kept within the available rotational time. If not, the cycle is increased by one 40-ms revolution for each extension of available processing time.

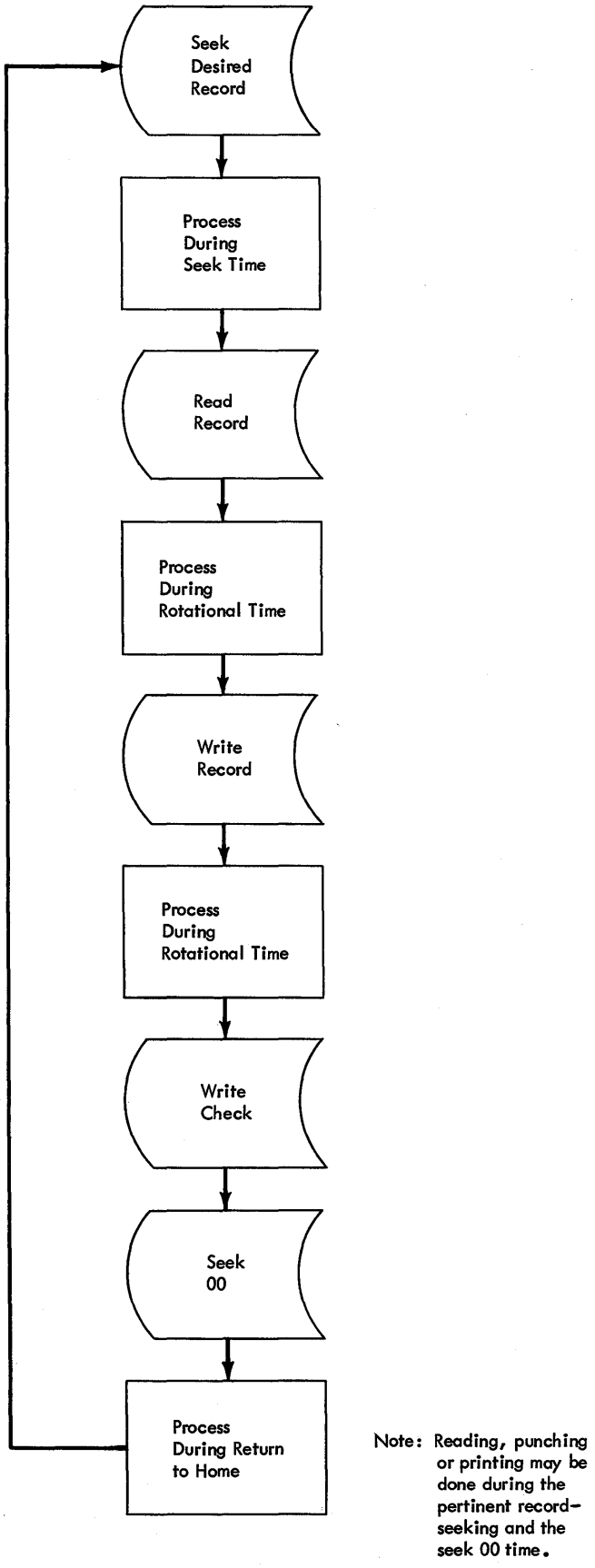


Figure F-39. Block Diagram for Dummy Seek Technique

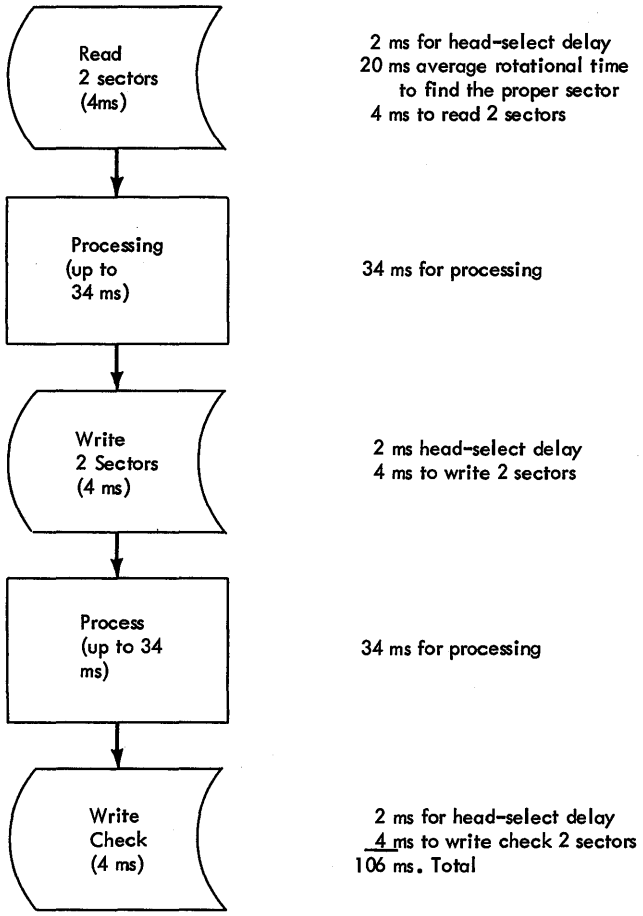


Figure F-40. Disk-Storage Timing for a Two-Sector Record

Processing time between a write operation and a WRITE DISK CHECK instruction can be used for updating control totals and/or arranging fields of printing. When the print-storage special feature is installed, most disk operations may be completely overlapped by the printing operation.

Dummy-Seek to Cylinder 00

Return-to-home seek has two access motions: return-to-home, and advance-from-home. Over-all job-time may be reduced if some other I/O or CPU operation is required before the next seek address is given. In this case, a seek to cylinder 00 will direct the access mechanism back to the home position until it is needed. When the next seek is initiated, the access mechanism is already at the home position, and need only travel directly to the correct cylinder (Figure F-39).

Timing Considerations for Reading and Writing

When designing a program utilizing the disk pack, the programmer should consider ways to place read, write, and write-check operations to save job time. Because the disks revolve at 1500 rpm, 40 ms are required to complete a revolution, and 2 ms to read or write one sector. The rotational time that must elapse before a disk operation can be executed should be utilized for processing, if possible.

Assume, for example, that a 2-sector record (200 characters) is to be read, updated, and then returned to the file. The timing chart and block diagram for this operation are shown in Figure F-40.

IBM 1311 Error Routine

Figure F-41 shows the correct method of programming input/output operations on the IBM 1311 Disk Storage Drive. The method presented is, basically, the routine generated by the IBM 1440 Input/Output Control System. Explanation of the notes in Figure F-41 are:

Note 1

Where possible, utilize seek time by including a processing routine in the busy loop.

Note 2

IOCS does not test for ANY DISK ERROR after the seek and write operations. If an error occurs at either of these points, it is caught later. Tests for ANY DISK ERROR can be made after every BUSY test, however, and can often be justified by the ability to locate more easily the cause of the error.

Note 3

When the direct-seek special feature is used, the disk-control field contains a number equal to twice the number of cylinders to be crossed. If a number is used that causes the access mechanism to attempt to go past the last, or 100th cylinder, the disk drive remains in a BUSY status until manually turned off and then back on. A machine malfunction or programming error may allow a reverse direct seek to go past the first cylinder and cause a system hang-up. In either case, the cause of the programming or machine error must be corrected. In testing programs using the direct seek, it is recommended that the busy loops after all disk input/output instructions include a routine that halts the system after a length of time has elapsed sufficient for the longest possible seek operation.

Note 4

Although not noted in this block diagram, the contents of the address registers at the time of a halt should uniquely identify the cause of the halt.

Note 5

The sequence of tests shown is justified by the fact that:

1. In the event of cylinder overflow, checking parity first ensures that the portion read or written is correct.
2. In the event of cylinder overflow, both the unequal-address compare and the wrong-length-record indicators are on. If only the wrong-length-record indicator is on, the error must be a true wrong-length-record error.

Note 6

If cylinder overflow is encountered, the three low-order digits of the address in the disk-control field will be 200, 400, 600, 800, or 000.

Note 7

None of the IBM programming systems will produce a block that overflows from one disk pack to another. However, IOCS can accommodate such a block in an input file. If the condition occurs when processing labeled files, the program must add 20 to the address in the disk-control field and change the drive number in the alternate-code position before branching to the SEEK instruction.

Note 8

When using the direct-seek special feature, the error routine should include a separate, return-to-home seek instruction instead of going back to the common SEEK DISK instruction of the main program. The reason for this is that when using the DIRECT SEEK, the program must be sure of the starting point of the seek. Because an error condition exists, assume that the program is not sure of the present position.

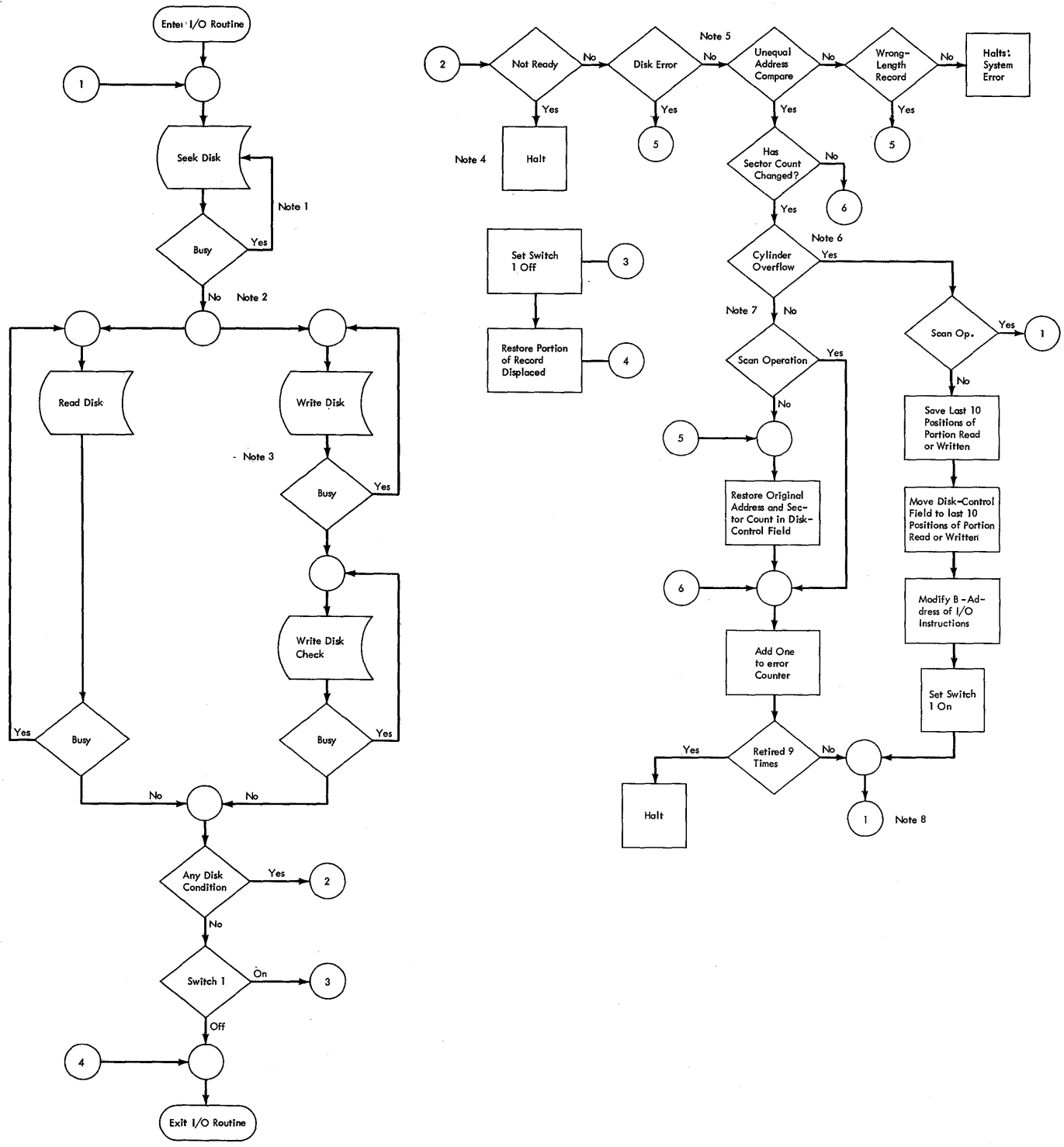


Figure F-41. IBM 1311 Operation and Error Routine

IBM 1009 Data Transmission Unit

The IBM 1009 (Figure G-1) permits high-speed two-way communication between the IBM 1440 Data Processing System and any of the following terminals:

IBM 1013 Card Transmission Terminal
IBM 7701 Magnetic Tape Transmission Terminal
IBM 7702 Magnetic Tape Transmission Terminal
IBM 7710 Data Communication Unit
IBM 7711 Data Communication Unit
IBM 7740 Communication Control System
IBM 7750 Programmed Transmission Control
or another 1009 attached to another 1440, 1401, 1460, 1410, 7080, or other system.

With this unit, the 1440 system can transmit at speeds up to 600 characters per second over toll or leased communications-company lines. Refer to *IBM 1009 Data Transmission Unit*, Form A24-1039, and *IBM 1009 Special Features* section of this publication.

IBM 1440 Programming Logic

When a terminal is made up of a 1009 and a 1440 system, transmitting and receiving follow set patterns.

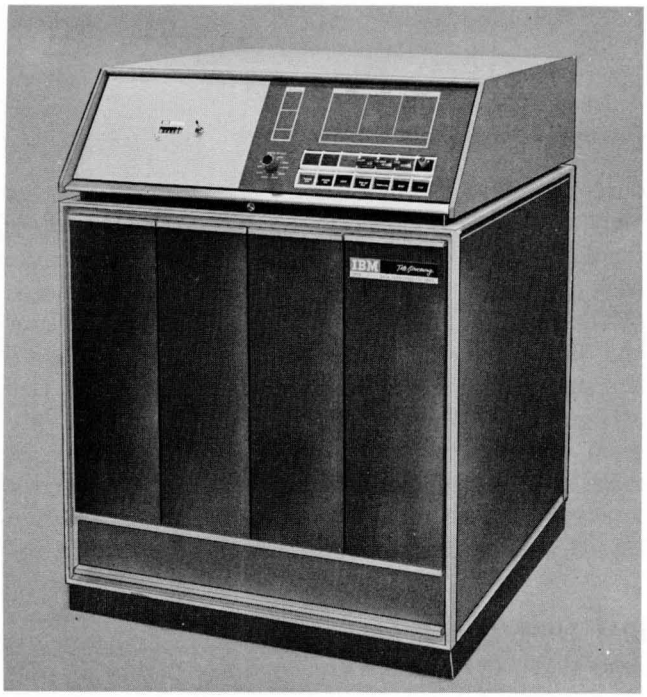


Figure G-1. IBM 1009 Data Transmission Unit

Block diagrams of the logic are provided as programming aids.

Transmit Subroutine

Before the transmitting 1440 program moves the first message from the cards or tape and assembles it in the read-out area, it first tests to see that the receiving 1009 is ready to accept data. This is done by testing indicators 3 and/or 4 (Figure G-2). Indicator 4 is tested in case the last message of the previous transmission was received in error. Then a U %D1 E instruction starts the transmission of the message. One character is transmitted at a time to the receiving 1440, through the two IBM 1009 Data Transmission Units connected to the 1440 system. Before each character is sent to the transmitting 1009, the 1440 checks for a group mark with a word mark that signals the end-of-message. If there are more characters in the message, the 1440 program increases the B-address of the move or load instruction that stores the character in the 1009, and repeats the transmitting process. When the transmitting 1440 encounters an end-of-message signal, it must wait for 250 ms (milliseconds) while the receiving 1440 sends back a good-transmission or transmission-error signal. The transmitting 1440 can use these 250 ms for any processing that does not call upon the 1009. This maximum delay of 250 ms, commonly referred to as *turn-around time*, is substantially reduced on short-distance transmissions. On a four-wire (full duplex) service, the turn-around delay is eliminated.

The next part of the subroutine includes two tests for the status of the message. See *Branch if Indicator On* instruction for an explanation of this test loop.

When a good-message condition is recognized, the program branches to initialize and load a new message.

Receive Subroutine

The receiving 1440 program first tests to see that the 1009 is in a RUN condition (Figure G-3). Then it prepares the read-in area, and sets up conditions for ready-to-receive. This includes acknowledging the previous message, it any. Depending upon the type of communications-company data set used, anywhere from 200 to 300 ms elapse before the first character is received. This allows for turn-around-time when half-

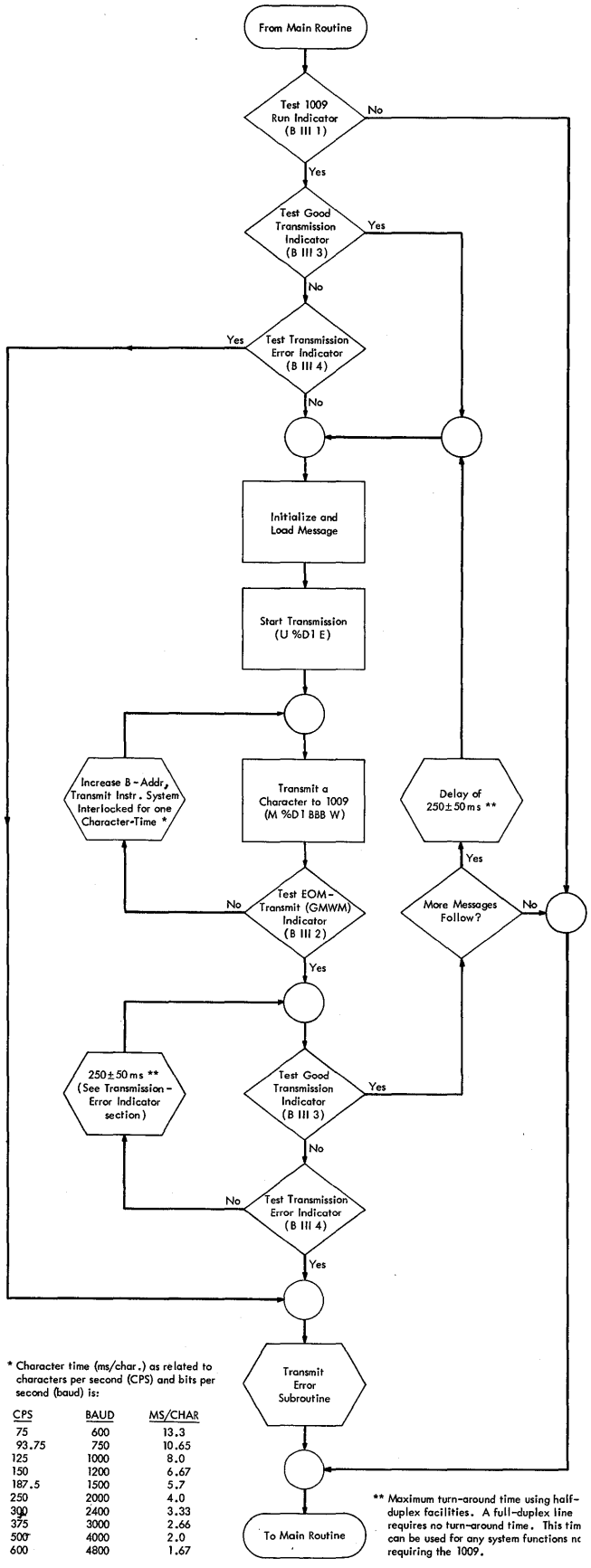


Figure G-2. Transmit Subroutine

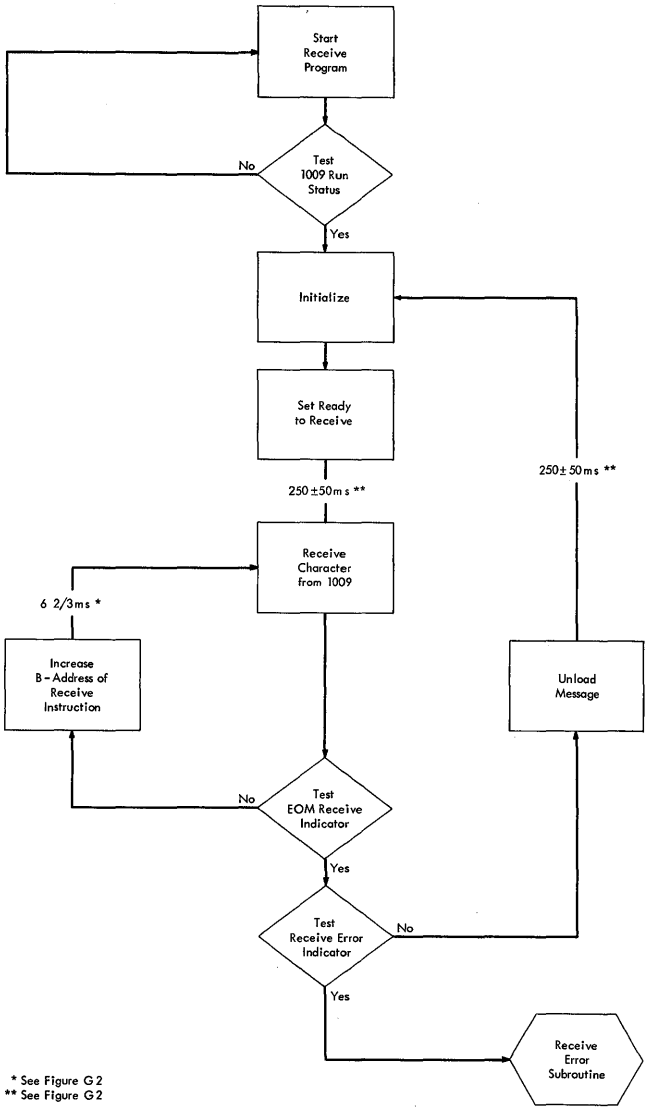


Figure G-3. Receive Subroutine

duplex facilities are used. If the communications-company transmission facilities are full-duplex (4-wire), there is virtually no turn-around-time. After it receives each character, the 1009 checks for an end-of-message (EOM) signal. The receiving 1440 tests the indicator and, if the indicator is off, increases the B-address by one, and returns to receive another character. If it recognizes an EOM signal (the indicator is on), the program branches to test the receive error indicator. If there is an error, the program branches to an error subroutine. If there is no error, the message is unloaded, and the program returns to the initializing step.

IBM 1009 Instructions

Several 1440 instructions are expanded to provide program control for operations that involve the IBM 1009 Data Transmission Unit.

Instructions applying to the 1009 cannot be successfully chained.

Start Transmission

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%D1	E

Function. This instruction initiates a start-of-message signal if the 1009 is in a send-run condition (transmit-receive switch is set to TRANSMIT). If the 1009 is in a receive-run condition (transmit-receive switch set to RECEIVE) the instruction causes an alarm to sound, signaling that operation intervention is necessary.

The A-address specifies the 1009 and the d-character specifies the start transmission operation.

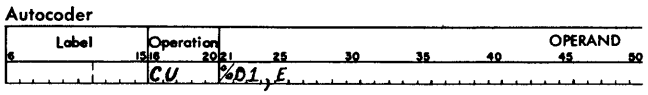
Word Marks. Word marks are not affected.

Timing. T = .0666 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	Ebb

Example. Signal the 1009 to initiate a start-of-message signal (Figure G-4).



Assembled Instruction: U %D1 E

Figure G-4. Start Transmission

Set Ready to Receive

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%D1	D

Function. The receiving 1009 signals the transmitting station that it is ready to receive and indicates the status of the previous message (see *Branch if Indicator On* instruction).

Word Marks. Word marks are not affected.

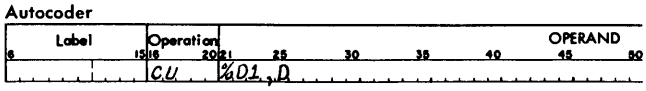
Timing. T = .0666 ms.

Note. The transmit-receive switch on the receiving 1009 should be set to RECEIVE. If it is set to TRANSMIT, the alarm is sounded to signal the operator.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	Dbb

Example. Signal the transmitting station and indicate the condition of the message received (Figure G-5).



Assembled Instruction: U %D1 D

Figure G-5. Set Ready to Receive

Move Character to the Transmitting 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%D1	BBB	W

Function. The transmitting 1440 sends the single character at the B-address to 1009. The d-character, W, specifies a transmit operation.

Word Marks. Word marks are not affected.

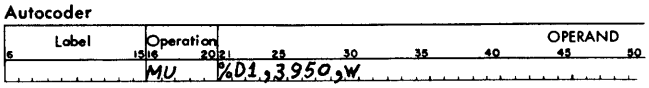
Timing. T = .1110 ms.

Note. If a group mark with a word mark is sensed in 1440 storage, an end-of-message transmit condition is recognized.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B + 1

Example. Move the character at location 3950 to the 1009 (Figure G-6).



Assembled Instruction: M %D1 3950 W

Figure G-6. Move Character to the Transmitting 1009

Move Character from the Receiving 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%D1	BBB	R

Function. This instruction transfers the single character in the receiving 1009 to the receiving 1440 core-storage location specified by the B-address. The d-character specifies a receive operation.

Word Marks. Word marks are not affected.

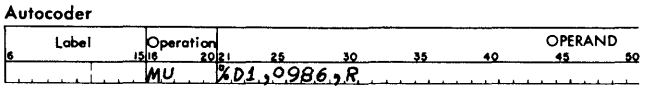
Timing. T = .1110 ms.

Note. When the 1009 recognizes an end-of-message condition, the receiving 1440 gets an end-of-message receive signal and inserts a group mark in the core-storage location specified by the next M %D1 BBB R instruction.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B+1

Example. Read a character from the 1009 and place it in core-storage position 0986 (Figure G-7).



Assembled Instruction: M %D1 986 R

Figure G-7. Move Character from the Receiving 1009

Load Character to the Transmitting 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%D1	BBB	W

Function. The transmitting 1440 sends the character at the B-address to the 1009. The d-character, W, specifies a transmit operation.

Word Marks. If a word mark is associated with the character, the 1440 converts the word mark to a word-separator character (A841). In two separate transmission cycles, the 1440 sends the word separator, then the character to the 1009. The re-cycle for the character associated with the word separator is automatic.

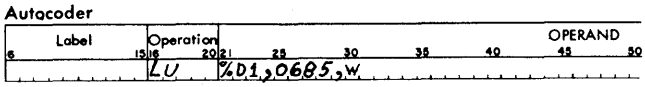
Timing. T = .1110 ms.

Note. A group mark in core storage signals an end-of-message transmit condition.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B+1

Example. Send the character and word mark at location 0685 to the 1009 (Figure G-8).



Assembled Instruction: L %D1 685 W

Figure G-8. Load Character to the Transmitting 1009

Load Character from the Receiving 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%D1	BBB	R

Function. This instruction transfers the single character in the 1009 to the receiving 1440 storage location specified by the B-address. The d-character, R, signals a receive operation.

Word Marks. If a word mark is associated with the character, it is transmitted and inserted in core storage with the character. Two transmission cycles are required to transfer the character and the word mark. The re-cycle for the word-marked character is automatic. The 1440 converts the word-separator character to a work mark.

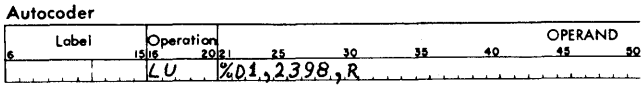
Timing. T = .1110 ms.

Note. When the 1009 recognizes an end-of-message condition, the receiving 1440 interprets an end-of-message receive signal and inserts a group mark in the core-storage location specified by the next L %D1 BBB R instruction.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B+1

Example. Read a character with word mark from the 1009 and place it in core-storage position 2398 (Figure G-9).



Assembled Instruction: L %D1 L98 R

Figure G-9. Load Character from the Receiving 1009

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. This instruction tests the indicator specified by the d-character. If the indicator is on, the program branches to the I-address for the next instruction. If it is off, the program continues with the next instruction in sequence.

d-character	*Indicator*
1	1009 RUN
2	END-OF-MESSAGE TRANSMIT
3	GOOD TRANSMISSION
4	TRANSMISSION ERROR
5	END-OF-MESSAGE RECEIVE
6	RECEIVE ERROR

Indicators

1009 Run. Turns on when the 1009 is in RUN condition. If the 1009 is not in a RUN condition, the program should stop, or loop until the RUN condition is established.

End-of-Message Transmit. Turns on in the transmitting 1440 when the 1401 or 1460 senses a group mark with a word mark at the B-address during the execution of an L %D1 xxx W, or M %D1 xxx W instruction. The indicator is turned off by the next U%D1 E instruction. The 1009 is busy during the next 250 milliseconds.

Good Transmission. Turns on if the signal sent to the transmitting station by the U %D1 D instruction specified that a good transmission occurred. The transmitting 1440 should test this indicator and branch to the routine for the next message if it is on. If the indicator is not on, the program should advance to test the transmission-error indicator.

Transmission Error. Turns on if the signal sent to the transmitting station by the U %D1 D instruction specified that a transmission error occurred. The transmitting 1440 should test this indicator and branch to an error subroutine if an error occurred. If there was no indication of error, the program should loop to retest the good transmission indicator. The logic behind this technique is that it is possible that a good-transmission condition exists, but that the signal has not been received by the transmitting 1440 before the first good-transmission test is given. This loop is repeated until one of the indicators is turned on.

End of Message Receive. Turns on when the end-of-message signal is recognized by the receiving 1009.

Receive Error. Turns on if an error was detected during the transmission from the transmitting 1009 to the receiving 1440. The receiving 1440 tests the indicator and branches to an error routine if it is on.

Word Marks. Word marks are not affected.

Timing.

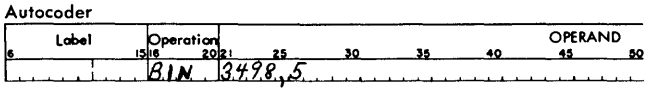
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to location 3498 if the end-of-message receive indicator is on (Figure G-10).



Assembled Instruction: B D98 5

Figure G-10. Branch If End-of-Message Receive Indicator On

Suppress 3-Second Alarm

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	A

Function. This instruction prevents the 3-second alarm from sounding during a delay (such as tape rewind). Normal alarm functions will be restored when any subsequent instruction addresses the 1009. This instruction can be given when a delay in processing can be foreseen.

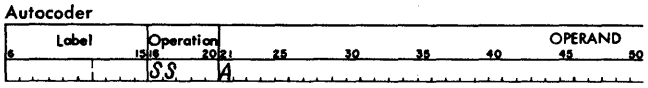
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Abb	Abb

Example. Suppress the 3-second alarm during process delay (Figure G-11).



Assembled Instruction: K A

Figure G-11. Suppress 3-Second Alarm

Suppress 3-Second Alarm and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	A

Function. This instruction prevents the 3-second alarm from sounding during a delay such as, tape rewind. Normal alarm function is restored when any subsequent instruction addresses the 1009. This instruction can be given when the delay in processing can be foreseen. The address of the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

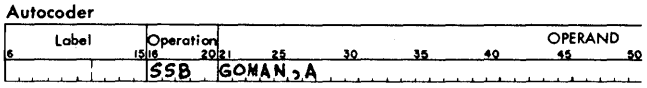
Timing.

Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Suppress the 3-second alarm during processing, delay, and branch to GOMAN (0926); see Figure G-12.



Assembled Instruction: K 926 A

Figure G-12. Suppress 3-Second Alarm and Branch

IBM 1412 Magnetic Character Reader, Model 1

The IBM 1412 Magnetic Character Reader (Figure G-13) can be used as a second unit to sort documents (off-line), or it can be attached to an IBM 1440 Data Processing System with sorting controlled either by the 1412 or by the stored program (on-line). This flexibility permits the most efficient use of both the data processing system and the 1412. Refer to *IBM 1412 Magnetic Character Reader*, Form A24-1421 and A24-3004.

Data Flow

The IBM 1412 reads documents into the processing unit for processing when the reader is instructed by the stored program to feed a document. Control over the selection of data for transmission to the processing unit remains with the 1412 read-field keys.

Sorting functions are identical in the reader on-line mode and reader off-line mode because sorting, in both instances, remains under the control of the 1412.

The stored program must instruct the 1412 to read each document. After the data is read into the processor, the stored program can determine the pocket to which each document is directed by examining the sort-column digit. In the case of error documents, the read-check indicator and the appropriate field-error indicators can be interrogated to determine the type of error.

IBM 1412 Instructions

One IBM 1412 can be connected with the IBM 1440 Data Processing System to read magnetic-ink characters. Instructions that control 1412 operations are presented in this section.

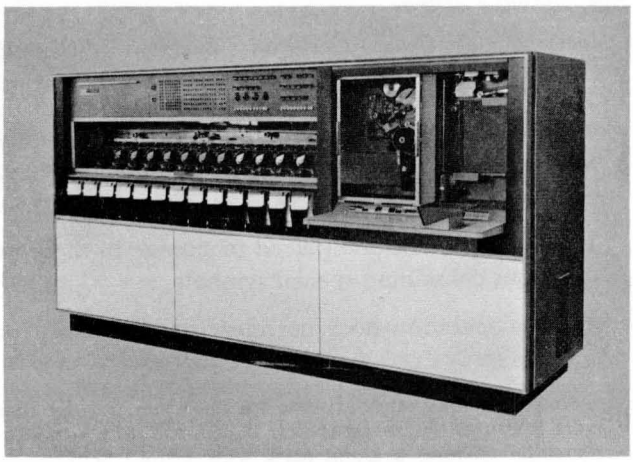


Figure G-13. IBM 1412 Magnetic Character Reader

Instructions applying to the 1412 cannot be successfully chained.

Engage MICR (Magnetic Ink Character Recognition) Reader

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%S1	E

Function. This instruction selects the 1412 and starts document feeding.

This instruction must precede LOAD FROM MAGNETIC CHARACTER READER and MOVE FROM MAGNETIC CHARACTER READER instructions. Once selected, the 1412 begins feeding documents. Document feeding continues until the 1412 is instructed to stop feeding documents, or until a system stop occurs.

Word Marks. Word marks are not affected.

Timing. T = .0666 ms + I/O.

Refer to tming chart (Figure G-30).

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%21	Ebb

Example. Engage the 1412 and start feeding documents (Figure G-14).

Autocoder

Label	Operation	OPERAND
	CU	%S1,1,E

Assembled Instruction: U %S1 E

Figure G-14. Engage Magnetic Character Reader

Disengage MICR Reader

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%S1	D

Function. This instruction disengages the 1412 and signals it to stop feeding documents.

Word Marks. Word marks are not affected.

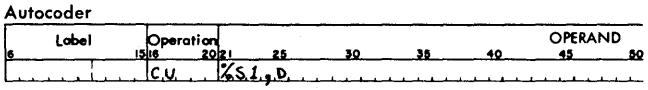
Timing. T = .0111 (L_I + 1) ms + I/O.

Refer to the timing chart (Figure G-30).

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%21	Dbb

Example. Disengage the 1412 and stop feeding (Figure G-15).



Assembled Instruction: U %S1 D

Figure G-15. Disengage Magnetic Character Reader

Load from MICR Reader

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%S1	BBB	R

Function. This operation causes a serial transfer of characters from the 1412 to the processing unit. The 1412 must have previously been signaled to begin feeding documents before this instruction can be executed. The data processing system cannot perform any other operation during the execution of this instruction.

The first character (including A. B. A. special symbols) transferred from the reader to the processing unit is placed in the storage location specified by the B-address. Subsequent characters transmitted from the same document enter successively lower storage locations.

When the load instruction is used, a word mark is automatically placed in each storage position containing an A. B. A. special symbol.

Note: The read operation is completed when the processing unit receives an end-of-data signal.

The earliest-occurring of the following conditions generates an end-of-data signal:

1. The high-order field-definition symbol for the last field selected for reading passes the read head (photocell 3).
2. The leading edge of the document reaches photocell 3A during processing unit on-line operations. When this condition generates an end-of-data signal, 7.5 ms are available for stacker selection.
3. The trailing edge of the document reaches photocell 3.
4. The processing unit encounters a group mark with a word mark in read-in storage.

A read instruction is required for each document. If the leading edge of the document reaches photocell 3 without a read instruction for that document, the late-read indicator turns on. The indicator turns off when the leading edge of the next document reaches photocell 2 (read-ready time for the next document). When the stored program is controlling document distribution, the document must be directed to a pocket with a SELECT STACKER instruction (Kd). When the 1412 controls distribution, the late-read document automatically enters the reject pocket.

Word Marks. The word marks are read into storage as a result of sensing A. B. A. special symbols. Therefore each position in storage containing A. B. A. symbols also contains a word mark.

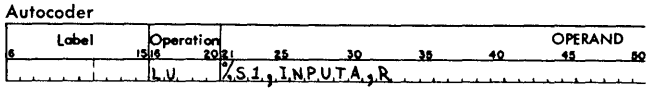
Timing. T = .0999 ms + message length + document movement* + 1.

*Refer to Figure G-30.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%21	Groupmark + 1 (see text)

Example. Transfer characters serially from the 1412 to area labeled INPUTA (0705); Figure G-16.



Assembled Instruction: L %S1 705 R

Figure G-16. Load from Magnetic Character Reader

Move from MICR Reader

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%S1	BBB	R

Function. This operation is the same as the LOAD FROM MAGNETIC CHARACTER READER operation, with two exceptions:

1. No word marks are placed in storage in those positions containing special symbols.
2. This operation does not clear word marks from storage.

Note: If a word mark exists in storage where a group mark mark is placed by an end-of-data signal, the storage position thereafter contains a group mark with a word mark.

Word Marks. Word marks are not affected.

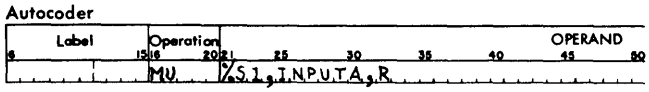
Timing. T = .0999 ms + message length + document movement* + 1.

*Refer to Figure G-30

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%21	Groupmark + 1 (see text)

Example. Transfer characters serially from the 1412 to area labeled INPUTA (0705); Figure G-17.



Assembled Instruction: M %S1 705 R

Figure G-17. Move from Magnetic Character Reader

Select Stacker—MICR Reader

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	d (See Figure G-18)

Function. This instruction directs the document that was just read by the reader to the reader pocket specified by the d-character.

The reader must be operating in the processing unit on-line mode to execute this instruction. If this instruction is given when the reader is operating in the reader on-line mode, the 1412 feeding stops and the processing unit light turns on.

The d-character (A-M) representing one of the 13 reader pockets is sent to the 1412 and activates the chute blade that directs the document to the selected pocket.

The SELECT STACKER instruction must be given before the document reaches photocell 4. Processing time available to determine pocket selection depends on the position of the leading edge of the document when all required reading has been completed. For example, a minimum of 13 ms is available for pocket selection if the transit-routing field is the last read field selected.

d-character	Reader Pocket
A	A
B	B
C through L	0 through 9
M	Reject

Figure G-18. Select Stacker d-Characters

If the trailing A. B. A. field-definition symbol is missing from the last read field selected, an end-of-data signal is provided by either photocell 3 or 3A to guarantee at least 7.5 ms for a SELECT STACKER decision.

Photocell 3A is located in the transport mechanism in such a position that the end-of-data signal occurs after all possible inscribing positions on the document have passed under the read head. If a SELECT STACKER instruction is not given in time, the 1412 stops feeding documents (sort-compare) and directs all documents that have left the separator station to the reject pocket.

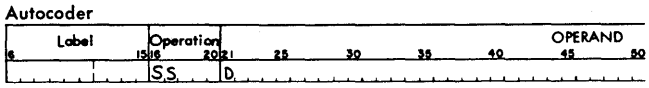
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

Example. Selects the document that was just read into pocket number 1 (Figure G-19).



Assembled Instruction: K D

Figure G-19. Select Document into Stacker 1

Select Stacker and Branch — MICR Reader

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	d

Function. The SELECT STACKER AND BRANCH instruction functions exactly like the SELECT STACKER instruction, except that the address of the next instruction to be executed is specified by the I-address.

Word Marks. Word marks are not affected.

Timing.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Branch if MICR Reader Late-Read Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	1

Function. This operation tests the 1412 to determine whether or not it has detected a late-read condition (document reaching photocell 3 without a READ instruction for that document). This indicator turns off when the next document creates a read-ready condition.

Word Marks. Word marks are not affected.

Timing.

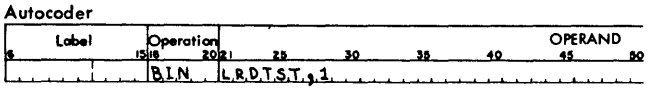
No branch, or branch without indexing. T = .0666 ms.

Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	1bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the late-read indicator. If the indicator is ON, branch to LRDTST (0623); see Figure G-20.



Assembled Instruction: B 623 1

Figure G-20. Branch If Magnetic Character Reader Late-Read Indicator On

Branch if MICR Reader Read-Not-Ready Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	2

Function. This instruction tests the read-not-ready indicator and branches to the instruction specified by the I-address when the indicator is ON. The leading edge of a document passing photocell 2 turns off the indicator when the trailing edge of the previous document has passed under photocell 3. This occurs for the 15 ms prior to the reading of the document. This indicator turns on when the leading edge of the document passes photocell 3. A read instruction should be given only when the read-not-ready indicator is off.

Word Marks. Word marks are not affected.

Timing.

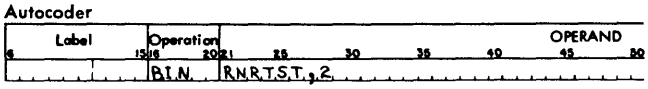
No branch, or branch without indexing: T = .0666 ms.

Branch (with indexing): T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	2bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the read-not-ready indicator. If the indicator is ON, branch to RNRTST (0735); Figure G-21.



Assembled Instruction: B 735 2

Figure G-21. Branch If Magnetic Character Read-Not-Ready Indicator On

Branch if MICR Reader Read-Check Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	3

Function. This operation tests the read-check indicator and branches to the instruction specified by the I-address when the indicator is on. The indicator turns on during a read operation if any of the selected fields have:

1. an unreadable digit or symbol
2. a wrong sequence of symbols
3. missing digits or symbols
4. a missing field
5. an account number self-check-digit verification error.

This indicator turns off when the leading edge of the next document passes photocell 3.

Word Marks. Word marks are not affected.

Timing.

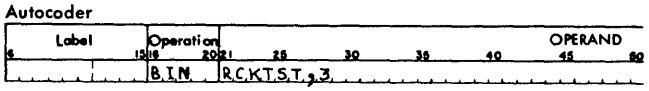
No branch, or branch without indexing: T = .0666 ms.

Branch (with indexing): T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	3bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the read-check indicator. If the indicator is on, branch to RCKTST (0432); Figure G-22.



Assembled Instruction: B 432 3

Figure G-22. Branch If Magnetic Character Reader Read-Check Indicator On

Branch if MICR Reader Amount-Field Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	4

Function. This operation tests the amount-field indicator and branches to the instruction specified by the I-address when the indicator is ON. The indicator turns on during a 1412 read operation if:

1. Any of the characters in the amount field (including the amount special symbols) are unreadable.
2. Special symbols are missing or out of sequence.
3. The field is missing.
4. The field length is invalid.
5. The late-read indicator is on. The amount-field indicator turns off when the leading edge of the next document passes photocell 3.

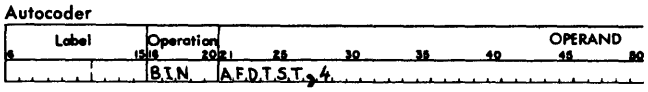
Word Marks. Word marks are not affected.

Timing.
No branch, or branch without indexing: T = .0666 ms.
Branch with indexing: T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	4bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the amount-field indicator. If the indicator is ON, branch to AFDTST (0688); see Figure G-23.



Assembled Instruction: B 688 4

Figure G-23. Branch If Magnetic Character Reader Amount-Field Indicator On

Branch if MICR Reader Process-Control Field Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	5

Function. This operation tests the process-control field indicator and branches to the instruction specified by the I-address when the indicator is on. The indicator turns ON during a 1412 read operation if:

1. Any of the characters in the process-control field (including the special symbols) are unreadable.
2. The field is missing.
3. Special symbols are out of sequence or are missing.
4. The field length is invalid.
5. The late-read indicator is on. The process-control field indicator turns off when the leading edge of the next document passes photocell 3.

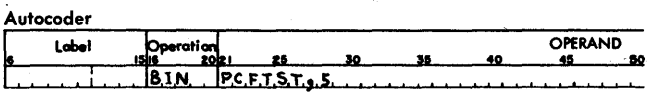
Word Marks. Word marks are not affected.

Timing.
No branch, or branch without indexing: T = .0666 ms.
Branch with indexing: T = .0777 ms.

Address Registers After Indexing.

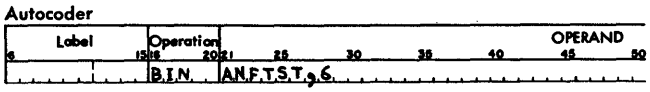
	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	5bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the process-control field indicator. If the indicator is ON, branch to PCFTST (0892); see Figure G-24.



Assembled Instruction: B 892 5

Figure G-24. Branch If Magnetic Character Process-Control Field Indicator On



Assembled Instruction: B 392 6

Figure G-25. Branch If Magnetic Character Account-Number Field Indicator On

Branch if MICR Reader Account-Number Field Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	6

Function. This operation tests the account-number field indicator and branches to the instruction specified by the I-address when the indicator is on. The indicator turns on during a 1412 read operation if:

1. Any of the characters in the account-field (including the special symbols) are unreadable.
2. The field is missing.
3. Special symbols are missing or out of proper sequence.
4. The field length is invalid.
5. The late-read indicator is on. If the 1412 is equipped with the self-checking number special feature, self-checking digit errors also turn on the account-number field indicator. The account-number field indicator turns off when the leading edge of the next document passes the photocell 3.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	6bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the account-number field indicator. If the indicator is on, branch to ANFTST (0392); see Figure G-25.

Branch if MICR Reader Transit-Routing Field Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	7

Function. This operation tests the transit-routing field indicator and branches to the instruction specified by the I-address when the indicator is on. The indicator turns on during a 1412 operation if:

1. Any of the characters in the transit-routing field (including the special symbols) are unreadable.
2. The field is missing.
3. Special symbols (except the dash) are missing or out of sequence.
4. The field length is invalid.
5. The late-read indicator is on. The transit-routing field indicator turns off when the leading edge of the next document passes photocell 3.

Word Marks. Word marks are not affected.

Timing.

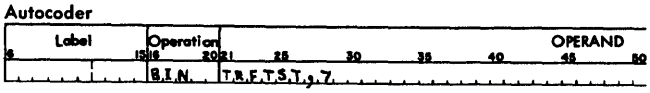
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	7bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the transit-routing field indicator. If the indicator is ON, branch to TRFTST (0543); see Figure G-26.



Assembled Instruction: B 543 7

Figure G-26. Branch If Magnetic Character Reader Transit-Routing Field Indicator On

Branch if MICR Reader Document-Spacing Check Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	8

Function. This instruction tests the document-spacing check indicator and branches to the instruction specified by the I-address when the indicator is on. The testing must be made only after the trailing edge of the document passes photocell 3. This is nearly equivalent to read-ready time.

The document-spacing check indicator turns on when there is less than a minimum space between documents, or when a document is longer than a specified maximum.

This indicator turns off when the leading edge of the next document passes photocell 3.

Word Marks. Word marks are not affected.

Timing.

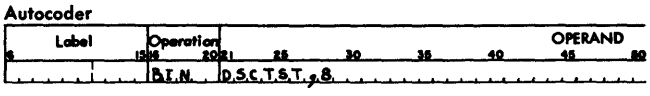
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Indexing.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	8bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the check indicator, and branch to location labeled DSCTST (0650); see Figure G-27.



Assembled Instruction: B 650 8

Figure G-27. Branch If Magnetic Character Reader Document Spacing Check Indicator On

Programming Considerations on the IBM 1412

The conditions described in this section are items that will assist the programmer in developing the programs.

Document Spacing Error

The document-spacing indicator turns on if:

1. The space between documents at the read station is less than 3 inches.
2. The distance between leading edges of adjacent documents at photocells 3 and 4 is less than 9 inches.
3. The document is more than 9 inches long.

If the indicator is on at read-ready time, the document just read is directed to the reject pocket, regardless of SELECT STACKER instructions.

Engage Line

After the engage line is turned on by an ENGAGE 1412 instruction, it is turned off only:

1. by a DISENGAGE 1412 instruction.
2. by pressing START/RESET on the console panel.
3. if the check-reset light, located on the console panel, is on.

These machine errors require a restart procedure. This restart procedure must always include an ENGAGE 1412 instruction.

Serial Number (Auxiliary On-Us) Field

Consider these facts when processing data from the serial number field:

1. When operating in the processing unit on-line mode, inscribing in the serial number field must *not* extend beyond 7¼ inches from the leading edge if the serial number is to be read.
2. The reader on-line mode allows reading beyond the first 7¼ inches of a document.
3. There is no field-length count on the serial number field.

Word Marks

A word mark accompanies each special symbol read into the processing unit when the LOAD instruction is used. All special symbols (except the dash symbol) associated with active read fields read into the system.

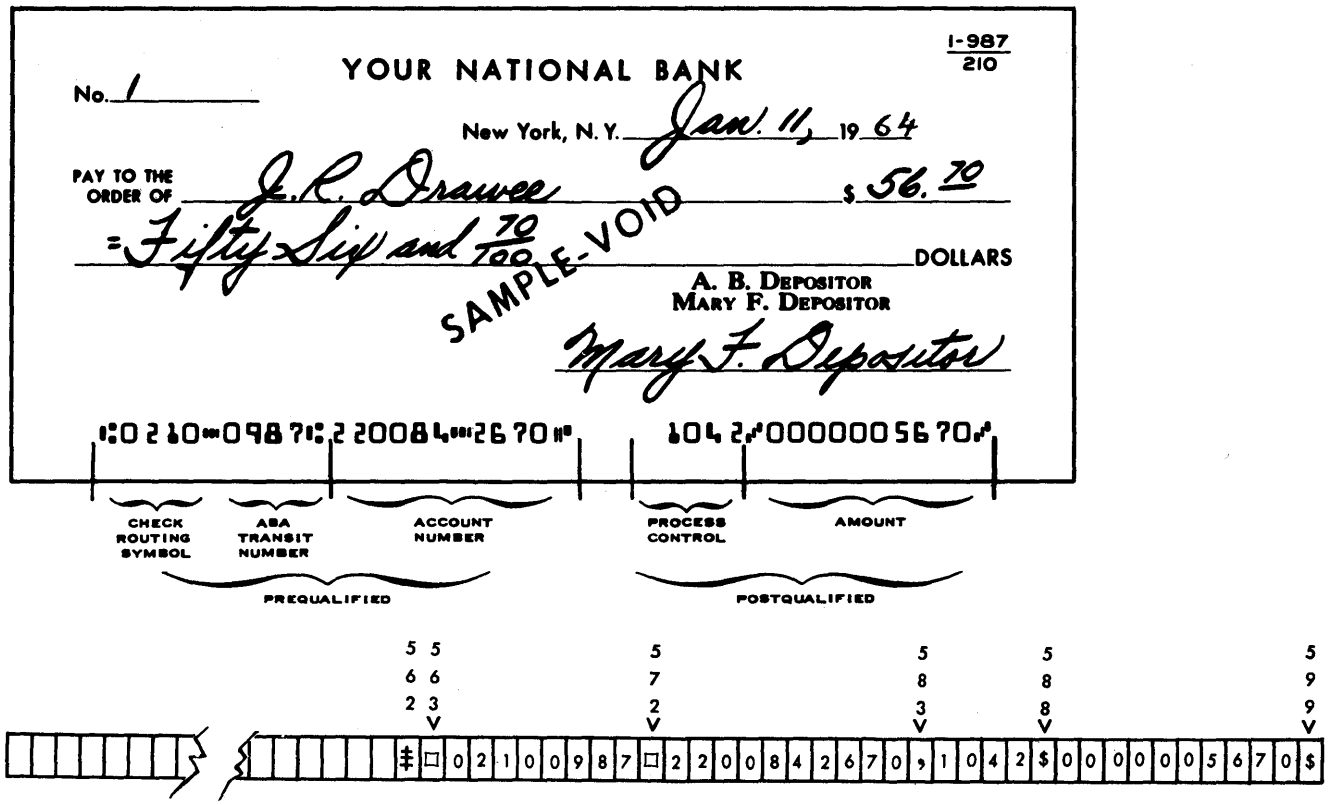


Figure G-28. Inscribed 6-Inch Check Read into Storage Correctly

Group Marks

A group mark only enters storage in the processing unit at the end of each READ FROM 1412 operation.

Clearing Storage after Read Errors

Clear the read-in area before each read-from-1412 operation. This ensures that only data from the document currently being read is in storage at the end of any read operation.

Data in Storage

A field cannot be read if the proper leading special symbol of that field is not recognized. The field is terminated by recognition of the proper trailing special symbol or any other field definition special symbol. (Dashes are not interpreted as field definition special symbols.) The following examples are used to illustrate the preceding statements. Note that a field is not valid unless it has the proper leading symbol and the proper trailing symbol. For the examples, a document containing all fields except the serial number field has been chosen. All inscribed fields on the document have been selected.

Data Stored without Read Errors

Figure G-28 illustrates data correctly inscribed on the original document, and illustrates its appearance in storage.

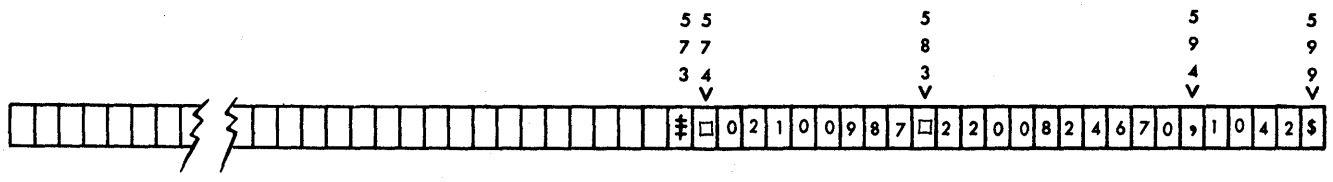


Figure G-29. Data Read into Storage, First Amount Symbol Not Identified

Data Stored with Read Error Conditions

1. If the low-order amount field symbol (1 S S 1) is not identified, nothing enters storage until a special symbol is identified. The dollar amount field never enters storage. The amount and process control fields are in error. All fields except the amount field enter storage (Figure G-29). Note that the dash does not enter storage.
2. If the second amount symbol (2 S S 1) is not identified, the dollar amount field and process control field are in error. They appear in storage with the second amount symbol represented as an asterisk. All other fields are in storage.
3. If the first on-us symbol (1 S S 2) is not identified, the process control and account number fields are in error. They appear in storage with the first on-us symbol represented as an asterisk. All other fields are in storage.
4. If the first transit symbol (1 S S 3) is not identified, the account number and transit fields are in error. They appear in storage with the first transit symbol represented as an asterisk. All other fields are in storage.
5. If the second transit symbol (2 S S 3) is not identified, the transit field is in error. The second transit symbol appears as an asterisk. All other fields are in storage. When operating in the processing unit on-line mode, the end of data occurs, for documents less than 7¼ inches long, when the trailing edge of the document passes photocell 3, and, for documents 7¼ inches or longer, when the leading edge of the document passes photocell 3A. When operating in the reader on-line mode, the end of data occurs when the trailing edge of the document passes photocell 3.
6. If a dash in the transit field is not identified, the transit field is in error. An asterisk appears as an extra character position in storage.
7. A missing dash (S S 4), a missing blank space, extra dashes, or extra blank spaces cause no read-in errors. Blank spaces and missing dashes are not transmitted to storage.
8. A missing character in any field causes that field to be in error. It forces all data that is to the left of the missing character on the check to shift one position up in storage.
9. An unreadable character (including the dash symbol) causes that field to be in error and an asterisk to be inserted in that position.
10. A missing field symbol is the same as an unreadable symbol except that an asterisk is not placed in storage.
11. If the transit routing field is not selected and the first transit symbol (1 S S 3) is not identified, the 1412 continues to load the transit field and stops after sensing the second transit symbol. The first transit symbol enters storage as an asterisk. The account-number field is in error. The account-number field and the transit field enter storage. For this example, assume that the account-number field has been selected.
12. If the amount and transit fields are selected and the second amount symbol is not sensed, the 1412 continues to read data into storage until it senses a field-definition special symbol or an end-of-data condition.
13. If the amount and transit fields are selected and the first transit symbol is not sensed, only the amount field and the second transit symbol enter storage. The 1412 considers the second transit symbol to be the first transit symbol (1 S S 3) and continues to read data until it senses an end-of-data condition. If the serial number is on the check, sensing an *on-us* symbol causes an end-of-data signal.
14. A second *on-us* symbol (S S S 2) before the first transit symbol (1 S S 3) is acceptable. The S S S 2 acts as a closing symbol for the account number field and takes an extra position in storage. If the S S S 2 is unreadable, the account number field is in error and an asterisk enters storage, even though the first transit symbol is readable.
15. If the S S S 2 is missing but the first S S 3 is present and readable, the account number field enters storage as a valid field.
16. Any special symbol conflict causes all field error latches to turn on.

Stop Conditions

The stored program must be written to handle stops initiated in the data processing system.

Whenever possible, initiate system stops by pressing the 1412 stop-restore key. When this is done, source documents stop feeding in the 1412. All data read is properly entered into the system and correctly processed. The system can then be stopped with a mini-

mum of problems, and the program will have stopped in a read-not-ready condition. All additional processing will have been accomplished.

Whenever any other stop key in the system is pressed, the stored program executes the instruction being handled. Then, the processing unit stops operating.

All documents in the 1412 whose leading edges are past photocell 1 when the processing unit stops continue to feed through the reader. Those whose leading edges are past photocell 4 have been read into the system, and have been correctly distributed. Those whose leading edges have not reached photocell 4, but whose active read-fields have passed the read head when the system stops, have entered data into the system correctly, but have not been properly distributed. Those that have started under the read head but whose active data fields have not completely passed under the read head, have entered part of their data into the system, and have not been assigned distribution. Those whose leading edges have not yet reached photocell 3 have been neither read nor assigned distribution.

A system-error stop acts upon the 1412 in the same manner.

The stored program, therefore, must be written to print out (or otherwise indicate to the operator) the last data entered into storage. With this information the operator can determine the last document read and verify the accuracy of document distribution. This print-out indication is also needed in case of 1412 jams and 1412 emergency stops.

IBM 1412 Timings

Determining the 1412 Feeding Rate

To develop a formula for the 1412 feeding rate:

1. Documents move through the 1412 at a rate of 5 ms per inch.
2. The average space between documents approximates the average length of the documents.
3. One millisecond is equal to 1/1000 of one second.

To find the average 1412 document cycle time:

1. Determine the average length of the documents.
2. Multiply this figure by two. (This accounts for the space between documents.)
3. Multiply this result by 5 (ms per inch). This product is the average number of milliseconds required for each document.

Example: If the average document length is 8 inches, then 8 in/doc. $\times$ 2 $\times$ 5 ms/in. = 80 ms per document. Because 2 and 5 are constants, they can be combined to arrive at the formula: A $\times$ 10 = ms/document, where A is the average length of the document.

However, a figure in documents-per-minute will be more practical. To determine the number of milliseconds per minute, multiply 1000 (ms per sec.) $\times$ 60 (seconds per minute). 1,000 $\times$ 60 = 60,000 milliseconds in a minute.

To determine the number of documents per minute, divide the number of milliseconds per minute by the document cycle time (in milliseconds). Therefore, the formula is:

60,000 $\div$ (A $\times$ 10), or

6,000 $\div$ A

Example: Using the example of 8 inches for an average document:

6000 $\div$ 8 = 750 documents per minute.

Feed Call

Figure G-30 illustrates, schematically, the path of a document through the 1412. The positions of five photoelectric cells, which detect the presence or absence of a document, are shown.

Pressing the 1412 start key feeds documents to photocell 1. To move documents past photocell 1 when operating on-line, the stored program must initiate a *1412 feed call* by supplying an ENGAGE 1412 instruction. During normal operations, the feed call is available when documents reach photocell 1. This allows a continuous flow of documents from the separator station. However, if a feed call is not available when the leading edge of a document is sensed at photocell 1, the document stops. If a feed call is interrupted at any time after its leading edge is sensed at photocell 1, that document feeds on to a pocket.

Conditions that Interrupt the Feed Call. Conditions that interrupt a feed call and stop document feeding with a document under photocell 1 are:

1. A DISENGAGE 1412 instruction has been issued.
2. A pocket in the 1412 is full.
3. The unload-pocket/restart key has been pressed.

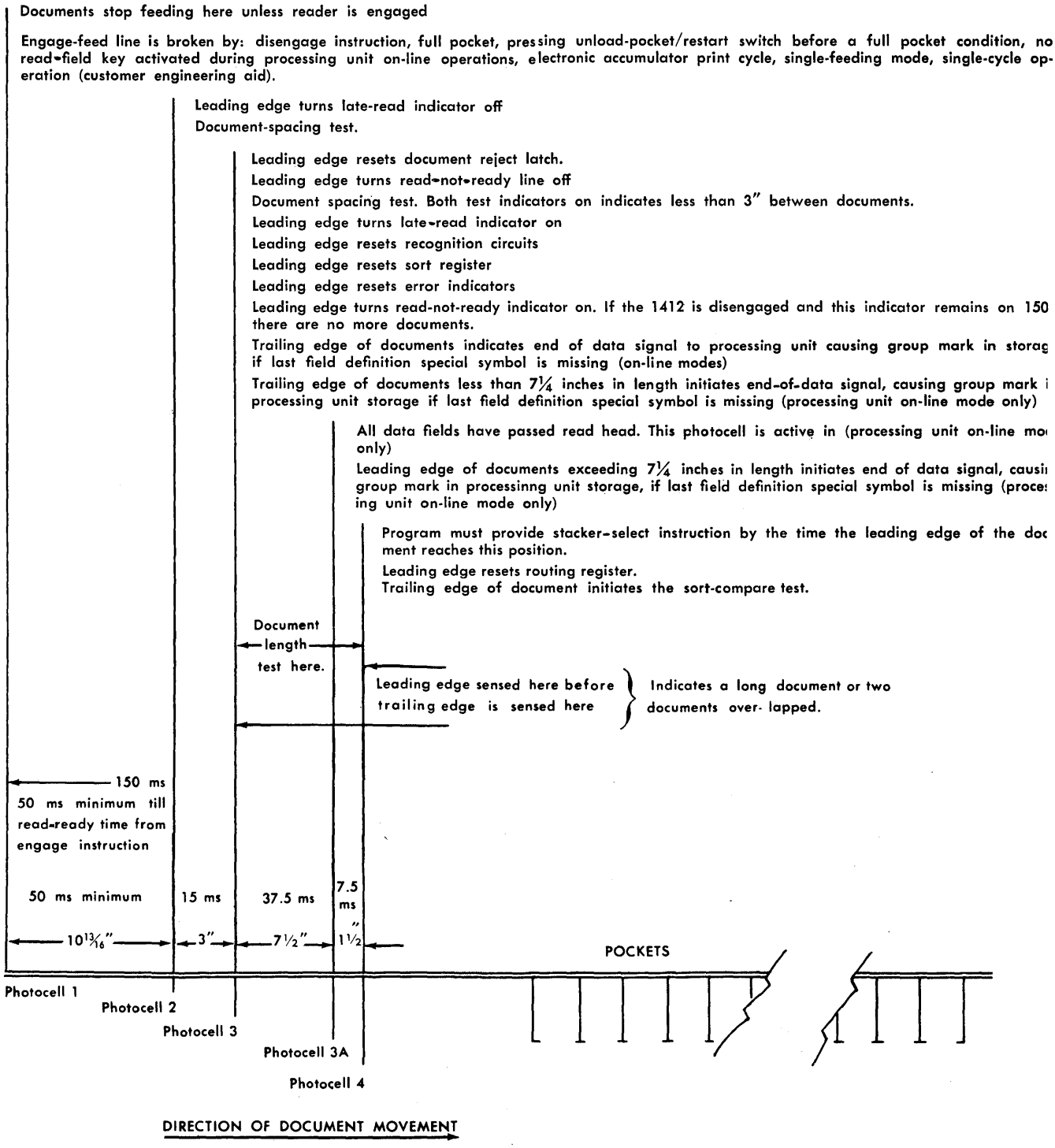


Figure G-30. IBM 1412 Timing Chart

4. No read-field key was pressed for a processing unit-on-line operation.

5. The electronic accumulator – sequence-checking feature is executing a print cycle.

6. The machine is single-cycling in the customer-engineering single-feed mode.

All these conditions except the disengage instruction cause the stored program to enter a programmed read-not-ready loop (no document sensed by photocell 2).

Time Between Documents

The minimum time between documents is 15 ms.

Documents in Flight

A disengage 1412 instruction interrupts the feed call. Any documents whose leading edges have passed photocell 1 must be processed by the processing unit program.

Photocell 3 is located near the read head. The distance between photocell 1 and photocell 3 is $10\frac{13}{16}$ inches. Normally, the spacing between documents is approximately equal to the length of the trailing document. The minimum spacing is 3 inches. When the minimum length check (6 inches) and the minimum spacing requirements are considered, a program would, under valid conditions, need to handle the processing of two documents after a disengage instruction. However, if less than the minimum spacing is present, or a document is short, it is possible to have as many as three documents whose leading edges have passed photocell 1, but have not yet reached the read head. Because every one of these documents must be read and distributed (if only to determine invalid spacing), write the program to halt if *more* than three documents are read after a disengage instruction.

When a read-not-ready condition exists for 115 ms after a DISENGAGE 1412 instruction has been executed, the stored program can proceed as if there are no more documents to be processed.

Select Stacker Timings

Processing time for stacker selection begins with the end-of-data signal. The SELECT STACKER instruction must be available to the 1412 when the document reaches photocell 4. The time available for selection can be determined by measuring the distance between the leading edge of the document and photocell 4 when the stored program receives an end-of-data signal.

A formula for select-stacker time is: $(9 - R) \times 5$.

9 = inches between read head and photocell 4.
R = distance between the leading edge of the document and the read head when the end-of-data signal is generated.
5 = number of milliseconds per inch of document travel.

When no read-error occurs, these timings apply:

1. *Amount field only is selected.* The available time is 35 ms for stacker selection and 35 ms for processing.

2. *Process control field is last field selected.* Timing depends upon the maximum number of digits to be read. Use the formula to determine select-stacker and processing time.

3. *Account number field is last field selected.* Read fields can use $4\frac{15}{16}$ inches of document space. Available time is 20 ms for stacker selection and 20 ms for processing.

4. *Transit routing field is the last field selected.* The transit/routing field can extend as far as $6\frac{3}{16}$ inches from the leading edge. This condition allows 13 ms for stacker selection and 15 ms for processing.

5. *Serial number field is selected.* If the serial number field is selected, only 7½ ms are available for stacker selection. Because this field may not be inscribed on some documents, the end-of-data signal must be generated by the leading edge of the document passing photocell 3A. Processing time is 15 ms.

Note: More than 7½ ms may be available for stacker selection when:

1. the document is completely read before the leading edge of the document reaches photocell 3A.
2. the document is less than 7½ inches long
3. a group mark with a word mark is reached in storage.

Because it is impossible to estimate the actual time available in such cases, 7½ ms should be used as the minimum time available for processing and program-sorting the error document.

IBM 1448 Transmission Control Unit

Data processing, without fast accurate communication, is limited by the numerous delays between the source of data and the processor. A data processing system serving as a central control for many remote locations requires the best techniques of data communication. The ideal method is a combination of communication and processing operations in an effective single system.

IBM Tele-processing systems are serving business and industry by combining computer operations and data transmission facilities into integrated data processing systems. Here are the speed, convenience, and efficiency of centralized data processing for the business, large or small, that is physically decentralized. Here is the control center for the financial institution that requires, at a central point, variable or fixed information from many locations. Refer to *IBM 1448 Transmission Control Unit*, Form A24-3010.

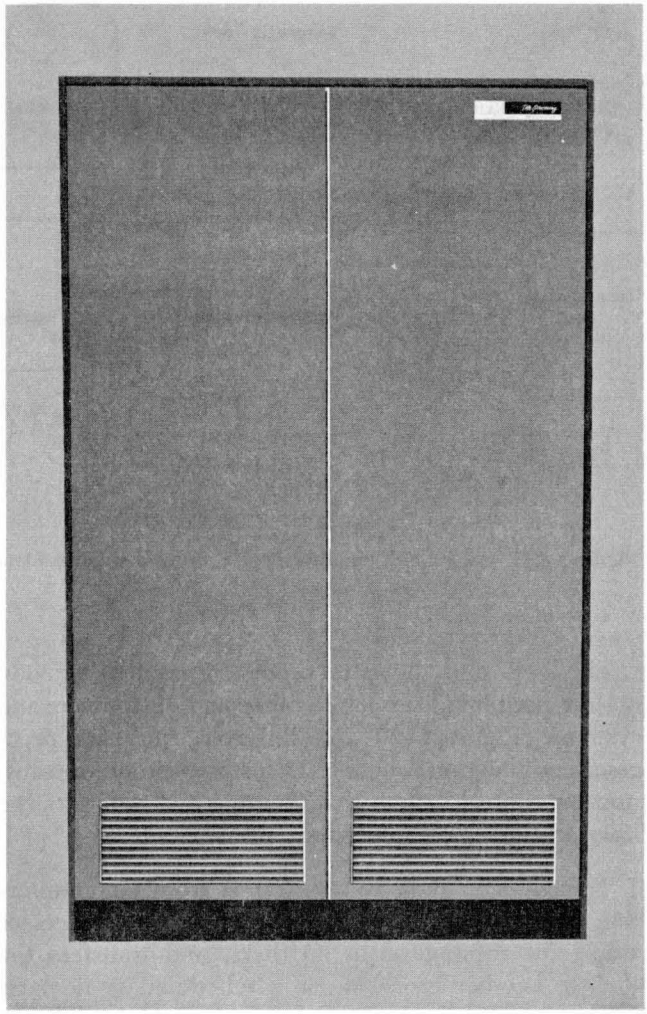


Figure G-31. IBM 1448 Transmission Control Unit

Transmission Control

Each line added to a communication network increases the possibility of delay and error. Speed and dependability of a communication system depend on the control equipment, which blends the array of transmission lines into an efficient network. Transmission control is the nucleus of any communication system made up of many lines, each with a number of terminals, leading to a central point.

The three basic purposes for transmission control in an integrated data processing system are:

1. to establish a connection between the central processor and the terminals on the communication circuit.
2. to prevent indeterminate situations on the line, such as distorted transmission or garbled or lost signals.
3. to allow for the requirements of the data processing equipment.

The IBM *1448 Transmission Control Unit* (Figure G-31) is an economical means of entering numeric, alphabetic, and special-character data directly into an IBM data processing system from as many as 40 half-duplex multipoint communication lines (Figure G-32). Information can be transmitted on half-duplex lines in either direction, but only one direction at a time. This IBM Tele-processing system component directs and regulates the flow of data and provides compatibility among terminals and processing and exchange devices.

The 1448 with its associated processor handles such applications as inquiry and file updating. The 1448 and processor combination controls transmission of information, and processes this information in-line.

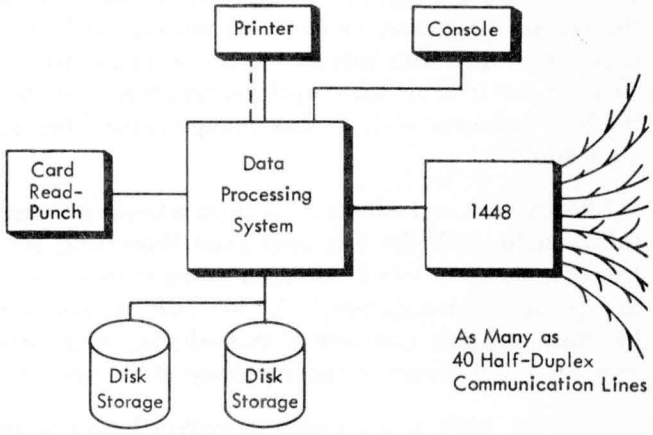


Figure G-32. The Processor Controls as Many as 40 Half-Duplex Channels

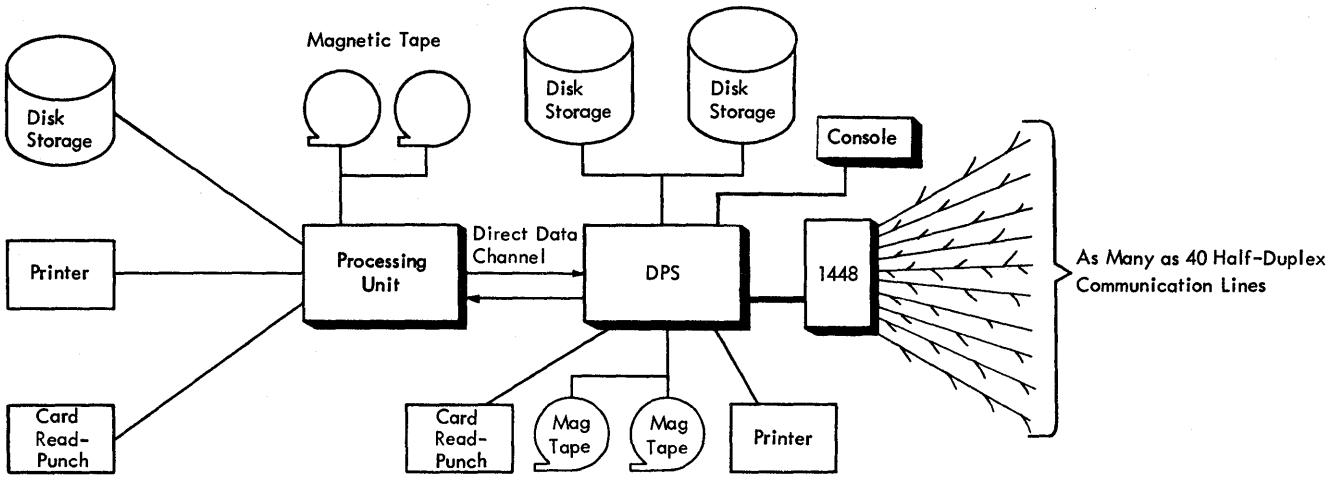


Figure G-33. IBM 1448 and Processor Serving a Central Data Processing System

The 1448 and the data processing system provide on-line peripheral service to other IBM data processing systems (Figure G-33). In this role, the 1448 processor combination functions as a stored-program transmission control system that controls and monitors the lines and assembles messages.

Transmission data is transferred from communication lines to processor core storage. The processor stores the message data on disks, and transfers the data to another processor on a scheduled or demand basis. The disks can also be transferred manually to other systems having IBM 1311 Disk Storage Drives. With the direct-data-channel feature, data is transferred directly to and from another IBM data processing system having direct-data-channel capabilities.

This expanded system covers a list of other operations: format checking of incoming messages, editing and scheduling outgoing messages from the data of the second processor, message accounting, and message switching. The second processor takes over a share of the total required systems function, and uses the first processor with its disk-storage capabilities as a backup.

Effectiveness and efficiency of large systems increase substantially with the IBM 1440 Data Processing System as a peripheral data converter (card-to-tape, tape-to-tape, and tape-to-printer). A 1448/1440 system (for transmission, data conversion, and editing) magnifies even more the economy and efficiency of the system.

With the 1448, a decentralized system becomes, in effect, centralized. The program of the processing unit control unit is an instrument of that control.

Operation

The exchange of information between the 1448 and the processor is initiated by a scan operation code associated with a priority interrupt. An interrupt is a temporary interruption of the processor's main routine by an external signal, in this case, from the 1448. The main routine continues in sequence after the interrupt routine, including the scan operation if completed.

The scan operation itself causes the automatic transfer of characters from the 1448 to the message assembly areas in the processor.

The stored program assigns the message-assembly and distribution areas, which are variable in both length and location.

IBM 1448 Instructions

The instructions described in this section are used with the IBM 1440 Data Processing System to provide for the exchange of information between the IBM 1448 and the processor.

Instructions applying to the 1448 cannot be successfully chained.

Scan

Instruction Format.

Mnemonic	*Op Code*	*B-Address*
None	O	BBB

Function. The SCAN instruction, in actual machine language, is made up of an alphabetic O as the opera-

tion code, and a 3-character address representing the high-order position of the scan control field.

The SCAN instruction is restricted to basic single-address format and must be followed by a word mark in the next location. An attempt to force a 2-address format causes an improper address in the B-address register at the beginning of instruction execution time, and the instruction affects storage locations other than the intended control field.

Word Marks. Word marks are not affected.

Timing. T = .0111 (6 + 2NI + 5M + 6P + 7R) ms.

NI = The number of idle lines or receiving lines with empty line buffers, and/or the number of transmitting lines with one or more characters in the line buffers at the beginning of the scan operation, and/or the number of polling lines that do not need a new polling address.

M = The number of receiving lines with one character in the line buffers at the beginning of the scan operation.

P = The number of receiving lines with two characters in the line buffers, and/or the number of transmitting lines with empty line buffers at the start of the scan operation.

R = The number of lines in RECEIVE-CONTROL status that are transferring 2-character polling addresses from the processor to the 1448.

Minimal execution time:
T = .0111(6 + 2L) ms.
L = The number of lines.

Maximum execution time:
T = .0111(6 + 7L) ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Storage address of the last data cycle. If no data cycles are taken, it is the address of the beginning of the control field.	End of Control Field + 3

Example. Scan 1448 lines for input or output traffic. In Figure G-34, the beginning address of the control field is labeled CTRLWD (0300).

Autocoder

Label	Operation	OPERAND
	O	CTRLWD

Assembled Instruction: O 300

Figure G-34. Scan

Interrupt Instructions

Any of these conditions in the 1448 makes the 1448 request an interrupt of the processor's main program:

1. Any buffer-full condition on the receiving lines.
2. Any buffer-empty condition on the transmitting lines.
3. Any status condition with an EOB (end-of-block) bit.
4. The 1448 requests the next polling address.

The interrupt routine contains the necessary preparatory operations and the scan operation. The actual interruption of the main program takes place when an instruction is being read, but before the execution of

DESCRIPTION	OPERATIONS			
	Interruptable		Non-Interruptable	
	Op Code	Length	Op Code	Length
Add	A	7	A	1,4
Branch	B	5,8	B	1,4
Branch if Bit Equal	W	8	W	1,4
Branch if Word Mark or Zone	V	8	V	1,4
Compare	C	7	C	1,4
Control Carriage	F	5	F	2
Control Unit			U	2,5
Clear	/	7	/	1,4
Clear Word Mark	□	7	□	1,4
Divide	%	7	%	4
Edit	E	7	E	4
Expand Compressed Tape	X	7	X	1,4
Halt	•	5,7	•	1,2,4
I/O Operations			All	All
Load (Excluding I/O)	L	7	L	1,4
Modify Address	#	7	#	1,4
Move (Excluding I/O)	M	7	M	1,4
Move Digit	D	7	D	1,4
Move Record	P	7,8	P	1,4
Move and Suppress Zeros	Z	7	Z	4
Move Zone	Y	7	Y	1,4
Multiply	@	7	@	4
No Operation	N	5,7,8	N	1,2,4
Scan			O	4
Select Stacker	K	5	K	2
Set Word Mark	,	7	,	1,4
Store A-Register	Q	7	Q	1,4
Store B-Register	H	7	H	1,4
Subtract	S	7	S	1,4
Translate	T	7,8	T	1,4
Zero and Add	?	7	?	1,4
Zero and Subtract	!	7	!	1,4

Figure G-35. Interruptible Instructions

that instruction. Only unchained operations can be interrupted. Figure G-35 is a list of interruptible operations.

The interrupt causes a program skip to address 182 in the processor, where the first instruction of the interrupt routine is located.

An interlock prevents interrupting while the system is in the interrupt routine. The ENABLE INTERRUPT AND BRANCH instruction resets the interlock at the end of the interrupt subroutine. The interrupt routine must include:

1. Store B-address register contents. The address in the B-address register must be decreased by four to establish the position of the interrupted main-program instructions. This new address is placed in the ENABLE INTERRUPT AND BRANCH instruction.
2. Saving program conditions (arithmetic overflow, high-low-equal compare and index locations), if they might be lost during the interrupt subroutine. After the interrupt subroutine, the program must restore these conditions.

Enable Interrupt

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	>

Function. The ENABLE INTERRUPT instruction is made up of K for the operation code and a bit configuration of 8-4-2 for the d-character. The interrupt subroutine is reset, and the program continues with the next sequential instruction.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms. The interrupt precedes a possible 1448 buffer-overflow by at least the number of milliseconds in time *A* of Figure G-36. It follows a scan operation no sooner than time *B* of Figure G-36. The minimum time between a scan operation and a possible 1448 buffer-overflow is time *C* of Figure G-36.

Character Rate of Terminal	Characters Per Second	
	14.8	60
Time A	67.5 ms	41 ms
Time B	59 ms	59 ms
Time C	132.5 ms	100.2 ms
Time D	128.5 ms	96.2 ms

Figure G-36. Interrupt Timing

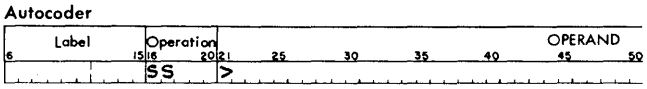
The maximum time for noninterruptible operations in the main program immediately following a scan operation is 132.5 ms (14.8 cps) minus the time taken by the interrupt routine. Assume that the time from interrupt to the execution of the scan operation is 4.0 ms.. The maximum length of a noninterruptible operation in the main program is then, about 128.5 ms (time *D* for 14.8 cps).

With a mixed system (more than one type of terminal) the shorter time applies.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	>bb	>bb

Example. Reset the interrupt indicator after an input/output request from a 1448 line (Figure G-37).



Assembled Instruction: K >

Figure G-37. Enable Interrupt

Disable Interrupt

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	<

Function. The DISABLE INTERRUPT instruction consists of K for the operation code and a bit configuration of B-A-8-4-2 for the d-character. This instruction, by setting the interrupt interlock, prevents the processor from honoring any interrupt requests. The interrupt interlock is reset by the ENABLE INTERRUPT instruction.

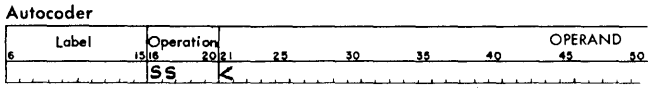
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	<bb	<bb

Example. Prevent (mask off) a 1448 interrupt while performing a disk operation (Figure G-38).



Assembled Instruction: K <

Figure G-38. Disable Interrupt

Enable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	>

Function. The ENABLE INTERRUPT AND BRANCH instruction, which is used to re-enter the main program, is made up of K as the operation code, a 3-character I-address representing the contents of the B-address register minus four at the time of interrupt, and the d-character with bit configuration 8-4-2. The interrupt routine interlock resets, and the program branches to the instruction address.

Word Marks. Word marks are not affected.

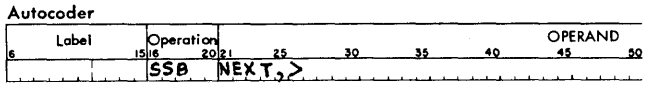
Timing.

Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Reset the interrupt indicator and return to a non-multiplexing routine by branching to NEXT (0800); Figure G-39.



Assembled Instruction: K 800 >

Figure G-39. Enable Interrupt and Branch

Disable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	<

Function. The DISABLE INTERRUPT AND BRANCH instruction is made up of K as the operation code, a 3-character I-address representing the next instruction, and a bit configuration of B-A-8-4-2 as the d-character. This instruction is the same as DISABLE INTERRUPT except that the next instruction is specified by the branch address.

Word Marks. Word marks are not affected.

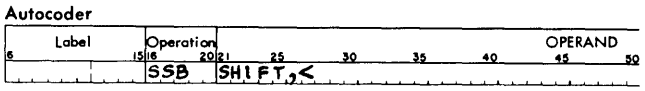
Timing.

Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Prevent a 1448 interrupt and branch to the disk-file routine labeled MORE (0900). See Figure G-40.



Assembled Instruction: K 900 <

Figure G-40. Disable Interrupt and Branch

Branch If End-of-Block

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	>

Function. The BRANCH IF END OF BLOCK instruction is made up of B as the operation code, an I-address, and the d-character > with bit configuration 8-4-2.

When a processor status character contains an end-of-block bit, the end-of-block indicator turns on. If it is ON, when the BRANCH IF END OF BLOCK instruction is executed, the next instruction is taken from that branch address. If the indicator is OFF, the program continues to the next sequential instruction.

The indicator is reset at the start of each scan operation.

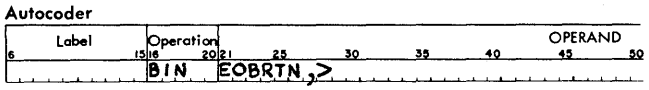
Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.
Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	>bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI



Assembled Instruction: B T13 >

Figure G-41. Branch if End of Block

Example. If the end-of-block indicator is ON, branch to routine beginning at EOBRTN (1313); Figure G-41.

Branch If Early Warning

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	<

Function. The BRANCH IF EARLY WARNING instruction is made up of B as the operation code, an I-address, and the d-character < with a bit configuration of B-A-8-4-2.

When the early-warning indicator is on, this instruction turns it off and causes a branch to the I-address (address of the chaining subroutine). There the program may first locate lines that require more assembly area, then provide new assembly blocks and add a link address to the previous blocks. When this type of storage allocation is used, the program issues this branch instruction after every scan operation.

The low-order positions of the storage block should contain at least the number of consecutive group marks that equals the maximum number of characters that can be transferred during a single scan operation for that line. The last group mark of the series can be followed by three positions for the link address provided by the chaining subroutine.

All group marks turn on the early-warning indicator. The first group mark provides the initial warning. Subsequent group marks are also used in one or more of these ways:

1. Locating the assembly block requiring chaining (by testing for absence of group marks).
2. Timing the buffers to allow for the actual delay in locating and chaining to the block in an early-warning condition.
3. Determining the penetration of data into the early-warning area (group-mark area).

Word Marks. Word marks are not affected.

Timing.

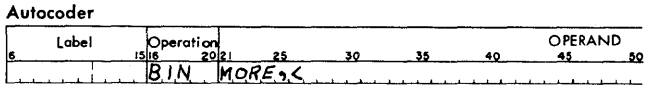
No branch, or branch without indexing: T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	<bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. If the early-warning indicator is ON, branch to area which contains chaining subroutine for storage allotment for additional input message areas. In Figure G-42, the early-warning area is labeled SHIFT (0700).



Assembled Instruction: B 700 <

Figure G-42. Branch on Early Warning

Direct-Data-Channel Interrupt

Direct-data-channel interrupt is standard with the transmission control unit (1448) attachment. It is not available on the direct data channel without the attachment.

With the direct-data-channel interrupt, an interrupt request in computer A (1440 with 1448) is made when computer B (host computer) indicates that it is requesting to read from computer A, or indicates that it is requesting to move data to computer B.

The main program is interrupted when an instruction is being read, but before the actual execution of that instruction. Only certain instructions can be interrupted. (See Figure G-35).

An interrupt request by the direct data channel causes the 1440 to start checking the instruction readouts for an interruptible point. This is accomplished by interrogating the fifth instruction (actually I_4) readout cycle for no B-register word mark.

The actual interrupt causes a program skip to address 181 where the first instruction of the interrupt routine should start.

The direct-data-channel interrupt request is reset when it actually causes the interrupt. If the 1448 interrupt request causes an interrupt first, the direct-data-channel interrupt is not reset, and a subsequent interrupt occurs.

Line Control

Time on the communication line is divided into two modes, line-control mode and text mode. The coded characters have a different meaning in each.

Line-Control Mode

In the line-control mode, the characters are interpreted as line-control signals, polling signals, and addressing signals. In this mode, signals control the transmission line, and are not read by the data processing components. Figure G-43 is a list of line-control characters.

Description	Symbol	Bit Configuration	Processor Character
End of Transaction (EOT)	Ⓒ	C-8-4-2-1	√‾ (tape mark)
End of Address (EOA)	Ⓓ	8-2-1	# (pound sign)
Negative Response (Control)	Ⓝ	B	- (hyphen)
Positive Response (Control)	Ⓨ	B-A-8-2-1	. (period)
Negative Response (Text)	ⓝ	B	- (hyphen)
Positive Response (Text)	ⓨ	B-A-8-2-1	. (period)
Positive Response (Inquiry)	ⓓ	8-2-1	# (pound Sign)
End of Block (EOB)	Ⓑ	A-8-2	≠ (record mark)

Figure G-43. Line-Control Characters

When a terminal receives an EOT (end-of-transaction) signal from the 1448, the terminal goes to, or remains in, the line-control mode. If the terminal is in selected status, it goes to a nonselected status.

Text Mode

In the text mode, the characters are interpreted the same as those that make up messages in the interchange between the 1448 and the terminal components. They consist of graphic characters, interstation-control characters (such as upper-case and line feed) and checking characters. For additional information, refer to *IBM 1448 Transmission Control Unit,* Form A24-3010.

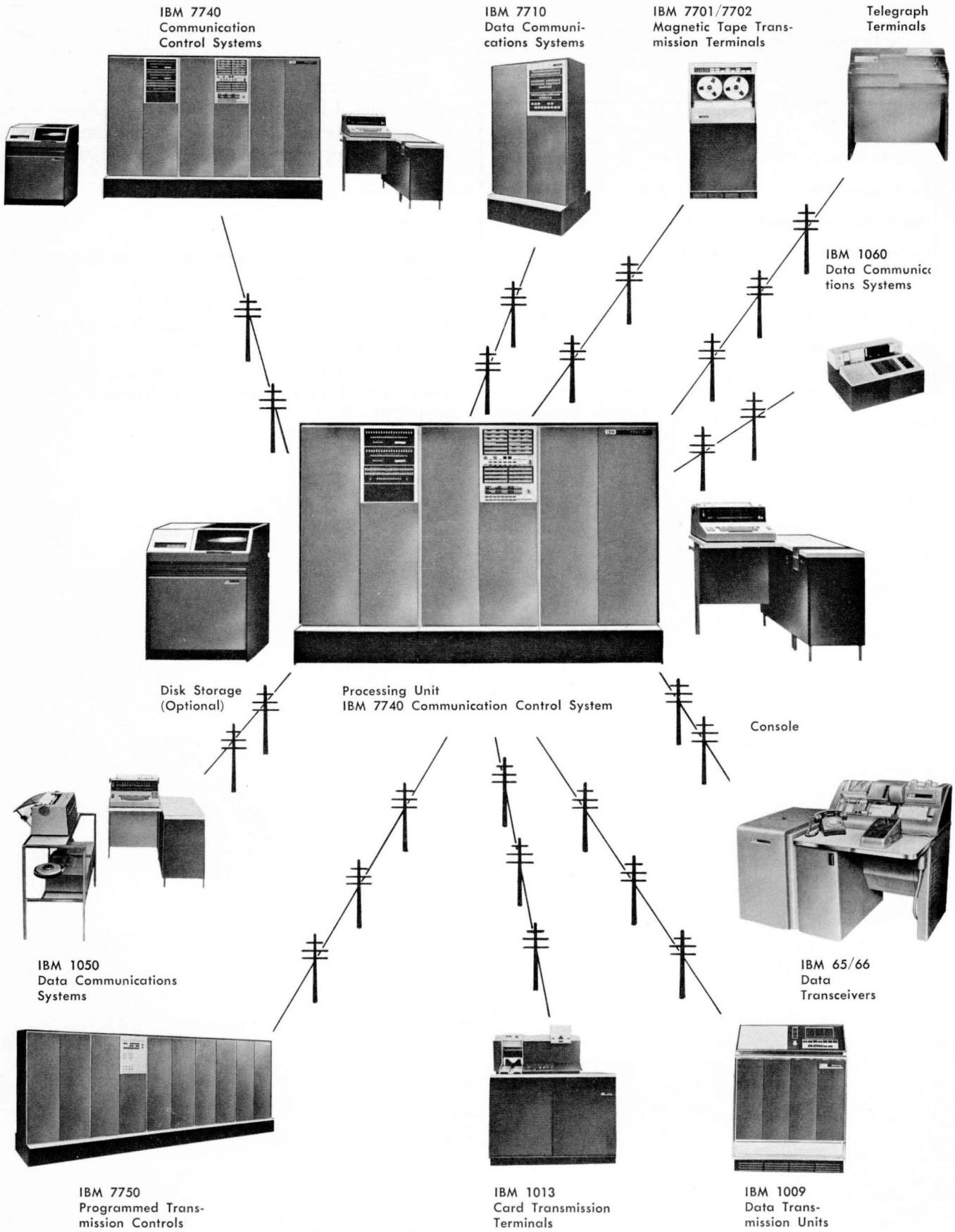


Figure G-44. Network Scope of an IBM 7740 Communication Control System

IBM 7740 Communication Control System

The IBM 7740 Communication Control System (Figure G-44), can be attached to the IBM 1440 through the serial input/output adapter (special feature).

The stored program in the 1440 has complete control over the transmission of data to and from the 7740 system. To facilitate this control, three types of instructions are used:

SIGNAL CONTROL instructions.

BRANCH IF INDICATOR ON instructions.

MOVE and LOAD instructions.

Refer to IBM *7740 Communication Control System*, Form A22-6753.

IBM 7740 Instructions

Instructions applying to the 7740 cannot be successfully chained.

Signal Control, or Signal Control and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		d
SSB	K	III	d

Function. The SIGNAL CONTROL instruction (Figure G-45), is used by the 1440 to:

1. Inform the 7740 system that the 1440 system requires it for the performance of a particular operation, or
2. Send a reply to the 7740 system to acknowledge the recognition of a specific condition.

Instruction *	Signal Sent to 7740 System
K B or K (III) B	Attention Response — given when the system recognizes an "Attention" signal sent from the 7740.
K C or K (III) C	Read Request — given when the system wants to read (receive) data from the 7740.
K D or K (III) D	Write Request — given when the system wants to write (send) data to the 7740.
K E or K (III) E	End Response — given when the system recognizes an "End" or "Unusual-End" condition signal sent from the 7740.
K F or K (III) F	Control Request — given when the system wants to send control data to the 7740.
K G or K (III) G	Sense Request — given when the system wants to receive status data from the 7740. This is normally done after the system receives an "Unusual-End" condition signal from the 7740.

* K (III) X form of instruction results in a branch to the specified I-address

Figure G-45. Signal Control Instruction Summary

If an I-address is specified in the instruction (mnemonic SSB), the signal-control function is performed, and the program branches to the address specified.

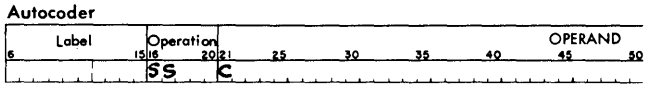
Word Marks. Word marks are not affected.

Timing.
No branch: T = .0333 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Signal Control	NSI	dbb	dbb
Signal Control and Branch (without indexing):	NSI	BI	blank
Signal Control and Branch (with indexing):	NSI	BI	NSI

Example. The system signals the 7740 requesting data from the 7740 (Figure G-46).



Assembled Instruction: K C

Figure G-46. Read Request

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	See Figure G-47

Function. This instruction and its associated d-characters are used by the system to check for various conditions on the 7740 system. When a tested condition is present, the program branches to the previously written subroutine. The BRANCH IF INDICATOR ON instruction d-characters and the tests they perform are shown in Figure G-47.

Word Marks. Word marks are not affected.

Timing.
No branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

d-character	Test Performed
1	Causes a branch to the specified I-address when a data transmission is ended and a condition is present of which the system must be aware. The condition remains ON until it is set OFF by executing a K E instruction.
2	Causes a branch to the specified I-address when a data transmission is successfully completed. The "Successful Completion" condition remains ON until it is set OFF by executing a K E instruction.
3	Causes a branch to the specified I-address when the 7740 system has received a Read, Write, Control, or Sense request from the system. The "Receive Request" condition remains ON until it is set OFF by executing a K E instruction.
4	Causes a branch to the specified I-address when the 7740 system wants the system to "service" it (attention signal). The attention signal remains ON until it is set OFF by executing a K B instruction.

Figure G-47. Branch If Indicator On d-Character Summary

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Tests for the attention signal. If the signal is present, branch to core-storage location 0385 (area labeled MSGST); see Figure G-48.

Instruction Name	Read Data	Read Data with Word Marks	Write Data with Word Marks	Write Data
Instruction Format	M (%A1) (BBB) R	L (%A1) (BBB) R	L (%A1) (BBB) W	M (%A1) (BBB) W
Function	Data without word marks received from 7740 system	Data with word marks received from 7740 system	Data with word marks sent to 7740 system	Data without word marks sent to 7740 system
Transmission Ended by	GM-WM sensed in system core storage or an "End" or "Unusual End" signal from the 7740 system			
Word Marks	Word marks are not transmitted	Word marks are transmitted		Word marks are not transmitted
Timing	T = .0111 (L_I + 1) ms + transmission and start time			
Address Registers After Operation	I-Add. Reg. NSI	A-Add. Reg. %11	B-Add. Reg. B + message length + 1	

Figure G-49. Read and Write Instruction Summary

Autocoder

Label	Operation	OPERAND
	B,IN	MSGST,,4

Assembled Instruction: B 385 4

Figure G-48. Branch If Attention Signal Indicator On

Read and Write

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%A1	BBB	R
MU	M	%A1	BBB	W
LU	L	%A1	BBB	R
LU	L	%A1	BBB	W

Function. The READ and WRITE instruction, M or L (%A1) (BBB) R or W, initiates the data transmission operation between the system and the 7740 system in the specified mode.

The parts of the instruction and their uses are:

M or L: The M or L operation code specifies whether the data transmission will be performed in the move mode or load mode. If the move mode (M op code) is specified, up to 7 bits per character (CBA8421) are involved in the data transmission. If the load mode (L op code) is specified, up to 8 bits per character (WM CBA 8421) are involved in the data transmission.

%A1: The A-address (%A1) specifies that data transmission between the system and the 7740 system will take place when the instruction is executed.

BBB: The B-address specifies the high-order position of the core-storage area involved in the data transmission.

R or W: A d-character R specifies a read operation. This d-character is used when the 7740 will send data. A d-character W specifies the write operation. This d-character is used when the 7740 will receive data.

Refer to Figure G-49 for a summary of the READ and WRITE instructions.

Example. Read data from 7740, without word marks, and place in core storage, beginning at location 0853 (area labeled INPDAT); see Figure G-50.

Autocoder

Label	Operation	OPERAND
	MU	%A1,,INPDAT,,R

Assembled Instruction: M %A1 853 R

Figure G-50. Read Data

IBM 1231 Optical Mark Page Reader

The IBM 1231 Optical Mark Page Reader (Figure G-51) provides a means of reading marked data from 8½" x 11" data sheets directly into the 1440 systems. The documents can be read at varying rates of speed, depending upon the mode switch setting. When set to CONTINUOUS, feeding is at a constant speed of 2000 documents per hour. When set to DEMAND, feeding is controlled by the computer program with speeds varying up to 1,600 documents per hour. The feeding mode selected depends upon the computer program control method used. Refer to *IBM 1231 Optical Mark Page Reader,* Form A21-9012.

IBM 1231 Instructions

The instructions described are for the control of the 1231 through the stored program in the 1440 system.

Instructions applying to the 1231 cannot be successfully chained.

Word Marks. Word marks are not affected by any 1231 branch instruction.

Timing. The following timing formulas apply to all 1231 branch instructions:

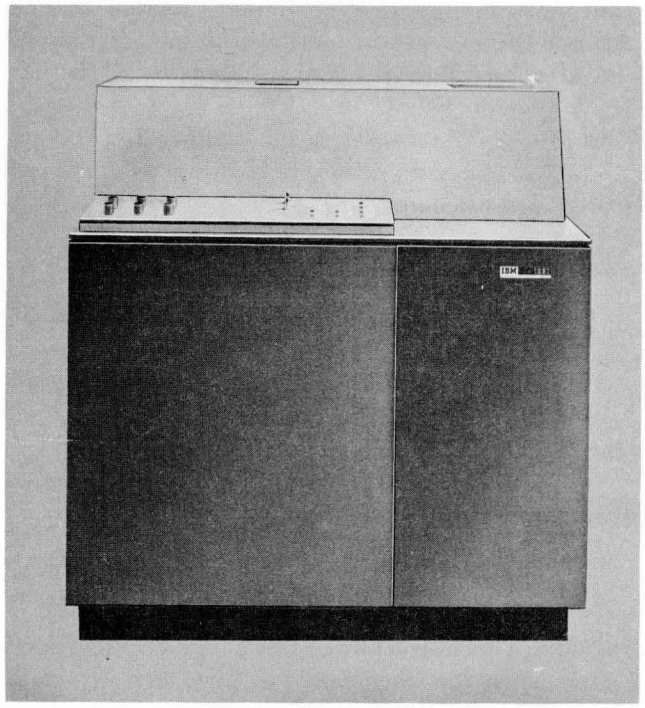


Figure G-51. IBM 1231 Optical Mark Page Reader

Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation (All Conditional Branch Instructions).

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Branch if Auto-Select

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	1

Function. This operation indicates that the document does not meet all the 1231 read conditions and that the 1231 has directed the document into the select stacker. When a document feeds through the read area, one of two conditions is sensed: an auto select or a buffer full. An auto select indicates the document did not satisfy the 1231 field checking switch settings. The 1231 can select a document itself if the internal editing conditions are not satisfied. If an auto select occurs, the machine clears the buffer and causes the next document to feed.

Branch if Buffer Full

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	2

Function. This operation indicates the document has passed all the 1231 internal program conditions and has been completely read. The buffer is full.

This indicator should always be program-tested first when entering the 1231 read subroutines from the main computer program.

Branch if 1231 Ready to Read

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	3

Function. This branch condition indicates that all normal operating conditions have been satisfied and the start key is pressed. This branch remains on

until the buffer is empty or until some interruption in the 1231 occurs (a manual stop, stacker full, hopper empty, jam or mis-feeding condition).

Branch if Hopper Empty

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	4

Function. This branch condition distinguishes an empty hopper from other 1231 conditions. It is normally used to initiate the end-of-job routine. This indicator is off except when the hopper is empty. The 1231 ready-to-read indicator is off when the empty-hopper condition is on.

Branch if Read Error and Over-Run Detection

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	5

Function. This indicates that data read from the 1231 has not been completely transferred to the computer. This condition is caused by a parity error detected by the 1231, or by a delay in the processor start-timing relationship. With this condition, the 1231 stops and waits for operator intervention. The indicator is turned off by pressing the reset or load keys on the 1231.

The document in error is the last one stacked in the normal stacker.

Branch if Timing-Mark Check

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	6

Function. This branch instruction indicates that the timing mark track was found in error, either by the presence of extra timing marks or absence of prescribed timing marks. If the control timing mark switch is set to YES, a timing mark check indication is sent to the computer 75 ms after a buffer-full indicator comes on. If the control timing mark switch is set to NO, the timing-mark check is sent to the computer immediately following a buffer-full indicator signal. This timing-mark indicator is turned off when the reset key is pressed to resume normal operations.

Move

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%S3	BBB	R

Function. This instruction causes data to be read from the 1231 storage into core-storage positions designated by the B-address in the instruction.

Word Marks. A word mark with a group mark must be one position to the right of the last position reserved in core storage for the data. Data word marks are not affected.

Timing. T = .0999 ms + access time + 1 delay-line cycle.

1 delay-line cycle = 3.582 ms.

Access time = 0 to 3.582 ms or average time of 1.791 ms.

Select Stacker

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	A

Function. This instruction causes a stacker-select operation when the document has been read correctly but fails the internal computer program test. A SELECT STACKER instruction must be given within 50 ms after a buffer-full signal is initiated.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Abb	Abb

Select Stacker and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	A

Function. This instruction causes a stacker-select operation when the document has been correctly read but fails the internal computer test. The next instruction address is contained in the I-address register. A

SELECT STACKER instruction must be given within 50 ms after a buffer-full signal is initiated.

Word Marks. Word marks are not affected.

Timing.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

IBM 1231 Program Control

Continuous Feed Mode

In the continuous-feed mode, the computer program must be designed to repetitively test the BUFFER-FULL and READY-TO-READ indicator at least once every 1800 ms when the 1231 is placed in operation on-line. The explanation of the block diagram (Figure G-52), is as follows:

Start

Document feeding is initiated by pressing the 1231 start key if the processing unit is in program-run status If the 1231 start key is pressed when the computer program is in a halt condition, document feeding is inhibited. After the first document is fed, feeding is continuous until an interruption occurs in either the 1231 or the processing unit.

Buffer Full

This indicator is turned on immediately following the reading of a complete data sheet. BUFFER-FULL is tested repetitively as part of a loop. The buffer-full signal is used as the program reference point for each succeeding document cycle.

Read

This instruction is used to transfer data from 1231 storage to the processing unit. The READ instruction must be executed within 150 ms following BUFFER-FULL. If the 1231 storage is still full when the next sheet is detected at the read head, the process-check light is turned on and the 1231 stops. The maximum time required to transfer the contents of the 1231 output storage to the processing unit is 7.2 ms.

Read Error

This branch comes on to indicate that a 1231 read-out error occurred during the transfer of data to the processing unit. When a read-out error occurs, READY-TO-READ is turned off and the process-check light turned on. The top data sheet in the normal stacker must be reprocessed because the contents of buffer storage is lost during read-out. At this point, operator intervention is required. The computer can, however, branch to other subroutines until the 1231 is again placed in ready-to-read status.

Select Condition in Document

This programmed check of data can be used to test any desired condition for which the document is selected. The decision to select the document must be given within 50 ms after a buffer-full indication.

Timing-Mark Check

This branch comes on to indicate that the timing-mark count did not agree with the setting of the timing-mark check switch. At least 75 ms must elapse between the signal of a buffer-full and the timing-mark check if the control-timing-marks switch is set to YES. A minimum of 75 ms can be used by the program in a process/timing loop. Any procesing of data from the document just read (before the timing-mark check is made) may require the recovery of data from various storage locations if a timing-mark check occurs. The signal of timing-mark check indicates that the information read from the document is invalid.

Auto Select

When this branch comes on, it indicates that the document just read was sent to the select stacker. When this condition arises, at least 1800 ms can elapse before the next READ instruction needs to be given. This branch, therefore, can be used as an exit to other processing routines.

Ready-to-Read

This indicator signifies that the 1231 is in an operating status, that all normal operating conditions are satisfied, and that the start key has been pressed. READY-TO-READ OFF indicates that an interruption has occurred in the 1231. This branch can provide an exit from the subroutine.

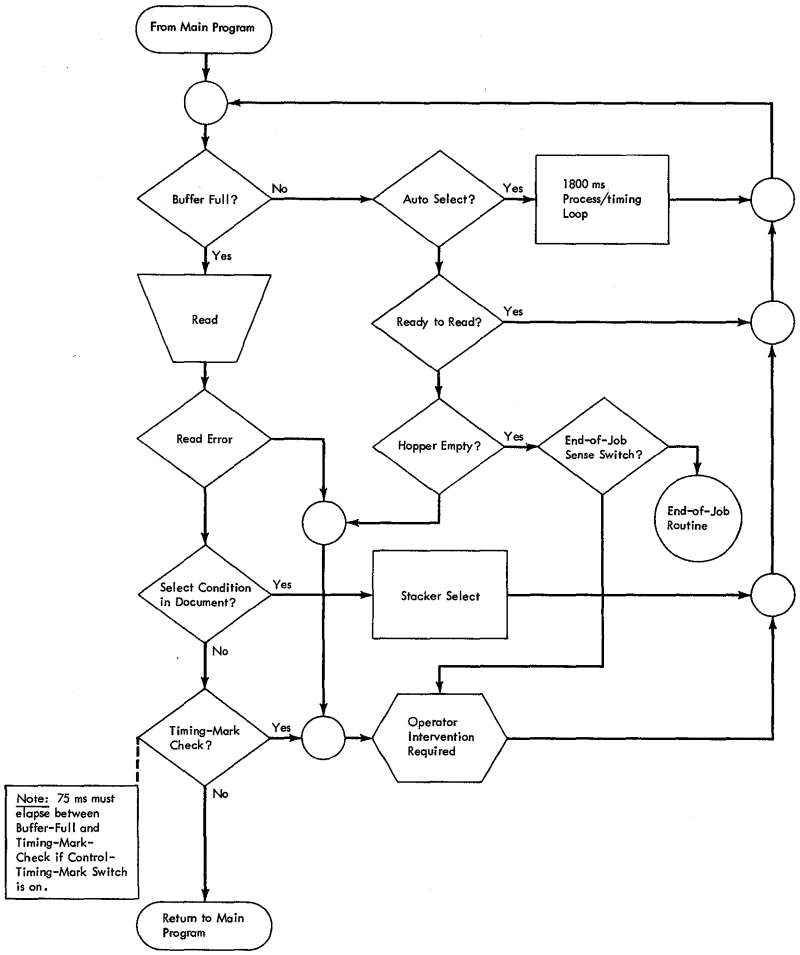


Figure G-52. IBM 1231 Program Control – Continuous Feed Mode

Hopper Empty

This branch indicates that no more documents remain in the feed. The operator should turn on the end-of-job sense switch when the last batch of documents is placed in the hopper.

Select Stacker

Certain data sheets may be selected at the programmer's option. The STACKER SELECT instruction must be executed within 50 ms after the signal of a BUFFER FULL.

On-Demand Feed

No special timing considerations are necessary when entering the subroutine in the on-demand mode. (Figure G-53). When the 1231 start key is pressed, the first document is fed past the read station and the data is stored in the 1231. The next document is not fed until the contents of the output storage is transferred to the computer.

Buffer Full

BUFFER FULL ON indicates that the document has been read and stored in the 1231. A successful test of this indicator is normally followed by a read instruction. BUFFER FULL should precede a test for READY TO READ in the on-demand mode, because it is possible that the ready to read indicator has been turned off after a document has been read due to a full stacker. This sequence of branch tests ensures that the last document stored in the 1231 just prior to a stop is transferred to the computer.

Ready-to-Read

This branch indicator comes on to show that the 1231 is in an operating status; all normal operating conditions are satisfied, and the start key has been pressed.

READY TO READ OFF indicates that an interruption has occurred in the 1231 and provides an exit from the subroutine when the 1231 stops.

Read

This instruction causes a transfer of data from the 1231 storage to the computer read-in area. A read instruction may be executed at any time following a BUFFER FULL indication. Execution of the read instruction in the on-demand mode also initiates the next document-feed cycle, unless there is a stop condition in the 1231.

Read Error

This indicator comes on to show that a 1231 read-out error occurred during the transfer of data from the 1231 to the computer. Because an incomplete record has been sent to the computer, the subroutine should include a provision to clear the read-in area. The top document in the normal stacker must be re-processed. Operator intervention is required for this condition. However, the computer can branch to other subroutines until the 1231 is again placed in READY TO READ status.

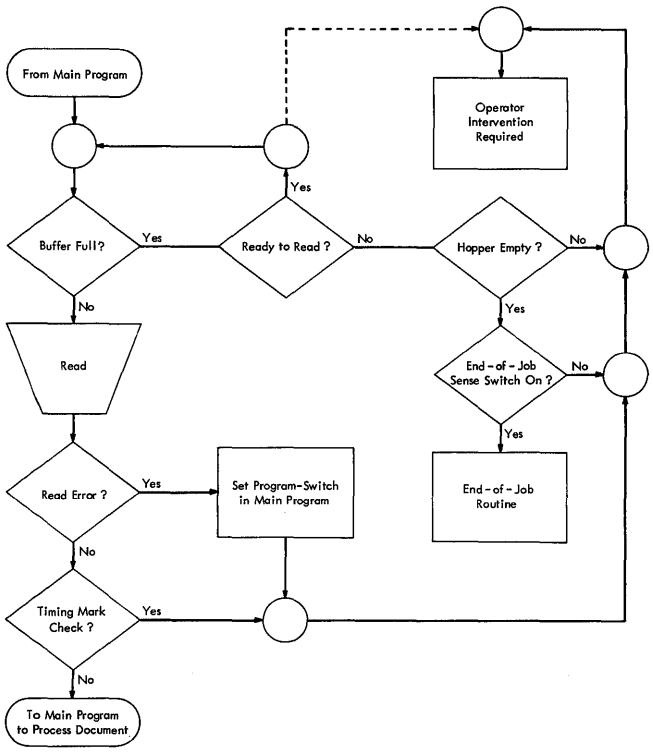
Hopper Empty

This branch indicator is set ON when the document hopper is empty. If this indicator is on, the program can be made to stop the system, or to branch to another routine.

Timing-Mark Check

This branch indicator comes on when the timing-mark track on the data sheet just read was found to be unacceptable, either by extra or missing timing marks.

At least 75 ms must elapse between the buffer-full signal and the timing-mark check, when the control-timing-marks switch is set to YES. If the documents being processed do not contain control timing marks the control-timing-marks switch can be set to NO, and the timing-mark check can be made immediately after the buffer-full indication. This branch is turned off when RESET is pressed to resume normal operation.



Note: 75 ms must elapse between Buffer-Full and Timing-Mark-Check if Control-Timing-Mark switch is set to YES.

Figure G-53. IBM 1231 Program Control – On-Demand Mode

IBM 7770 Audio Response Unit, Model 1, Programming

The IBM 7770 Audio Response unit provides a spoken reply message to a digital inquiry entered by the user. This reply is formed by selecting pre-recorded words in a specific sequence to form a message. Because the 7770 is merely an input, storage, and output device, the processing unit must generate and issue the word sequence to the 7770. This is accomplished by a user-written program. Refer to *IBM 7770 Audio Response Unit,* Form A22-6800.

The 7770 message-control portion of the program should consist of the following phases:

Inquiry Input: Read the inquiry into core storage. Determine if it is a test message. Check inquiry for proper length.

Evaluation of Input: Determine the information required. Check security code, if present.

Information Retrieval: Obtain requested information from system resources (core storage, disk or tape files, etc.). GET output-message format, if required.

Message Assembly and Output: Extract requested information from the account file, and place in the proper order in the output area. Write the output area to the 7770.

The IOCS for 1311 file provides the necessary GET and PUT instruction macros for the retrieval and placement of 7770 input and output information. The IOCS requirements must be properly defined. See *IOCS Usage.*

Figure G-54 shows the program necessary to service the 7770 in a disk-storage environment. Perform the unconditional branch of the polling subroutine about every 3 to 5 seconds of main-line program time (a maximum of 10 seconds is permitted). This time depends on the type of main-line program being run. The wide latitude of time between polling operations is provided because polling and servicing routines should allow service of all waiting inquiries before returning to the main-line program.

Because the polling and servicing subroutines together probably will not exceed 300 milliseconds (including one disk seek to GET the record about which inquiry is being made), the delays in CPU service will not cause excessive wait-time for the calling party.

The BRANCH IF TAPE MARK instruction is particularly important. When the 7770 does not have an inquiry to be processed, the read will return a first-character-tape-mark response. This must be interpreted by the user's program as no-service-required and a return to the main-line program can take place at this point.

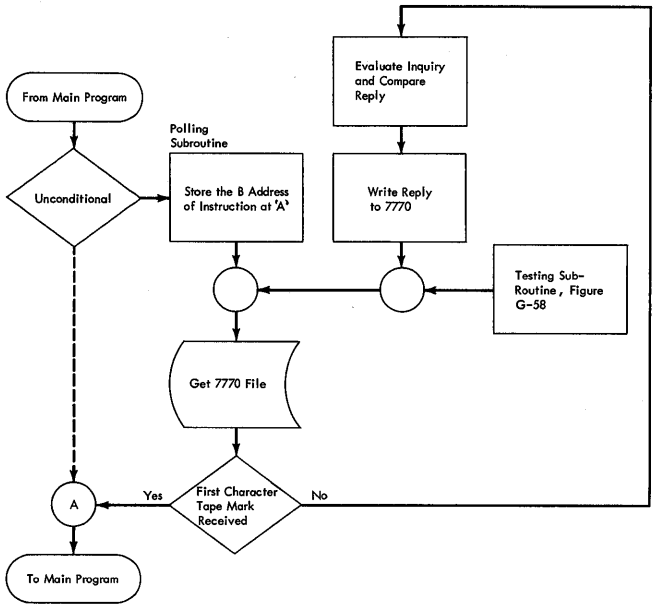


Figure G-54. Inquiry Processing Subroutine

IBM 7770 Instructions

Generally, the same instructions and error routines used with disk storage pply to the 7770. The following is a more detailed explanation of the program phases. See the *IBM 1311 Disk Storage Drive* section.

Inquiry Input

The inquiry is brought into core storage by a read-file instruction sequence (Figure G-54). In the event of transmission errors, a reread is possible in exactly the same way as a file reread. If the reread does not correct the error condition, a standard message can be sent to the 7770 indicating the necessity of a redial by the calling party. Except in the case of the transfer of a first-character-tape-mark to core storage, the user must return a reply to the 7770. The 7770 must always receive a write after a read, if a tape mark is not received.

Because of the unique character of the 7770, one input line can be used for testing while the other ones

are performing their normal function. For this reason, a testing routine should be built into the user's normal servicing program. This is simplified because the test message entered from the CE panel is one character followed by a group mark (ǂ); no other input messages have this characteristic. In addition, no actual processing of this test message is required. It is read into core storage, recognized by the program, and returned unmodified to the 7770. Because the inquiry is already in core storage (through the polling routine), a routine similar to the one shown in Figure G-55 might be used.

If the inquiry is *not* of a prescribed length, an error condition exists. There may or may not be a test made at the discretion of the user. In applications requiring inquiries of various lengths due to the particular characteristics of the access of stored information (i.e., in banks, mortgage accounts might be six digits and savings accounts seven digits), additional verification techniques may be required. However, if all inquiries are to be of one specific length, record length might be considered adequate.

Other record-verifying techniques may be used at the option of the user. In the event of an invalid inquiry, an error message (such as redial) must be sent by the user's program to the 7770 to release it for further line servicing. After a read-back check with no errors, the program must delay for about 2.8 ms. before proceeding to the next read. This delay is required because of 7770 timing, and can be accomplished by a simple timing loop subroutine.

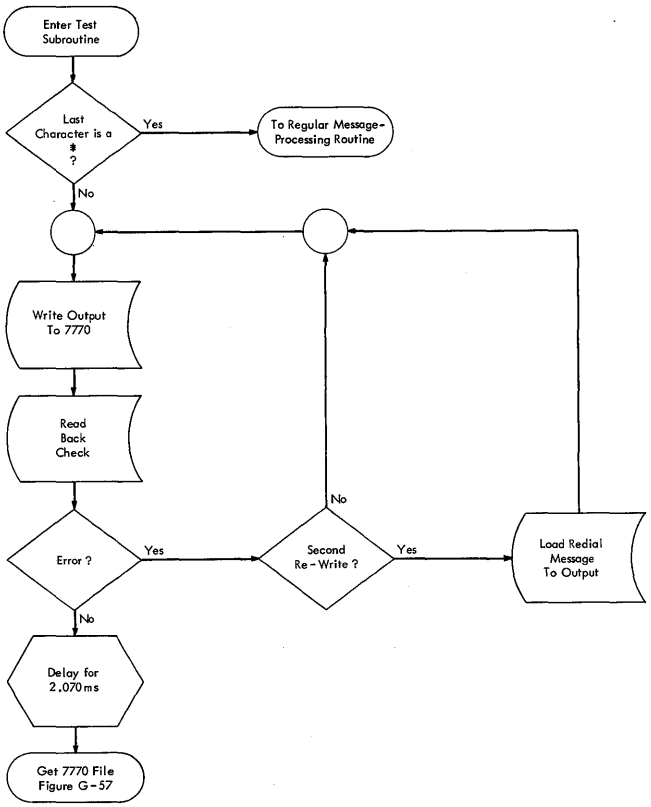


Figure G-55. Testing Subroutine

Evaluation of Input

The information required by the inquiry from file storage must be indicated by the inquiry if many different types of spoken replies are to be given. For example, a retailer may desire to know the amount of a certain stock number on hand. To obtain the information desired, and only that information, assign transaction codes as part of the inquiry (Figure G-56).

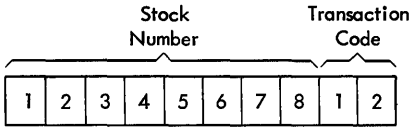


Figure G-56

The length of the stock number and transaction code is variable at the discretion of the user. The user's program must interpret the meaning of both the stock number and transaction code. The evaluation of the inquiry might also involve the length of the inquiry. See *Inquiry Input.* In some applications, the user may desire to have the transaction code located elsewhere in the inquiry. Because this program is interruptible, the placement is not limited.

On occasion, some stored information must be inaccessible except to certain persons. To perform this function, several methods may be used. See *IBM 7770 Audio Response Unit — Model 1,* Form A22-6800. Some security codes may appear as part of the data from which the response message is to be composed. Others may be part of the inquiry itself in much the same manner as the transaction code.

If a blank is received as the first character of an inquiry read from the 7770, the user's program should interpret this as a transmission error and return a redial message.

Information Retrieval

Once the actual inquiry has been evaluated, a response message must be composed. To do this, the general source of information, described by the inquiry, is brought into core storage. The general source of information in a file system would probably be one or more sections from disk.

To simplify message assembly, some systems may utilize a message format method. With the inquiry transaction code evaluated, it would be known that a certain form of reply is required. Prepare this form in advance and store it either in core storage or on file, to be moved to the output area. There it is added to information extracted from the data brought into core storage to form the completed response message.

Message Assembly and Output

A number of methods may be employed for the composition of response messages. An understanding of vocabulary organization is necessary. See *IBM 7770 Audio Concepts and Vocabulary,* Form A22-6805.

The response message may be assembled for output in the same buffer area reserved for input because only one message is handled at a time. The maximum length of an inquiry is 40 characters. The maximum length of the response is 38 characters plus one group mark. The response message is sent to the 7770 through a write disk to disk control field (DCF) 8000000001. A read-back check is issued after a write. The transmission error indicator must be tested immediately and, in the event of an error, a user-written error routine must issue a rewrite within 2.2 ms of the read-back check. If longer delay before rewrite is desired, a seek should be given within 2.2 ms.

Note 1: Because a calling party may dial or otherwise unintentionally put in an incorrect inquiry code, the user should program-protect the caller from getting information that is incorrect. Because it is generally not possible to determine if a number has been incorrectly dialed until a response is made to that number, it is advisable to provide some means of checking the input number. One of the easiest methods of doing this is to program the IBM 7770 to repeat back to the calling party the number as received by the 7770. If the number repeated back is not what the caller expects, the caller should place the inquiry again. Other checking methods may be used, but the repeat-back is one of the simplest.

The group mark (end-of-message character) must appear as the last character of any message. A message may contain a maximum of 38 address characters plus one group mark.

Note 2: If the Extended Vocabulary feature is installed on the 7770, the maximum inquiry length decreases to 36 characters and the response length to 35 characters plus group mark.

IOCS Usage

The 7770 operates with the 1311 File IOCS program packages. See *Autocoder for IBM 1440: Operating Procedures,* Form C24-3011 for 1440 applications. The 7770 appears as a slave 1311 file.

Figure G-57 shows the minimum definitions.

AUTOCODER

Line	Label	Operation	OPERAND
23 25	26 35	36 40	41 45 50 55 60
0,1,		D,I,O,C,S,	
0,2,	I,O,D,E,V,I,C,E,S,		D,I,S,K,
0,3,	P,R,O,C,E,S,S,T,Y,P,E,		R,A,N,D,O,M,
0,4,	D,I,S,K,D,R,I,V,E,S,		8,
0,5,		D,T,F,	
0,6,	F,I,L,E,T,Y,P,E,		D,I,S,K,,R,A,N,D,O,M,,I,N,P,U,T,
0,7,	U,P,D,A,T,E,		Y,E,S,
0,8,	I,O,A,R,E,A,		
0,9,	N,S,E,C,T,O,R,S,		1,
1,0,	N,R,E,C,O,R,D,S,		0,
1,1,	N,A,T,O,P,A,C,K,		8,

Figure G-57. Minimum 7770 IOCS Definitions

The disk-control field of the 7770 is 8000000001. A GET file instruction brings information into core storage from the 7770 buffers. The information retrieved is inspected by a user's program and appropriate action is taken.

If the GET brings all BCD blanks into storage, the user's program should evaluate all BCD blanks as a no-service-required indication and return to his original program. If the GET returns data from the 7770 other than BCD blanks, the user program must branch to a subroutine that evaluates the data and takes appropriate action to assemble a digital response message to be returned to the 7770 on a PUT file. To avoid lengthy delays in responding to the 7770 by the processing unit after a PUT has been satisfactorily completed, issue another GET to the 7770 (after the prescribed delay) to ascertain if service is not required by another input line.

If advanced programming is specified in the control card at assembly time, the error routine IOCEHT in the IOCS must be changed to a NO OP instruction, and the HALT AND BRANCH instruction at IOCEHT+7 must be changed to an UNCONDITIONAL BRANCH instruction to the user's error routine.

If advanced programming is not specified in the control card, this procedure is followed, except that the labels IOCEHT and IOCEHT+7 are replaced with IOCMHT and IOCMHT+7, respectively.

To determine if a 1311 operation or a 7770 operation has caused an error, a program switch must be set by the user prior to issuing a 7770 macro. Thus the user's error routine could test this switch to determine whether to treat a given error as a 1311 error or as a 7770 error.

Large-Vocabulary Programming

Special consideration must be given in programming the 7770 if more than 63 words of vocabulary are

available on the vocabulary drum. A BCD character ordinarily has six information bits that decode to a maximum of 64. The addition of one more information bit is required and that is the word mark. To transfer information to or from the 7770 with word marks, the WDW and RDW instructions are used.

This mode of operation is not supported by the 1311 file IOCS and must be programmed by the user. However, the PUT used by the 7770 may be altered by the user to permit word mark transfers as follows:

1. Load an L with a word mark into FILENAME+16 prior to PUT macro.
2. After the PUT macro, restore the M with word mark.
3. Be certain that a group mark with a word mark is in the proper position to satisfy move and load mode conditions for checking length.
4. Prior to the issuing of the GET macro, the user must supply an address of the form xxxxxx0 to the storage location located at the IOCS label IOCADR (this does not have to be done prior to the issuing of the PUT macro).

available on the vocabulary drum. A BCD character ordinarily has six information bits that decode to a maximum of 64. The addition of one more information bit is required and that is the word mark. To transfer information to or from the 7770 with word marks, the WDW and RDW instructions are used.

This mode of operation is not supported by the 1311 file IOCS and must be programmed by the user. However, the PUT used by the 7770 may be altered by the user to permit word mark transfers as follows:

1. Load an L with a word mark into FILENAME+16 prior to PUT macro.
2. After the PUT macro, restore the M with word mark.
3. Be certain that a group mark with a word mark is in the proper position to satisfy move and load mode conditions for checking length.
4. Prior to the issuing of the GET macro, the user must supply an address of the form xxxxxx0 to the storage location located at the IOCS label IOCADR (this does not have to be done prior to the issuing of the PUT macro).

IBM 1285 Optical Reader

The IBM 1285 Optical Reader Model 1 (Figure G-58), serves as an input device for the IBM 1440 Data Processing System. The 1285 reads printed paper tapes such as those produced on cash registers and adding machines. Using advanced optical-recognition techniques to read *directly* from the source document of many business transactions, the 1285 eliminates much of the time required by a system in which information is punched into cards before being entered into the system.

Refer to *IBM 1285 Optical Reader,* Component Description, Form A24-3256, for additional information.

IBM 1285 Instructions

Instructions applying to the 1285 cannot be successfully chained.

Read in Move Mode

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%VI	BBB	R

Function. This instruction reads either one character or one line into core storage from the 1285. When header or full-line correction information is entered from the keyboard, characters are transmitted in single-character mode – one character for each read instruction. The character is entered in the core-storage location specified by the B-address and the operation is terminated. Assuming the operator keys in information from left to right, the B-address of this instruction should be modified by +1 for each subsequent read operation.

Figure G-58. IBM 1285 Optical Reader

When data is read from the journal tape, characters are transmitted in line mode, one line of information for each read instruction. The line is read from right to left, with the first character being read into the core-storage location specified by the B-address. The contents of the B-address register are automatically modified by −1 before transferring each subsequent character.

A character entered from the keyboard for single-character on-line correction is handled in the same manner as one read from the tape in this line-reading mode.

A read operation is normally terminated when the reader senses the left margin of the tape. When this occurs, a group mark is automatically inserted in the core-storage position to the left of the last character read, and the end-of-line indicator is turned on. If a group mark with word mark is detected in core storage before reading the last character of the line, the read operation is terminated, and the end-of-line branch indicator is not turned on.

A read operation is also terminated if the branch on error indicator is set on.

Word Marks. Word marks are not affected. A group mark with a word mark will terminate the read operation.

Timing. See IBM 1285 Timing Considerations section.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%51	B_p-length of line.
or		
NSI	%51	GMWM-1

Form A24-3116-3
Page revised 12/66
By TNL N24-0392

IBM 1285 Optical Reader

The IBM 1285 Optical Reader Model 1 (Figure G-58), serves as an input device for the IBM 1440 Data Processing System. The 1285 reads printed paper tapes such as those produced on cash registers and adding machines. Using advanced optical-recognition techniques to read *directly* from the source document of many business transactions, the 1285 eliminates much of the time required by a system in which information is punched into cards before being entered into the system.

Refer to *IBM 1285 Optical Reader,* Component Description, Form A24-3256, for additional information.

IBM 1285 Instructions

Instructions applying to the 1285 cannot be successfully chained.

Read in Move Mode

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%VI	BBB	R

Function. This instruction reads either one character or one line into core storage from the 1285. When header or full-line correction information is entered from the keyboard, characters are transmitted in single-character mode – one character for each read instruction. The character is entered in the core-storage location specified by the B-address and the operation is terminated. Assuming the operator keys in information from left to right, the B-address of this instruction should be modified by +1 for each subsequent read operation, by a subroutine. Such a subroutine is necessary because of the single-character transmission from the keyboard to core in these modes. The subroutine should be entered when the reader first becomes ready (branch on indicator 7). Wait loops for characters (indicator 5) should include repeated tests for Reader Ready (7), for Ready to Read a line (6, indicating the end of data input for the subroutine), and for Reader Error (1, indicating operator Cancel Enter). If Reader Error occurs in this subroutine, it is advisable to notify the operator through a printout or coded halt, and to re-initialize the subroutine. The subroutine should also be entered after giving a Set Correction Mode instruction to begin Full-Line Correction or to handle re-entry of Header or Full-Line Correction data for verification.

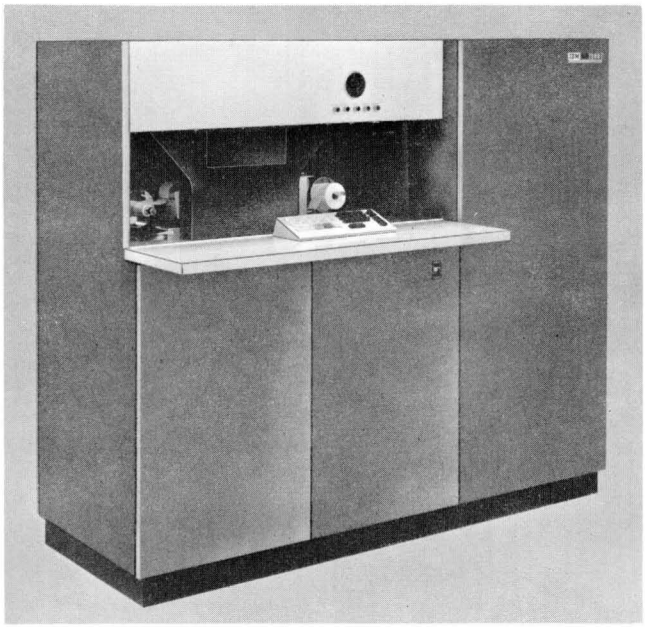


Figure G-58. IBM 1285 Optical Reader

When data is read from the journal tape, characters are transmitted in line mode, one line of information for each read instruction. The line is read from right to left, with the first character being read into the core-storage location specified by the B-address. The contents of the B-address register are automatically modified by −1 before transferring each subsequent character.

A character entered from the keyboard for single-character on-line correction is handled in the same manner as one read from the tape in this line-reading mode.

A line read operation is normally terminated when the reader senses the left margin of the tape. When this occurs, a group mark is automatically inserted in the core-storage position to the left of the last character read, and the end-of-line indicator is turned on. If a group mark with word mark is detected in core storage before reading the last character of the line, the read operation is terminated, and the end-of-line branch indicator is not turned on.

Example. Read the 1285 keyboard and place the character in the core-storage location labeled KB1285 (0881). The enter key must be pressed after each character key is pressed (Figure G-59).

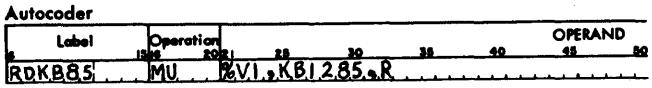


Figure G-59. Read IBM 1285 in Move Mode

Read in Load Mode

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%V1	BBB	R

Function. This instruction functions in the same manner as the read in move mode instruction except that, in line reading, word marks accompany characters sent from the keyboard in on-line reject correction. This allows the CPU to distinguish between characters read from tape and characters entered from the keyboard.

Go to Next Line

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		G
SSB	K	III	G

Function. This instruction causes the reader to advance to the next line. The normal procedure is to test the end-of-line branch indicator after a read instruction and, if on (indicating a valid read), to issue this instruction to cause the reader to begin scanning for the next line. If not given after a read operation is terminated, the next read instruction will reread the line.

The next instruction executed is that specified by the I-address, if supplied, or the next instruction in sequence, if no I-address is specified.

Word Marks. Word marks are not affected.

Timing.

Go to next line: T = .0333 ms.

Go to next line and branch (without indexing): T = .0666 ms.

Go to next line and branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Go to next line:	NSI	Gbb	Gbb
Go to next line and branch (without indexing):	NSI	BI	blank
Go to next line and branch (with indexing):	NSI	BI	NSI

Example. Cause the 1285 to advance to the next journal-tape line, and branch unconditionally to a subroutine labeled RDTAPE (1286) specified by the I-address (Figure G-60).

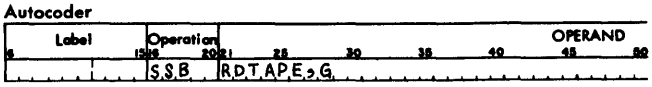


Figure G-60. Go to Next Line and Branch

Set Correction Mode

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		C
SSB	K	III	C

Function. This instruction causes the 1285 to go into a line display and sets up controls for character-by-character reading from keyboard entry. If the first line of the tape has not yet been read, the enter light turns on, indicating to the operator that he should enter header data. If at least one line of the tape has been read, the reject light turns on, indicating to the operator that he should do a full-line correction.

The next instruction executed is either specified by the I-address or the next sequential instruction (if no I-address is specified).

Word Marks. Word Marks are not affected.

Timing.

Set correction mode: T = .0333 ms.

Set correction mode and branch (without indexing): T = .0666 ms.

Set correction mode and branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Set correction mode:	NSI	Cbb	Cbb
Set correction mode and branch (without indexing):	NSI	BI	blank
Set correction mode and branch (with indexing):	NSI	BI	NSI

A read operation is also terminated if the branch on error indicator is set on.

Word Marks. Word marks are not affected. A group mark with a word mark will terminate the read operation.

Timing. See IBM 1285 Timing Considerations section.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%51	B_p-length of line.
or		
NSI	%51	GMWM-1

Example. Read the 1285 keyboard and place the character in the core-storage location labeled KB1285 (0881). The enter key must be pressed after each character key is pressed (Figure G-59).

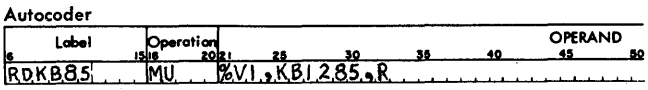


Figure G-59. Read IBM 1285 in Move Mode

Read in Load Mode

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%V1	BBB	R

Function. This instruction functions in the same manner as the read in move mode instruction except that, in line reading, word marks accompany characters sent from the keyboard in on-line reject correction. This allows the CPU to distinguish between characters read from tape and characters entered from the keyboard.

Go to Next Line

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		G
SSB	K	III	G

Function. This instruction causes the reader to advance to the next line. The normal procedure is to test the end-of-line branch indicator after a read instruction and, if on (indicating a valid read), to issue this instruction to cause the reader to begin scanning for the next line. If not given after a read operation is terminated, the next read instruction will reread the line.

The next instruction executed is that specified by the I-address, if supplied, or the next instruction in sequence, if no I-address is specified.

To accomplish line skipping, the program should wait for the Ready-to-Read-a-Line indicator (6) before issuing the next Go To Next Line.

Word Marks. Word marks are not affected.

Timing.

Go to next line: T = .0333 ms.

Go to next line and branch (without indexing): T = .0666 ms.

Go to next line and branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Go to next line:	NSI	Gbb	Gbb
Go to next line and branch (without indexing):	NSI	BI	blank
Go to next line and branch (with indexing):	NSI	BI	NSI

Example. Cause the 1285 to advance to the next journal-tape line, and branch unconditionally to a subroutine labeled RDTAPE (1286) specified by the I-address (Figure G-60).

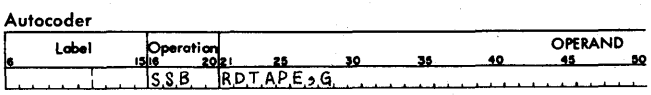


Figure G-60. Go to Next Line and Branch

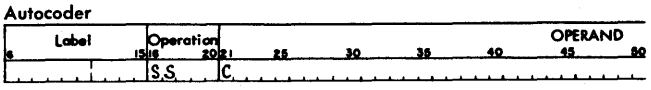
Set Correction Mode

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		C
SSB	K	III	C

Function. This instruction causes the 1285 to go into a line display and sets up controls for character-by-character reading from keyboard entry. If the first line of the tape has not yet been read, the enter light turns on, indicating to the operator that he should enter header data. If at least one line of the tape has been read, the reject light turns on, indicating to the operator that he should do a full-line correction.

Example. Cause the 1285 to go into line display and set up controls for character-by-character entry from keyboard (Figure G-60.1).



Assembled Instruction: K C

Figure G-60.1. Set Correction Mode

Mark a Line

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		M
SSB	K	III	M

Function. This instruction causes the line that the reader is on to be marked by the reject line marker. The line is transported to the top of the scan window and marked *after* the next GO TO NEXT LINE instruction. The MARK A LINE instruction can be given at any time after the ready-to-read-a-line indicator is turned on and before the GO TO NEXT LINE instruction.

Note: If the line is re-read (a GO-TO-NEXT-LINE instruction is not given), the effect of this instruction is cancelled.

The next instruction executed is that specified by the I-address, or the next sequential instruction (NSI) if no I-address is specified.

Word Marks. Word Marks are not affected.

Timing.

Mark a line: T = .0333 ms.

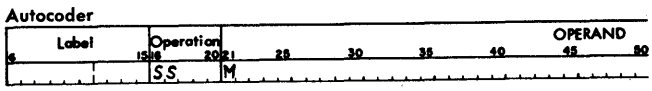
Mark a line and branch without indexing: T = .0666 ms.

Mark a line and branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Mark a line:	NSI	Mbb	Mbb
Mark a line and branch (without indexing):	NSI	BI	blank
Mark a line and branch (with indexing):	NSI	BI	NSI

Example. Cause the reader to mark the line it is now reading, after the next GO-TO-NEXT-LINE instruction (Figure G-61).



Assembled Instruction: K M

Figure G-61. Mark a Line

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. This instruction tests for the IBM 1285 operational conditions specified by the d-character. If the indicator is on, the next instruction is taken from the I-address. If off, the program goes to the next sequential instruction.

d-character	*Indicator*
1	ERROR
2	END-OF-LINE
3	READER-TRANSPORTING
4	MARKED-LINE
5	HEADER-INFORMATION
6	READY-TO-READ-A-LINE
7	READER-READY
8	END-OF-FILE

Indicators

Branch if Error. This indicator (d-character 1) is turned on if any of the following conditions exist. This indicator remains on until tested.

- A process check occurs in the processing unit during a read operation.
- A skew error occurs during a read operation.
- The scanner is unable to follow a line due to extraneous material on the tape during a read operation.
- A reject display exceeds the time limit.
- A line of header or a full-line of correction data is cancelled (cancel-enter sequence) from the reader.

Branch if End of Line. This indicator (d-character 2) is turned on after the last character of a line is transferred to the processing unit and the reader senses the left edge of the tape. If this indicator is not on when a read operation is completed, an error condition may exist, depending on the program and the tape format. This indicator is turned off by the go-to-the-next-line instruction, or by re-reading the same line.

The next instruction executed is either specified by the I-address or the next sequential instruction (if no I-address is specified).

Purposes:

1. To call for re-entry of Header or Full-Line data if verification is used.
2. To call for Full-Line Correction. This is recommended in the event of persistent (10 tries) Error indications after reading a line, or the presence of a reject symbol (@) entered in Single-Character Correction. The latter may be indicated by the presence of a reject symbol with a word mark in core storage after a Read in Load Mode, or the presence of the Reject Character in Line (4) indicator when a sense switch indicates the use of on-line reject correction.

Word Marks. Word Marks are not affected.

Timing.

Set correction mode: T = .0333 ms.

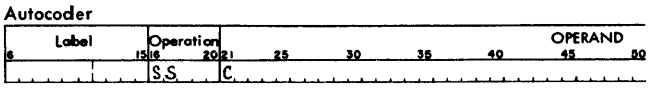
Set correction mode and branch (without indexing): T = .0666 ms.

Set correction mode and branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Set correction mode:	NSI	Cbb	Cbb
Set correction mode and branch (without indexing):	NSI	BI	blank
Set correction mode and branch (with indexing):	NSI	BI	NSI

Example. Cause the 1285 to go into line display and set up controls for character-by-character entry from keyboard (Figure G-60.1).



Assembled Instruction: K C

Figure G-60.1. Set Correction Mode

Function. This instruction causes the line that the reader is on to be marked by the reject line marker. The line is transported to the top of the scan window and marked *after* the next GO TO NEXT LINE instruction. The MARK A LINE instruction can be given at any time after the ready-to-read-a-line indicator is turned on and before the GO TO NEXT LINE instruction.

Note: If the line is re-read (a GO-TO-NEXT-LINE instruction is not given), the effect of this instruction is cancelled.

The next instruction executed is that specified by the I-address, or the next sequential instruction (NSI) if no I-address is specified.

Word Marks. Word Marks are not affected.

Timing.

Mark a line: T = .0333 ms.

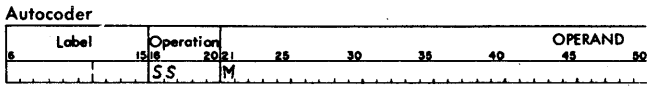
Mark a line and branch without indexing: T = .0666 ms.

Mark a line and branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Mark a line:	NSI	Mbb	Mbb
Mark a line and branch (without indexing):	NSI	BI	blank
Mark a line and branch (with indexing):	NSI	BI	NSI

Example. Cause the reader to mark the line it is now reading, after the next GO-TO-NEXT-LINE instruction (Figure G-61).



Assembled Instruction: K M

Figure G-61. Mark a Line

Mark a Line

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		M
SSB	K	III	M

Force On-Line Correction

Instruction Format.

	Mnemonic	*Op Code*	*I-Address*	*d-character*
A	SS	K		F
A	SSB	K	xxx	F

Branch if Reader Transporting. This indicator (d-character 3) is turned on when the transport mechanism is started to bring a new segment of tape over the scan window. It is turned off when the transport mechanism is stopped. Note that a minimum of 1 millisecond elapses between execution of the GO-TO-NEXT-LINE instruction and the beginning of the transport operation. Also, this indicator is turned off as soon as the transport stops, leaving about 6 milliseconds until the ready-to-read-a-line indicator is turned on.

This indicator may be used to determine if sufficient time is available to execute other instructions. This indicator is also on when displaying a line.

Branch if Marked Line. This indicator (d-character 4) is turned on if the last line read will be marked by the reject line marker. If the MARK A LINE instruction was given, or if the line contains a reject symbol (@), it will be automatically transported to the reject line marker after the next GO TO NEXT LINE instruction is given.

This indicator is reset by a GO TO NEXT LINE instruction, or by the re-reading of the line (either line read or full-line correction).

Branch if Header Information. This indicator (d-character 5) is turned on when a character of header information or full-line correction is entered from the keyboard. It is reset for each character read by the program. As long as this indicator is on, the reader will not respond to the start key to begin processing the journal tape. When no more information is to be entered, the start key is pressed, and the ready-to-read-a-line indicator is turned on.

Branch if Reader Ready to Read a Line. This indicator (d-character 6) is turned on when the reader has found a line and is ready to read it. The indicator is turned off by a go to next line instruction, or by loss of the reader-ready condition.

Note: When a line is found, the scanner enters the normalizing mode and continues from character to character in the line until a read instruction is given. At that time, the flying spot sweeps back to the low-order character.

To keep read time to a minimum, a read instruction should be given within 1.5 ms after this indicator is turned on. Otherwise, up to 3 ms may be lost in returning to the low-order position.

Branch if Reader Ready. This indicator (d-character 7) is turned on when a tape is loaded and the first line is found by the scanner. The indicator remains on until the operator runs the end of the tape through the transport by pressing the end-of-file key, or until an error occurs that requires operator intervention.

Branch if End of File. This indicator (d-character 8) is turned on when the trailing end of the roll is run through the transport by pressing the end-of-file key. The indicator is reset when another roll is loaded.

Word Marks. Word marks are not affected.

Timing.

No branch or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to a subroutine labeled GO1285 (1644) if the reader-ready indicator is on (Figure G-62).

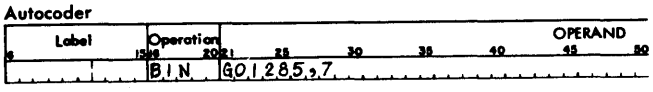


Figure G-62. Branch if Reader-Ready Indicator On

IBM 1285 Timing Considerations

The reading speed of the IBM 1285 Optical Reader depends upon factors which vary with individual applications. The optimum reading speed when using the IBM 1428 character font is given by the following formula:

Throughput

$$\text{(Lines per minute)} = \frac{60{,}000}{1.7W + 1.9S + \frac{47}{L} + 1.4C} \pm 5\%$$

Where:
C = characters per line
L = lines per inch
W = tape width in inches
S = distance from tape edge to first character in inches

With the NCR Optical Character Recognition Font, the optimum reading speed is found with the following formula, using the same symbols for the same variables:

$$\text{(Lines per minute)} = \frac{60{,}000}{1.7W + 1.9S + \frac{47}{L} + 1.55C - 0.5} \pm 5\%$$

Function. This instruction will cause the next line read by a Read Instruction to be read in On-Line Correction Mode, regardless of the setting of the off-line correction on the 1285 Operator Panel. It allows a program decision to change correction mode on re-reading of a line. The switch light should be on (in off-line correction mode) to make use of this instruction.

The next instruction executed is that specified by the I-address, if supplied, or the next instruction in sequence, if no I-address is specified.

Word Marks. Word marks are not affected.

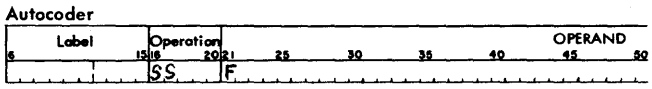
Timing:

Force On-Line Corr.: T = .0333 ms.
Force On-Line Corr. & Br. (without indexing): T = .0666 ms.
Force On-Line Corr. & Br. (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Force On-Line Corr.	NSI	Fbb	Fbb
Force On-Line Corr. & Br. (without indexing):	NSI	BI	blank
Force On-Line Corr. & Br. (with indexing):	NSI	BI	NSI

Example. Causes the next line read by a Read Instruction to be read in On-Line-Correction Mode (Figure G-61.1).



Assembled Instruction: K F

● Figure G-61.1 Force On-Line-Correction

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. This instruction tests for the IBM 1285 operational conditions specified by the d-character. If the indicator is on, the next instruction is taken from the I-address. If off, the program goes to the next sequential instruction.

d-character	*Indicator*
1	ERROR
2	END-OF-LINE
3	READER-TRANSPORTING
4	REJECT-CHARACTER-IN-LINE
5	HEADER-INFORMATION
6	READY-TO-READ-A-LINE
7	READER-READY
8	END-OF-FILE

Indicators

Branch if Error. This indicator (d-character 1) is turned on if any of the following conditions exist. This indicator remains on until tested.

- A process check occurs in the processing unit during a read operation.
- A skew error occurs during a read operation.
- The scanner is unable to follow a line due to extraneous material on the tape during a read operation.
- A reject display exceeds the time limit.
- A line of header or a full-line of correction data is cancelled (cancel-enter sequence) from the reader.

A test for this indicator should be the first instruction following a READ. Wait loops for Header Information (5) should also include a test for Error (Cancel-Enter).

Branch if End of Line. This indicator (d-character 2) is turned on after the last character of a line is transferred to the processing unit and the reader senses the left edge of the tape. If this indicator is not on when a read operation is completed, an error condition may exist, depending on the program and the tape format. This indicator is turned off by the go-to-the-next-line instruction, or by re-reading the same line.

Branch if Reader Transporting. This indicator (d-character 3) is turned on when the transport mechanism is started to bring a new segment of tape over the scan window. It is turned off when the transport mechanism is stopped. Note that a minimum of 1 millisecond elapses between execution of the GO-TO-NEXT-LINE instruction and the beginning of the transport operation. Also, this indicator is turned off as soon as the transport stops, leaving about 6 milliseconds until the ready-to-read-a-line indicator is turned on.

This indicator may be used to determine if sufficient time is available to execute other instructions. This indicator is also on when displaying a line.

Branch if Reject Character in Line. This indicator (d-character 4) is turned on if the last line read (Line Reading only) contains at least one reject symbol (@). This indicator is turned off by Go-to-Next-Line or by Re-read instruction (either line read or full line correction).

Note: Items that can lower the optimum calculated speed are: rescans, branch-1 errors, rereads of wrong length lines, and line marking. Both equations pertain to calculations for constant line length and uniform line spacing only.

Using the IBM 1428 character font, two examples of throughput are given.

C = 10 characters per line
L = 4 lines per inch
W = 2 inches
S = 0.2 inch

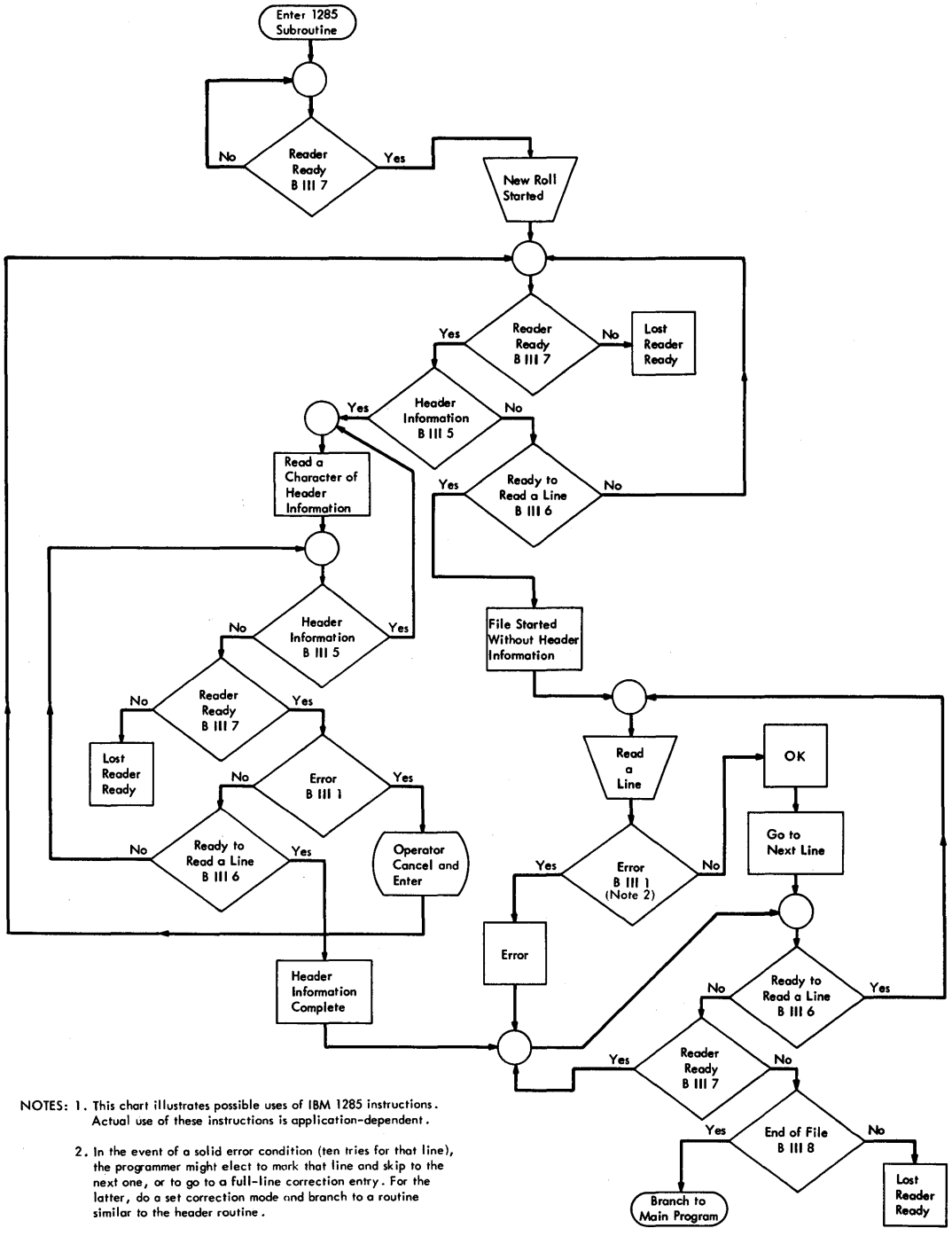
Throughput = 2030 lines per minute

C = 4 characters per line
L = 4 lines per inch
W = 1 5/16 inches
S = 0.1 inch

Throughput = 3035 lines per minute

IBM 1285 Programming Techniques

Figure G-63 shows how some of the IBM 1285 instructions might be used. Figure G-63 is only a guide: the application to be performed will dictate the actual use of the instructions.



NOTES: 1. This chart illustrates possible uses of IBM 1285 instructions. Actual use of these instructions is application-dependent.

2. In the event of a solid error condition (ten tries for that line), the programmer might elect to mark that line and skip to the next one, or to go to a full-line correction entry. For the latter, do a set correction mode and branch to a routine similar to the header routine.

Figure G-63. IBM 1285 Programming Schematic

Branch if Header Information. This indicator (d-character 5) is turned on when a character of header information or full-line correction is entered from the keyboard. It is reset for each character read by the program. As long as this indicator is on, the reader will not respond to the start key to begin processing the journal tape. When no more information is to be entered, the start key is pressed, and the ready-to-read-a-line indicator is turned on.

Branch if Reader Ready to Read a Line. This indicator (d-character 6) is turned on when the reader has found a line and is ready to read it. The indicator is turned off by a go to next line instruction, or by loss of the reader-ready condition.

Note: When a line is found, the scanner enters the normalizing mode and continues from character to character in the line until a read instruction is given. At that time, the flying spot sweeps back to the low-order character.

To keep read time to a minimum, a read instruction should be given within 1.5 ms after this indicator is turned on. Otherwise, up to 3 ms may be lost in returning to the low-order position.

Any wait loops testing this indicator should also include a test for Reader Ready (7).

Branch if Reader Ready. This indicator (d-character 7) is turned on when a tape is loaded and the first line is found by the scanner. The indicator remains on until the operator runs the end of the tape through the transport by pressing the end-of-file key, or until an error occurs that requires operator intervention.

Any wait loops testing other indicators should include a test for Reader Ready. If Reader Ready is lost, the End-of-File indicator (8) should be tested.

Branch if End of File. This indicator (d-character 8) is turned on when the trailing end of the roll is run through the transport by pressing the end-of-file key. The indicator is reset when another roll is loaded.

The test for this indicator should be executed immediately after the loss of the Reader Ready (7) indicator.

Word Marks. Word marks are not affected.

Timing.

No branch or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Branch to a subroutine labeled GO1285 (1644) if the reader-ready indicator is on (Figure G-62).

Autocoder

Label	Operation	OPERAND
	B I N	G0,1,2,8,5,,7

Assembled Instruction: B W44 7

Figure G-62. Branch if Reader-Ready Indicator On

IBM 1285 Timing Considerations

The reading speed of the IBM 1285 Optical Reader depends upon factors which vary with individual applications. The optimum reading speed when using the IBM 1428 character font is given by the following formula:

Throughput

$$(\text{Lines per minute}) = \frac{60{,}000}{1.7W + 1.9S + \frac{47}{L} + 1.4C} \pm 5\%$$

Where:
- C = characters per line
- L = lines per inch
- W = tape width in inches
- S = distance from tape edge to first character in inches

With the NCR Optical Character Recognition Font, the optimum reading speed is found with the following formula, using the same symbols for the same variables:

$$(\text{Lines per minute}) = \frac{60{,}000}{1.7W + 1.9S + \frac{47}{L} + 1.55C - 0.5} \pm 5\%$$

Note: Items that can lower the optimum calculated speed are: rescans, branch-1 errors, rereads of wrong length lines, and line marking. Both equations pertain to calculations for constant line length and uniform line spacing only.

Using the IBM 1428 character font, two examples of throughput are given.

C = 10 characters per line
L = 4 lines per inch
W = 2 inches
S = 0.2 inch

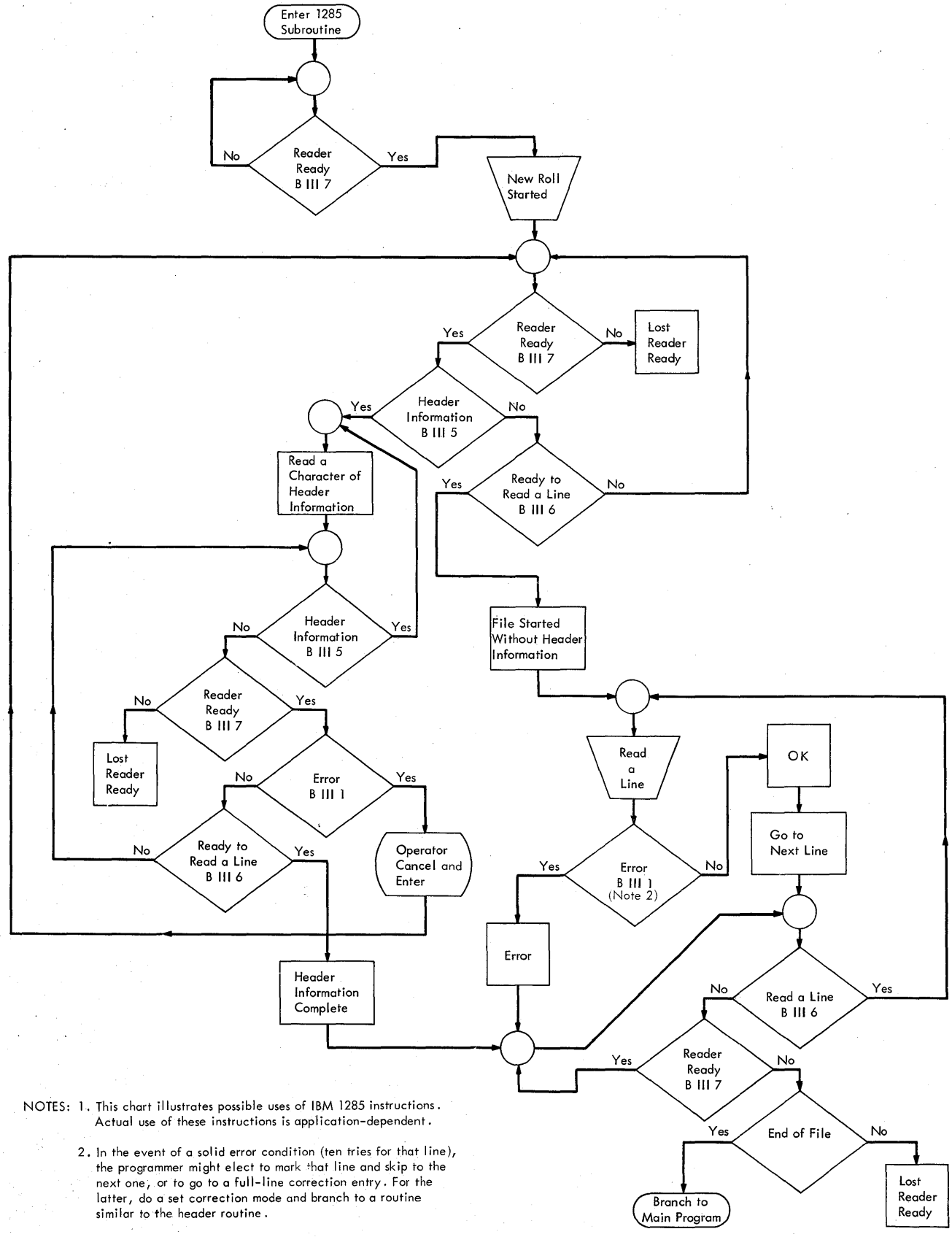
Throughput = 2030 lines per minute

C = 4 characters per line
L = 4 lines per inch
W = 1 5/16 inches
S = 0.1 inch

Throughput = 3035 lines per minute

IBM 1285 Programming Techniques

Figure G-63 shows how some of the IBM 1285 instructions might be used. Figure G-63 is only a guide: the application to be performed will dictate the actual use of the instructions.



NOTES: 1. This chart illustrates possible uses of IBM 1285 instructions. Actual use of these instructions is application-dependent.

2. In the event of a solid error condition (ten tries for that line), the programmer might elect to mark that line and skip to the next one, or to go to a full-line correction entry. For the latter, do a set correction mode and branch to a routine similar to the header routine.

Figure G-63. IBM 1285 Programming Schematic

IBM 1026 Transmission Control Unit

The IBM 1026 Transmission Control Unit (Figure G-64) provides an economical means of communicating numeric, alphabetic, and special-character data between the IBM 1440 system and any of the following sources, via a half-duplex multipoint communications line (publication form numbers appear in parentheses):

- IBM 1030 Data Collection System (A24-3018)
- IBM 1050 Data Communication System (A24-3020)
- IBM 1060 Data Communication System (A24-3034)
- IBM 1070 Process Communication System (A24-5780).

As many as four 1026 units can be attached to the IBM 1440 system. Information can be transmitted on a half-duplex line in either direction, but in only one direction at a time. The 1026 directs and regulates the flow of data and provides compatibility among terminals and the central processing unit. Refer to IBM *1026 Transmission Control Unit,* Form A24-3244.

Mode Switch

This 3-position toggle switch has the following functions:

1. The bottom reset position provides a means of resetting the 1026 controls after system power has been turned on.
2. The center 1026 position allows normal 1026 operations.
3. The top 1447 position allows the processing unit to address the 1447 console directly.

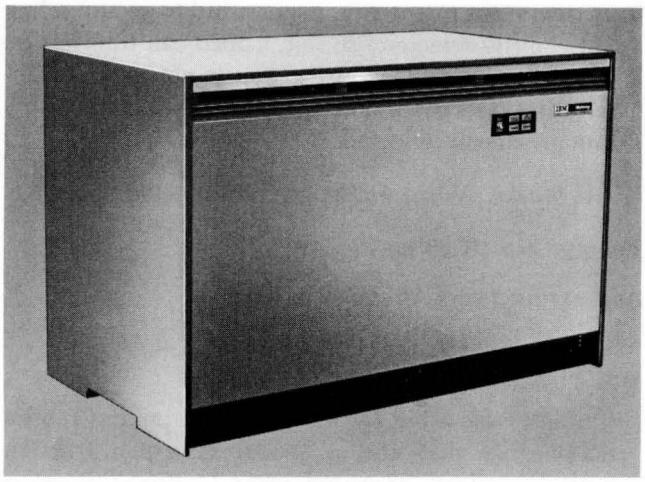


Figure G-64. IBM 1026 Transmission Control Unit

IBM 1026 Programming Considerations

When the IBM 1026 is attached to the IBM 1440 system, the indexing and store-address register features are required if IOCS is to be used. Without IOCS, programming for the 1026 is made easier and more flexible by using these features.

Once a particular line has been selected, it need not be reselected before executing instructions pertaining to that line, unless an intervening line-select operation is executed. The *Select Request-Service Indicator* and *Select Digital Time Unit* instructions, as well as *Select Line* instructions, deselect the line previously selected.

Some special-character d-modifiers used in IBM 1026 instructions are not printable graphics for certain IBM 1440 system configurations. Confusion between unconditional and conditional branches can be reduced by identifying such conditional branches in the comments area of the autocoder coding sheet.

Instructions requiring a U op code with a d-character op code must be coded in actual machine language.

IBM 1026 Programming Example

Although actual IBM 1026 programming depends upon the application, the following programming examples must be considered.

To write to the 1447 console:

```
WCONSO  SBR   EXITW+3    Save return to main line.
U6SEL     U6             Select 1026 channel.
        BIN   U6SEL,□      Branch if buffer busy.
        WCP   TYPEW        Write to console printer.
EXITW   B     0            Branch to main line.
TYPEW   DA    1X100, G     Typeout I/O storage area.
                1, 1
```

To read from 1447 console:

```
RCONSO  SBR   EXITR+3    Save return address to
                           main line.
W1        U6             Select 1026 channel.
        WCP   POLST        Poll the console.
        BIN   W1,*         Branch if error.
W2        U6             Select 1026 channel.
        BIN   R1, Q        Branch if service request.
        B     W2           Branch if no service
                           request.
R1      RCP   TYPER        Read from console.
EXITR   B     0            Return to main line.
TYPER   DA    1X200, G     Type-in I/O area.
                1, 1
POLST   DCW   @A0@, G      Console polling list.
```

The polling count is stored in the first position of the read-in I/O area.

IBM 1026 Instructions

Instructions applying to the IBM 1026 cannot be successfully chained.

Select Request-Service Indicator

Instruction Format.

Mnemonic	*Op Code*	*d-character*
None	U	#

Function. This instruction causes the master request-service (Q) indicator to be selected. This instruction deselects the 1026 line previously selected.

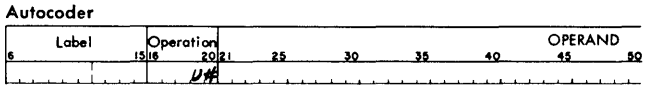
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	#bb	#bb

Example. Select the master request-service indicator so that it can be tested. This instruction does not *test* the indicator. (Figure G-65).



Assembled Instruction: U #

Figure G-65. Select Request-Service Indicator

Select Line

Instruction Format.

Operation	*Mnemonic*	*Op Code*	*d-character*
Select Line 1	None	U	6
Select Line 2	None	U	7
Select Line 3	None	U	8
Select Line 4	None	U	9

Function. These instructions cause selection of the indicators and data channels for a particular IBM 1026 for later examination or data transfer.

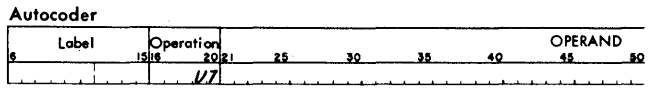
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

Example. Select the second 1026 so that indicator tests and data transfers can be handled relative to line 2 (Figure G-66).



Assembled Instruction: U 7

Figure G-66. Select the IBM 1026 on Line 2

Select Digital Time Unit

Instruction Format.

Mnemonic	*Op Code*	*d-character*
None	U	0

Function. This instruction causes selection of the IBM 1032 Digital Time Unit so that clock data can be transferred into core storage by a subsequent *Read Console Printer* (RCP) instruction. This instruction deselects the 1026 line previously selected. The desired 1026 line must be reselected before 1026 operations can proceed.

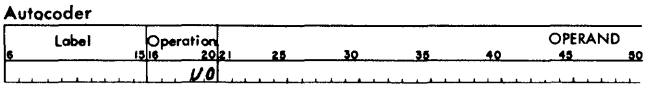
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	0bb	0bb

Example. Select the digital time unit as the data-transfer source for the next read-console-printer instruction (Figure G-67).



Assembled Instruction: U 0

Figure G-67. Select IBM 1032 Digital Time Unit

Bid Operation

Instruction Format.

Mnemonic	*Op Code*	*d-character*
None	U	5

Function. This instruction issues a bid to secure a line (previously selected) so that a 1026 can transmit on the polling line. This instruction is not itself a line-selection operation, and does not deselect the 1026 line previously selected.

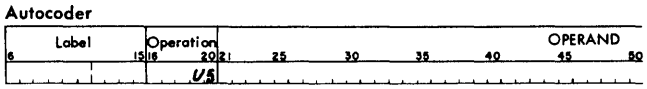
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	5bb	5bb

Example. Issue a bid for line 2, so that polling can be initiated on that line. A SELECT LINE 2 instruction must have been executed prior to this instruction (Figure G-68).



Assembled Instruction: U 5

Figure G-68. Issue IBM 1026 Line Bid

Read from 1026

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RCP	M	%T0	BBB	R
RCPW	L	%T0	BBB	R

Function. This instruction causes data currently held in the selected 1026 buffer (on the last line selected) to be read into core storage, beginning at the address specified by the B-address.

Data transfer continues until:

1. An EOB is sensed from the line, or
2. A group mark with a word mark is sensed in core storage (an error condition), or
3. A group mark with a word mark is written into the 1026 buffer from the IBM 1447 Console.

Records can contain up to 208 characters.

Word Marks. Word marks are not affected when in move mode (M op code).

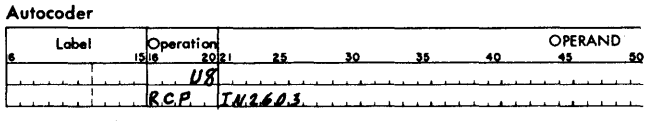
Word marks generated from the 1447 console are transferred to core storage when operating in load mode (L op code).

Timing. T = .0999 ms + maximum of 4.2 ms for first character + .02 ms for each character transferred.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%30	$B + L_B + 1$

Example. Read from 1026 on line 3, and place the data in core storage beginning at the location labeled IN2603 (0632). Line 3 must first have been selected (Figure G-69).



Assembled Instruction: U 8
M %T0 632 R

Figure G-69. Read from IBM 1026

Write to 1026

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WCP	M	%T0	BBB	W
WCPW	L	%T0	BBB	W

Function. This instruction causes information such as a polling list or data to be transferred from core storage to the 1026 attached to the most recently selected line. The transfer stops when:

- An *EOB* (in the message) is transferred to the 1026 buffer to go to the line.
- A group mark with a word mark is transferred to the 1026 buffer.
- After 210 characters have been written.

Word Marks. Word marks are not affected when operating in move mode (M op code).

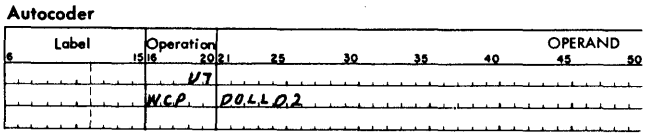
When operating in load mode (L op code), word marks are transferred to the 1447 console. Word marks are ignored when transferred to the line.

Timing. T = .0999 ms + 4.6 ms (delay-line constant) + .02 ms for each character transferred.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%30	$B + L_B + 1$

Example. Write the contents of the core-storage area labeled POLL02 (11,601) to line 2 (Figure G-70).



Assembled Instruction: U 7
M %T0 F0J W

Figure G-70. Write to IBM 1026

Enable Interrupt and Enable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		>
SSB	K	III	>

Function. This instruction (d-character bit-configuration 8-4-2) resets the interrupt-routine interlock. The program continues with the next sequential instruction unless an I-address is present. The inter-

rupt interlock is set either automatically upon entering the interrupt or when the DISABLE INTERRUPT instruction is executed. Refer to *Example of Interrupt Subroutine* section of *IBM 1026 Transmission Control Unit,* Form A24-3244. Figure G-71 lists the interruptible instructions of the IBM 1440 system.

Word Marks. Word marks are not affected.

Timing.

Enable Interrupt: T = .0333 ms.

Enable Interrupt and Branch without indexing: T = .0666 ms

Enable Interrupt and Branch with indexing: T = .0777 ms.

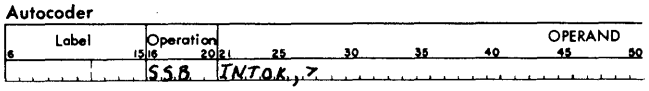
DESCRIPTION	OPERATIONS			
	Interruptable		Non-Interruptable	
	Op Code	Length	Op Code	Length
Add	A	7	A	1,4
Branch	B	5,8	B	1,4
Branch if Bit Equal	W	8	W	1,4
Branch if Word Mark or Zone	V	8	V	1,4
Compare	C	7	C	1,4
Control Carriage	F	5	F	2
Control Unit			U	2,5
Clear	/	7	/	1,4
Clear Word Mark	□	7	□	1,4
Divide	%	7	%	4
Edit	E	7	E	4
Expand Compressed Tape	X	7	X	1,4
Halt	•	5,7	•	1,2,4
I/O Operations			All	All
Load (Excluding I/O)	L	7	L	1,4
Modify Address	#	7	#	1,4
Move (Excluding I/O)	M	7	M	1,4
Move Digit	D	7	D	1,4
Move Record	P	7,8	P	1,4
Move and Suppress Zeros	Z	7	Z	4
Move Zone	Y	7	Y	1,4
Multiply	@	7	@	4
No Operation	N	5,7,8	N	1,2,4
Scan			O	4
Select Stacker	K	5	K	2
Set Word Mark	,	7	,	1,4
Store A-Register	Q	7	Q	1,4
Store B-Register	H	7	H	1,4
Subtract	S	7	S	1,4
Translate	T	7,8	T	1,4
Zero and Add	?	7	?	1,4
Zero and Subtract	!	7	!	1,4

Figure G-71. Interruptible Instructions

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg*
Enable Interrupt	NSI	>bb	>bb
Enable Interrupt and Branch (without indexing):	NSI	BI	blank
Enable Interrupt and Branch (with indexing):	NSI	BI	NSI

Example. Enable the system to accept interruptions, and branch to the subroutine labeled INTOK (15,407); Figure G-72.



Assembled Instruction: K DOG >

Figure G-72. Enable Interrupt and Branch

Disable Interrupt and Disable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		<
SSB	K	III	<

Function. This instruction (d-character B-A-8-4-2) sets the interrupt interlock and prevents the system from honoring any interrupt requests. The interrupt interlock is reset by the ENABLE INTERRUPT instruction. If an I-address is present, a branch occurs.

Word Marks.. Word marks are not affected.

Timing.

Disable Interrupt: T = .0333 ms.

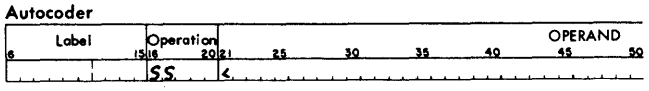
Disable Interrupt and Branch without indexing: T = .0666 ms.

Disable Interrupt and Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Disable Interrupt:	NSI	<bb	<bb
Disable Interrupt and Branch (without indexing):	NSI	BI	blank
Disable Interrupt and Branch (with indexing):	NSI	BI	NSI

Example. Prevent the system from honoring any interrupt requests (Figure G-73).



Assembled Instruction: K <

Figure G-73. Disable Interrupt

Address Mode				
Status	Q	⌑	*	Line Condition
Transmit EOB†	Off	Off	Off	Good transmission to terminal or console.
	Off	Off	On	This combination should not occur in transmit mode.
Transmit	Off	On	Off	Program initiated operations are in process (addressing or transmitting).
	Off	On	On	This combination should not occur in transmit mode.
Control EOB Check†	On	Off	Off	1. Answerback not received by 1026 in response to text. 2. Loss of communication line detected during transmission.
Transmit Control EOB†	On	Off	On	Ⓝ received in response to addressing.
Transmit Control EOB Check†	On	On	Off	1. Addressing answerback time-out (no response received from terminal). 2. Loss of communication line detected during addressing.
Transmit EOB Check†	On	On	On	1. Transmission (to terminal or console) has VRC check which was detected by 1026. 2. Text answerback from terminal is other than Ⓨ.

Polling Mode				
Status	Q	⌑	*	Line Condition
Idle†	Off	Off	Off	1. End of polling list. 2. Text answerback has been sent to terminal without multiblock feature. 3. Line has been idled. 4. Disconnect operation has been completed. 5. Connection not established when dialing on an automatic calling line.
Buffer Check†	Off	Off	On	Input message from terminal or console exceeds capacity of buffer or console.
Receive Control	Off	On	Off	Program initiated operations are in process (polling, bidding, receiving, sending of answerbacks).
	Off	On	On	This combination should not occur in polling mode.
Control EOB Check†	On	Off	Off	1. Text time-out (22.6- 23 seconds). 2. Successful bid for line due to polling time-out (522-533 milliseconds). 3. Loss of communications line during polling or receiving.
Control EOB†	On	Off	On	1. Ⓒ received by 1026 in response to text answerback (Ⓨ or Ⓝ). 2. Successful bid for line when response to polling is Ⓝ 3. Connection established by dialing on automatic calling line.
Receive EOB†	On	On	Off	Good message received from terminal or console.
Receive Check EOB†	On	On	On	Message has VRC or LRC check (from terminal or console).

†Status causes request for interrupt

Figure G-74. IBM 1026 Status-Indicator Settings

Branch if Request-Service Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	Q

Function. This instruction causes a branch to the specified I-address if the selected line (U6, U7, U8, U9, or U0) has its request-service (Q) indicator on, or if the master Q indicator is selected (U#) and on. The master Q indicator is on if any line's Q indicator is on, or if any line's buffer-busy (⌑) indicator is off. The request-service indicator is not reset by this instruction. Figure G-74 is a summary of the 1026 condition indicators.

The IBM *1026 Programming Considerations* section illustrates the use of IBM 1026 instructions.

Timing.

No branch, or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	Qbb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the request-service indicator to determine if a request for service is pending for the selected 1026 line, or if any request-service indicator is on. If the test is positive, branch to a subroutine labeled ADDRES (0776); Figure G-75.

Autocoder

Label	Operation	OPERAND
	B.I.N.	ADDRES,Q

Assembled Instruction: B 776 Q

Figure G-75. Branch if Request-Service Indicator On

Branch if Buffer-Busy Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	⌑

Function. This instruction causes a branch to the specified I-address if the selected line's buffer-busy (⌑) indicator is on. The indicator is not reset by this instruction. It is reset when the desired line is again selected, and the particular selected 1026 buffer is

available. Refer to Figure G-74 for a summary of IBM 1026 condition indicators, and to the IBM *1026 Programming Considerations* section.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	□bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the availability of the 1026 buffer on line 4. If the buffer is busy, branch to a routine labeled BY264 (0199). Line 4 must first be selected (Figure G-76).

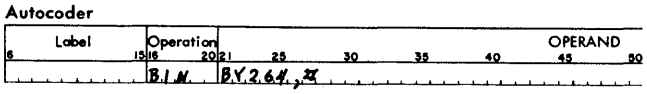


Figure G-76. Branch if Buffer-Busy Indicator On

Branch if Inquiry-Clear Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	*

Function. This instruction causes a branch to the I-address specified if the inquiry-clear (*) indicator for the selected line is on. This indicator is not reset by the branch instruction. See Figure G-74 for 1026 indicator conditions, and the IBM *1026 Programming Considerations* section.

Word Marks. Word marks are not affected.

Timing

No branch, or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	*bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. If the inquiry-clear indicator for line 1 is on, branch to a routine labeled IC2601 (12,635). If the indicator is not on, execute the next sequential instruction. The desired line must first have been selected (Figure G-77).

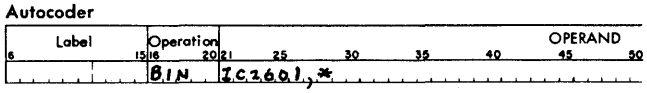


Figure G-77. Branch if Inquiry-Clear Indicator On

Branch if Time-Emitter Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	]

Function. The Time Emitter special feature sets an indicator every 3.6 seconds (1/1000 of an hour). This instruction (d-character B-8-4-1) causes a branch to the I-address if 3.6 seconds has elapsed since the indicator was last tested. This instruction resets the indicator, and another 3.6-second time interval is begun. When the interrupt feature is installed, the time emitter automatically causes an INTERRUPT every 3.6 seconds.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing:
T = .0666 ms.

Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	]bb
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Test the time-emitter indicator, and branch to a routine labeled MDHOWE (9999) if 3.6 seconds have elapsed since the indicator was last tested (Figure G-78).

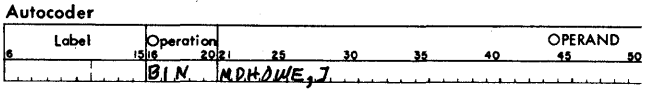


Figure G-78. Branch if Time-Emitter Indicator On

Only those IBM 1440 Data Processing System special features that require program instructions are listed here. For a complete list of special features for this system, refer to the IBM *1440 System Summary*, Form A24-3006.

Bit Test

This feature is a BRANCH instruction that compares the character located at the B-address with the d-character, bit-by-bit. If any bit in the character located at the B-address matches any bit in the d-character, the program branches to the specified I-address. (WM and C-bits are not compared.)

Branch if Bit Equal

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*B-address*	*d-character*
BBE	W	III	BBB	d

Function. The d-character can contain any character or any combination of bits (BA 8421) that can exist in a single position of core storage. If the character at the B-address contains any bit that matches any bit in the d-character, the program branches to the I-address. Otherwise, the program continues normally.

Word Marks. Word marks are not affected.

Timing.
No Branch: T = .1110 ms.
Branch (without indexing): T = .1110 ms.
Branch (with indexing): T = .1221 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	B-1
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Examine the storage location labeled UNPOS (0759) for a match in the d-character bit configuration. The d-character is a 9 (8- and 1-bits). Therefore, if the character contains either an 8- or 1-bit, the program branches to BITEST (0985), Figure H-1.

Autocoder

Label	Operation	OPERAND
	BBE	BITEST,UNPOS,9

Assembled Instruction: W 985 759 9

Figure H-1. Branch If Bit Equal

Direct Data Channel

This feature provides for the attachment of the 1440 to a 1401, to another 1440, or to a 1460 system. With the direct data channel special feature, the two processing systems are cable-connected through the serial I/O adapter feature on each system. When the direct data channel feature is in use, no other input/output unit can use the serial I/O adapter feature on either system.

The cable length between the two systems can be any length up to a maximum of 100 feet.

Data transmission takes place serially by character and parallel by bit (WM BA8421 plus a parity bit). The type of data transmission operation that can be performed is varied and at the discretion of the user. Depending on the program written, both systems can send and receive data, or one system can send data only while the other system can receive data only. To permit this flexibility, the direct data channel feature makes use of three types of instructions:

1. SIGNAL CONTROL instructions
2. BRANCH instructions
3. MOVE and LOAD instructions.

Signal Control Instructions

The SIGNAL CONTROL instructions are used by one processing system to

1. inform the other processing system that it wants to perform a particular operation, or
2. actually perform a particular function in the other system.

Signal control instructions applying to the direct data channel feature cannot be successfully chained.

Read Request

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	C

Function. This instruction informs the other system that the system initiating this instruction wants to read (receive) data from the other system. This condition is tested for in the other system with its B (III) 3 instruction.

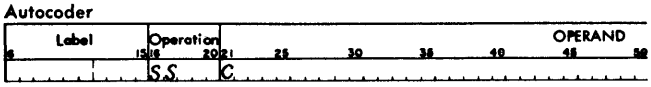
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Cbb	Cbb

Example. The system executing this instruction signals the other system that it wants to receive data from the other system (Figure H-2).



Assembled Instruction: K C

Figure H-2. Read Request

Read Request and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	C

Function. The READ REQUEST AND BRANCH instruction functions exactly like the READ REQUEST instruction, except that the address of the next instruction to be executed is specified by the I-address.

Word Marks. Word marks are not affected.

Timing.
Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Write Request

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	D

Function. This instruction informs the other system that the system initiating this instruction wants to send (write) data to the other system. This condition is tested for in the other system with its B (III) 4 instruction.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Dbb	Dbb

Example. The system executing the instruction signals the other system that it wants to send data to the other system (Figure H-3).

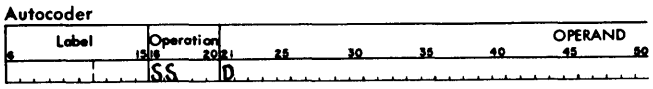


Figure H-3. Write Request

Write Request and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	D

Function. The WRITE REQUEST AND BRANCH instruction functions exactly like the WRITE REQUEST instruction, except that the address of the next instruction to be executed is specified by the I-address.

Word Marks. Word marks are not affected.

Timing.
Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Reset

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	E

If one of the systems is a 1401, this instruction must be given prior to the executing each read- or write-data instruction. This reset instruction must be executed by the 1401 every time the 1440 system is started or restarted. Because the 1401 start-reset key does not include this function, this resets the end-of-transmission circuitry in the 1440 system.

This reset instruction must be included at the start of every direct-data channel program.

Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Ebb	Ebb

Example. Reset the end-of-transmission circuitry in the other system (Figure H-4).

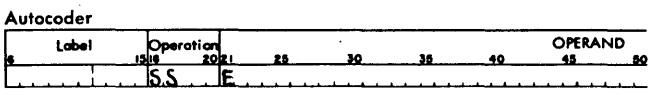


Figure H-4. Reset

Reset and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	E

Function. The RESET AND BRANCH instruction functions exactly like the RESET instruction, except that the address of the next instruction to be executed is specified by the I-address.

Word Marks. Word marks are not affected.

Timing.
Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Branch Instructions

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. This instruction and its associated d-characters are used by the system initiating these instructions to check for various conditions on the other system. When a tested condition is present, the program branches to the previously written subroutine. The BRANCH IF INDICATOR ON instruction, the condition it tests for, and other associated information is shown in Figure H-5.

Word Marks. Word marks are not affected.

Timing.
No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Conditions in System Initiating Line/Signal (System A)	Line/Signal		System B, Testing Conditions in System A with Branch-If-Indicator-On Instructions	
	Sent	Reset By	Branch Instruction *	Indicator Reset
Process Check due to detection of Transmission Error		Start Reset Key	B (III) 1	By executing the Branch Instruction in System B, or by pressing Start Reset Key in System A; both alternatives after pressing the Check Reset Key in System A.
End of Transmission. A GMWM was reached in the System A I/O-area during the previous data transfer. (The I/O-area in System A was smaller or equal in size to the System B).	I/O Disconnect	1401, 1440, 1460 executing a KE instruction in the other system. 1440, 1460 Start Reset Key	B (III) 2	By executing a KE Instruction in System B or by pressing the Start Reset Key in System A, if System A is a 1440 or 1460 System. Note: This indicator must be off in both systems before initiating any data transfer.
A Read Request Instruction (KC) has been executed in System A.	Read Request	Start Reset Key	B (III) 3	By executing a Write Data (with or without word marks) instruction, or pressing the Start Reset Key in System A.
A Write Request Instruction (KD) has been executed by System A.	Write Request	Start Reset Key	B (III) 4	By executing a Read Data (with or without word marks) instruction, or pressing the Start Reset Key in System A.
A Write Data Instruction is being executed in System A.		Start Reset Key	B (III) 6	When System A ends the Write operation. (This is done when System B has executed a read instruction or by pressing the Start Reset Key in System A, or if the Indicator 2 was not reset in System A.)
A Read Data Instruction is being executed in System A.		Start Reset Key	B (III) 7	When System A ends the read operation. (This is done when System B has executed a write operation or by pressing the Start Reset Key in System A, or if the Indicator 2 was not reset in System A.)
System A stopped (Stop key pressed, STOP Instruction, error stop, etc.).		Start Reset Key	B (III) 8	When System A starts.

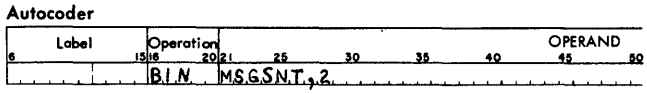
* The d-character must be in the operand field when using a BIN mnemonic.

Figure H-5. Branch If Indicator On Instruction Summary Direct Data Channel Feature

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test for end of transmission by other system. If other system did signal an end-of-transmission, branch to MSGSNT (0843), Figure H-6.



Assembled Instruction: B 843 2

Figure H-6. Branch If Indicator On

Move and Load Instructions

The MOVE or LOAD instruction [M or L (%H1) (BBB) R or W] is used by the systems to transmit or receive the data in either the move mode or the load mode. The parts of the instruction and their uses are:

M or L. The M or L operation code specifies whether the data transmission will be performed in the *move* mode or *load* mode. If the *move* mode is specified, up to 7 bits per character (CBA8421) are involved in the data transmission. If the *load* mode is specified, up to 8 bits per character (WM CBA8421) are involved in the data transmission. The same mode must be used by both systems for any one particular data transmission. Word marks would be lost if the message transmission were in the load mode, but the message reception were in the move mode.

%H1. The A-address (%H1) specifies that the direct data channel feature is used in performing this instruction.

BBB. The B-address specifies the high-order position of the message in core-storage area involved in the data transmission.

R or W. A d-character of R specifies a read operation. This d-character is used when the other system is sending the data. A d-character of W specifies a write operation. This d-character is used when the other system is receiving the data.

Instructions applying to the direct data channel feature cannot be successfully chained.

Read Data

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%H1	BBB	R

Function. This instruction causes the data sent from the other system to read into core storage, beginning at the core-storage location specified in the instruction.

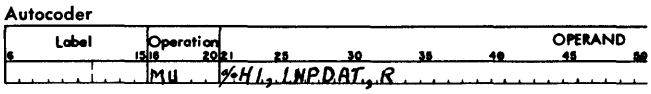
Word Marks. Word marks are not stored when operating in the *move* mode (M operation code).

Timing. T = .0999 ms + transmission and start time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + message length + 1

Example. Read data from the other system and place it in core storage, beginning at location 0633 (area is labeled INPDAT), Figure H-7.



Assembled Instruction: M %H1 633 R

Figure H-7. Read Data

Read Data with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%H1	BBB	R

Function. This instruction is similar to the READ DATA instruction except that word marks in the record area of core storage are removed, and word marks sent with the other data are written in core storage.

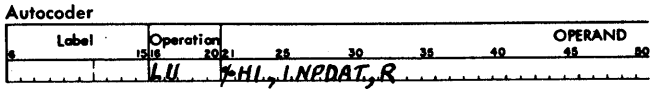
Word Marks. Word marks transmitted from other systems are written in core storage.

Timing. T = .0999 ms + transmission and start time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + message length + 1

Example. Read data from the other system, with its associated word marks, and place it in core storage, beginning at location 0633 (area is labeled INPDAT), Figure H-8.



Assembled Instruction: L %H1 633 R

Figure H-8. Read Data with Word Marks

Write Data

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%H1	BBB	W

Function. This instruction causes data to be sent to the other system from core storage, beginning at the core-storage location specified in the instruction.

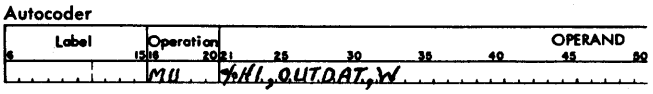
Word Marks. Word marks are not sent to the other system when operating in the *move* mode (M operation code).

Timing. T = .0999 ms + transmission and start time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + message length + 1

Example. Send data to the other system from the core-storage area labeled OUTDAT (first position of the data located in 0633), Figure H-9.



Assembled Instruction: M %H1 633 W

Figure H-9. Write Data

Write Data with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%H1	BBB	W

Function. This instruction is similar to the WRITE DATA instruction except that word marks in the output area of core storage are transmitted with the associated data.

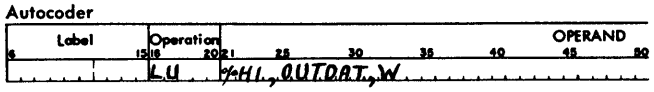
Word Marks. Word marks are sent to the other system.

Timing. T = .0999 ms + transmission and start time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%81	B + message length + 1

Example. Send data to the other system, with its associated word marks, from the core-storage area labeled OUTDAT (first position of the data located in 0633); Figure H-10.



Assembled Instruction: L %H1 633 W

Figure H-10. Write Data with Word Marks

Instruction Utilization in the Program

With the instructions just described, the specific type of system-to-system data transmission can be set up. The type of operation performed is at the discretion of the user because the operation is completely programmed.

Each system has its own specifically designed program, using the previously described instructions. Some instructions are used in both programs, while other instructions might appear in only one program, if at all. The instructions used are completely dependent on the specific type of data transmission involved.

1-Way System-to-System Data Transmission

To illustrate one kind of system-to-system data transmission, assume a hypothetical case where one system sends data only, while the other system receives data only. The program procedure illustrated in Figure H-11 is meant only as an example to show the use of the various instructions and should not be considered the optimum procedure for this kind of operation.

Sending-System Operation

1. The sending system enters its system-to-system data transmission program and executes a BRANCH

IF INDICATOR ON instruction, B (III) 8, which checks to see whether the other system is operating.

2. If the other system is stopped for any reason, the program branches into a previously specified subroutine, which may, for example:
   a. permit processing of some other information, or,
   b. stop the system that initiated the instruction, or,
   c. notify the system operator in some way.

3. If the other system is operating, another BRANCH IF INDICATOR ON instruction, B (III) 7, is executed, which checks the other system to see whether it is trying to execute a READ instruction. If the other system is trying to execute a READ instruction, it informs the sending system by setting the indicator tested by a B (III) 7 instruction.

4. When a B (III) 7 instruction results in a branch, the sending system immediately executes a WRITE instruction, M or L (%H1) (BBB) W.

   The actual data transmission occurs between the two systems and continues until one of the systems encounters a preset group mark with a word mark in its core storage. The group mark with a word mark terminates the data transmission operation and sends a termination signal to the other system. (See *Notes* at the end of this data-transmission example for further information.)

5. If any transmission error occurs in the sending system during the data transmission, the sending system stops at the end of the data transmission operation.

6. The system operator must start the system operating again and will either try to send the data again (step 4) or start at the beginning of the subroutine (step 1).

7. If no transmission error occurred in the sending system during the data transmission, the core-storage address contained in the B-address register is stored in a location specified by the STORE B-ADDRESS REGISTER instruction, H (BBB). This information is used later to determine whether the complete message was transmitted.

8. A BRANCH IF INDICATOR ON instruction, B (III) 1 is executed, which checks to see whether any transmission errors occurred in the other system.

9. If any error occurred in the other system, the end-of-transmission circuitry in that system is reset (KE instruction) and the sending system tries to send the data again. The actual data transmission does not start until the operator corrects the error condition in the other (receiving) system and starts that system operating again.

10. If no transmission error occurred, a BRANCH IF INDICATOR ON instruction, B (III) 2, is executed, which checks to see whether the other system ended the data transmission.

11. If the other system did not end the data transmission, it means that the entire message was transmitted. The subroutine ends, and the system returns to its main program.

12. If the other system did end the data transmission, a check must be made to see whether the entire message was transmitted. One method that could be used is to compare the address stored in step 7 with the address known to be the last core-storage address in the sending system data area.

9. If the two addresses do not compare, the end-of-transmission circuitry in the other system is reset, and the sending system tries to send the data again (step 4) because the receiving system did not receive the complete message.

13. If the two addresses do compare, the entire message was transmitted. The end-of-transmission circuitry in the other system is reset.

11. The subroutine ends, and the system returns to its main program.

Or, if the other system is not trying to execute a read instruction:

3. A BRANCH IF INDICATOR ON instruction, B (III) 7, is executed, which checks to see whether the other system is trying to execute a READ instruction.

14. If the other system is not trying to execute a READ instruction, the sending system informs the other system that the sending system wants to send data by executing a WRITE REQUEST instruction, KD.

Receiving-System Operation

In the receiving system operation being used in this example, three conditions can occur:

1. Sending system wants to send data, or,
2. Sending system is trying to execute a WRITE instruction, or,
3. Sending system does not want to send data and is not trying to execute a WRITE instruction..

Each one of these situations is explained.

15. The receiving system enters its system-to-system data transmission program and executes a BRANCH IF INDICATOR ON instruction, B (III) 8, which checks to see whether the other system is operating.

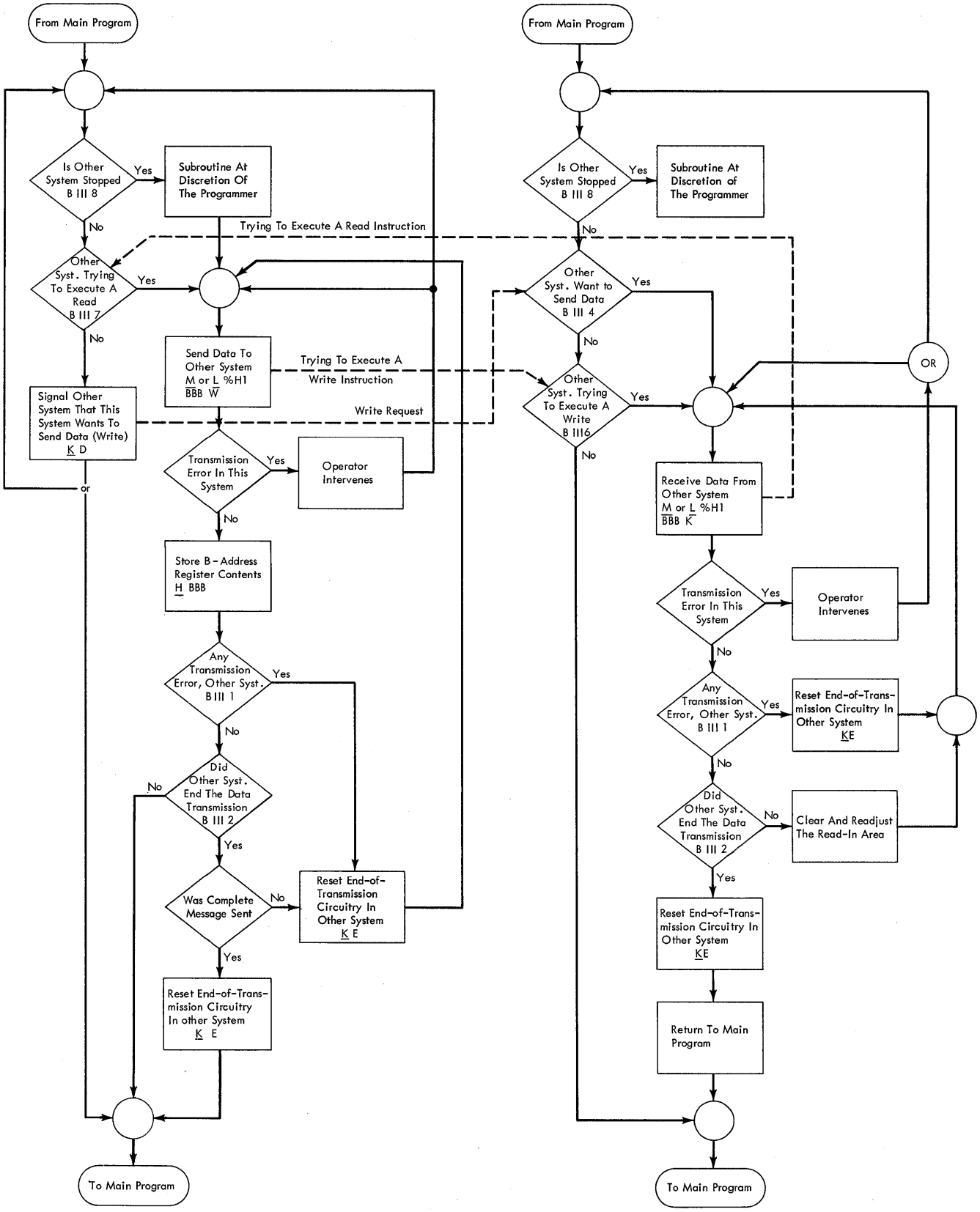


Figure H-11. Programming Example of One-Way System-to-System Data Transmission

16. If the other system is stopped for any reason, the program branches into a previously specified subroutine that may be similar to the subroutine described in step 2.

Condition 1: Sending system wants to send data.

17. If the other system is operating, another BRANCH IF INDICATOR ON instruction, B (III) 4 is executed. This instruction checks to see whether the other system wants to send data. This condition originated in step 14 of the sending system program. If the other system wants to send data, the receiving system immediately tries to execute a READ instruction, M or L (%H1) (BBB) R. The data transmission does not take place immediately, however. In trying to execute a READ instruction, the receiving system signals the sending system that it is trying to execute a READ instruction. After a negligible time interval, the sending system enters its system-to-system data-transmission program and this condition initiates the data transmission previously described in step 4.

18. When the execution of a B (III) 4 instruction results in a branch, the receiving system immediately executes a READ instruction, M or L (%H1) (BBB) R.

    The actual data transmission occurs between the two systems and continues until one of the systems encounters a preset group mark with a word mark in its core storage. The group mark with a word mark terminates the data transmission operation and sends a termination signal to the other system. (See *Notes* at the end of this data-transmission example for further information.

19. If any transmission error occurs in the receiving system during the data transmission, the system stops at the end of the data transmission.

20. The system operator must start the system operating again and will either try to receive the data again (step 18) or start at the beginning of the subroutine (step 15).

21. If no transmission error occurred in the receiving system during the data transmission, a BRANCH IF INDICATOR ON instruction, B (III) 1 is executed, which checks to see whether any transmission errors occurred in the other system.

22. If any error occurred in the other system, the end-of-transmission circuitry in that system is reset and the receiving system tries to receive the data again (step 18).

23. If no transmission error occurred, a BRANCH IF INDICATOR ON instruction, B (III) 2 is executed, which checks to see whether the other system ended the data transmission.

24. If the other system did not end the data transmission, it means that the receiving system read-in area was not large enough to accept the incoming message. A subroutine is executed to readjust the read-in area so that it can accept the entire incoming message, and the receiving system tries to receive the data again (step 18).

At approximately this same time, steps 10, 12 and 9 are being executed in the other system. As previously described in these steps, the other system automatically tries to send the message again. As soon as the read-in area is adjusted, another data transmission operation takes place.

25. If the other system did end the data transmission, the end-of-transmission circuitry in the other system is reset by initiating a RESET (KE) instruction.

6. The subroutine ends and the system returns to its main program.

Condition 2: Sending system trying to execute a WRITE instruction.

27. If the other system is operating, and is not trying to send data, another BRANCH IF INDICATOR ON instruction, B (III) 6 is executed. This instruction checks to see whether the other system is trying to execute a WRITE instruction as a result of an operator intervention or some other condition. If the other system is trying to execute a WRITE instruction, it informs the receiving system by setting the indicator tested by a B (III) 6 instruction. When the execution of a B (III) 6 instruction results in a branch, the program previously described in steps 18-26 is executed.

Condition 3: Sending system does not want to send data and is not trying to execute a WRITE instruction.

15, 17, 27, 26. If the other system is operating, but does not want to send data, and is not trying to execute a WRITE instruction, the subroutine ends and the system returns to its main program.

Notes:

1. The conditions that specify the terminating system are:
   a. The writing system terminates the data transmission (activates its I/O disconnect line to the other system) when the read input area is longer than the write output area.
   b. Both the reading and writing systems terminate the data transmission when the read input area is the same size as the write output area.

c. Both the reading and writing systems terminate the data transmission when the read input area is one core-storage position shorter than the write output area.
d. The reading system terminates the data transmission when the read input area is shorter than the write output area by more than one core-storage position.

2. When either system, or both systems, ends data transmission, one, or both, I/O disconnect lines are activated. As long as this line is active, neither system can execute a direct-data-channel input/output instruction. A WRITE DATA instruction is treated as a NO OPERATION instruction; a READ DATA instruction places a group mark in the first position of the addressed I/O area in core storage, and then ends the operation.

3. Each system must reset the other system's I/O disconnect line. If a B (III) 2 instruction causes a branch, the system in which the branch occurred must execute a KE instruction before the system can execute a direct-data-channel input/output instruction.

2-Way System-to-System Data Transmission

The programming involved in a 2-way system-to-system data transmission operation is, of necessity, more involved than the 1-way system-to-system programming just described. If both systems send and receive data, then the same program routine used by one system can also be used by the other system.

To permit maximum efficiency, each system must test the status of the other system at regular intervals. If the data transmission operations of one system have priority over the other system's operations, the program must include routines that will terminate, or delay, the other system's operations.

If two duplicate programs are used, each program should include a dissimilar timing loop so that the systems do not re-enter their routines together after terminating an operation.

If both programs try to execute READ instructions at the same time, both the systems stop operating because all program execution stops. Each system then waits for the other system to start sending data, but neither one ever starts. This condition can be eliminated by proper programming.

If both programs try to execute WRITE instructions at the same time, the write operations are completed, but neither message is transferred, resulting in the loss of one message in each system. This condition can be eliminated by proper programming.

Direct Seek

This special feature reduces access time on the 1311 disk storage drive (from 400 to 250 ms maximum, and from 250 to 150 ms average) by allowing the access assembly to be positioned directly at a new setting without returning to the home position.

The instruction used for the direct-seek operation is the same as that used with normal seek (M%F0 BBB R). The B-address position of the instruction contains the core-storage address of the high-order position of the 10-digit disk-control field used.

Disk-Control Field

Direct-seek operations use a 6-position sector address from the specified disk-control field (Figure H-12).

The first position of the disk-control field (the alternate code position) contains the disk-drive number (0, 2, 4, 6, or 8). An asterisk cannot be used for this operation.

The next four positions (2-5) contain a signed 4-digit number equal to twice the number of cylinders to be advanced (+) or retracted (−).

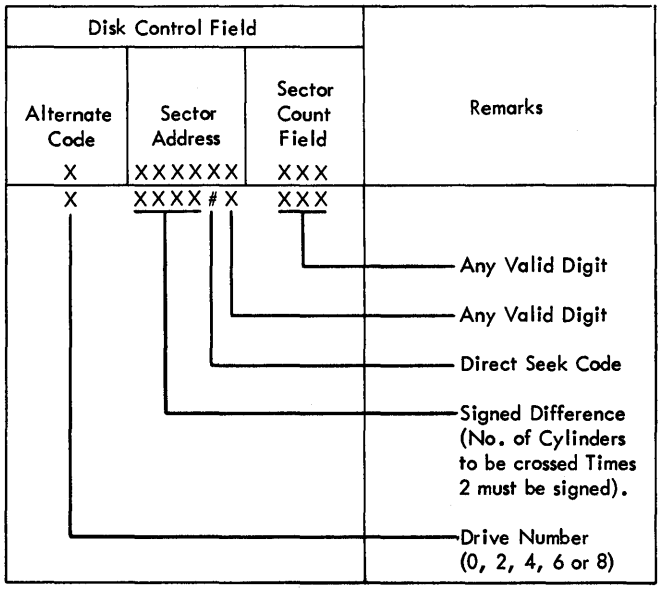


Figure H-12. Disk-Control Field for Direct Seek

CHARACTER	BIT CODE
•	BA 8 21
≠	BA 8421
$	B 8 21
Δ	B 8421
'	A8 21
≠	A8421
√	8421

Figure H-13. Unacceptable Characters in Sixth Position of Disk-Control Field

The sixth position contains a pound (#) sign to indicate a direct-seek operation. Any other character with an 8-2-1 bit combination will be taken to indicate a direct-seek operation and cannot, therefore, be used in the sixth position. See Figure H-13 for a list of these characters.

The signed difference field can be calculated by the method shown in Figure H-14. This method uses the four high-order positions of the disk address at which the access arm is positioned and the four high-order positions of the disk address to be sought. Both fields must be changed to either odd or even (either add one to an even number to make both numbers odd, or, subtract one from an odd number to make both numbers even). The old address is then subtracted from the new address. The result of the subtraction has the correct sign to indicate that the mechanism is to advance (+) or retract (−).

If, for some reason, the address fails to specify the module in the alternate-code position, the direct seek is executed on the master file.

			Both Odd
New Address	004372	0043	0043
		+1	
Old Address	003291	0032	0033
		difference	0010

Signed Difference +0010 (+because increase in address)

Figure H-14. Calculating Signed Difference

Number of Cylinders Traveled	Time In Milliseconds
1	54 Minimum
2	67
3	80
4	90
5	105
6	115
7	130
8	140
9	155
10	165
20	130
30	137
40	154
50	170
60	185
70	202
80	217
90	235
99	248 Maximum

Figure H-15. Cylinder Seek Time with Direct Seek

Direct-Seek Timing

Figure H-15 provides seek times when direct-seek feature is installed on the system.

Expanded Disk-Storage Control

This feature provides the controlling circuitry necessary for the attachment of any IBM 1301 Disk Storage units to the system. (The disk-storage control special feature is a prerequisite.)

For more detailed information on 1301 operation, refer to IBM *1301 Disk Storage, Models 11, 12, 21, and 22*, Form A24-3157.

Expanded Print Edit

The basic operations of the MOVE CHARACTERS AND EDIT instruction can be increased by the expanded print edit feature. With this feature, asterisk protection, floating dollar sign, decimal control, and sign control left, operations can be performed. The zero-suppression code in the control word should be in the position immediately to the left of the decimal, except as required in *Decimal Control.*

Note: Floating dollar sign and asterisk protection or floating dollar sign and decimal control cannot be used in the same edit operation. When asterisk protection and decimal control are combined, and a blank data field is edited, the result is asterisks in all positions to the left of, but not including, the decimal-control position.

Asterisk Protection

When asterisks are to appear to the left of significant digits, the asterisk protection feature is used (Figure H-16). The control word is written with the asterisk immediately to the left of the zero suppression code. Zero-balances can be protected with asterisks by placing control zeros in the right-most position. In this instance, asterisks print in all positions including the decimal position.

Forward Scan:

1. The normal editing process proceeds until the asterisk is sensed.
2. The corresponding digit from the A-field replaces the asterisk (in the output field).
3. The editing process continues normally until the B-field word mark is sensed and removed.

Reverse Scan:

1. Asterisks replace zeros, blanks, and commas, to the left of the first significant digit.
2. The word mark (set during the forward scan) signals the end of editing. It is erased, and the operation is stopped.

Floating-Dollar Sign

This feature causes the insertion of a dollar sign in the position at the left of the first significant digit in an

A-field	00257426
Control word (B-field)	bbb, b*0. bb&CR
Forward scan	002,574.26 CR
Reverse scan	**2,574.26 CR
Results of edit	**2,574.26 CR

Figure H-16. Asterisk Protection

amount field (Figure H-17). The control word is written with the $ immediately to the left of the zero-suppression code.

Note: The control word must be larger than the A-field.

Three scans are necessary to complete this editing operation.

First Forward Scan:

1. The editing proceeds until the $ is sensed.
2. The corresponding digit from the A-field replaces the $ (in the output field).
3. Editing continues until the B-field word mark is sensed and removed.

Reverse Scan:

1. Blanks replace both zeros and commas to the left of the first significant digit.
2. The reverse scan continues until the word mark (set during the first forward scan) signals the start of the second forward scan.

Second Forward Scan:

1. The word mark is erased and the scan continues until the first blank position is sensed. This blank position is replaced by $, and the operation stops.

Sign Control Left

CR or minus symbols can be placed at the left of a negative field, if the sign control left feature is used

A-field	00257426
Control word (B-field)	bbbb, b$0. bb
First forward scan	002,574.26
Reverse scan	bbb 2,574.26
Second forward scan	$2,574.26
Results of edit	$2,574.26

Figure H-17. Floating Dollar Sign

A-field	00378940
Control word (B-field)	CR&bbb, bb0. bb
Forward scan	CRb003, 789.40
Reverse scan	CRbbb3, 789.40
Results of edit	CR 3,789.40

Figure H-18. Sign Control Left

(Figure H-18). The control word is written with the CR or minus symbols in the high-order position.

Forward Scan:

1. The scan proceeds until the zero suppression character in the control field is sensed.
2. The corresponding character from the A-field is placed in this position of the output field.
3. A word mark is automatically inserted in this position in the output field.
4. Editing continues and the CR or minus symbols are undisturbed in their corresponding positions in the output field, only if the sign of the A-field is minus. If the sign is plus, they are blanked.

Reverse Scan:

1. Blanks in the output field replace zeros and commas. The scan continues until the automatically set word mark is sensed.
2. This word mark is erased and the operation ends.

Decimal Control

This feature ensures that decimal points print only when there are significant digits in the A-field (Figure H-19).

Two scans are sufficient to complete this editing operation, *unless* the field contains no significant digits. Then three scans are required.

First Forward Scan:

1. When the zero suppression code (0) is sensed during editing, the corresponding digit from the A-field replaces this position.
2. A word mark is set automatically in this position in the B- (output) field.
3. Editing continues normally until the B-field word mark is sensed and removed.

1. A-field	00000
Control word (B-field)	bbb. b0 ←
First forward scan	000.00 →
Reverse scan	bbb. 00 ←
Second forward scan	bbb
Results of edit	(Blank Field)
2. A-field	29437
Control word (B-field)	bbb. b0 ←
First forward scan	294.37 →
Reverse scan	294.37
Result of edit	294.37
3. A-field	00001
Control word (B-field)	bbb. b0 ←
First forward scan	000.01 →
Reverse scan	bbb. 01
Results of edit	.01

Figure H-19. Decimal Control

Reverse Scan:

1. Blanks in the output field replace zeros and commas until the decimal point is sensed.
2. The decimal point and the digits at its right are unaltered. The automatically-set word mark is erased. If there are no significant digits in the field, the second forward scan is initiated. Otherwise, the edit operation stops.

Second Forward Scan:

1. Blanks replace the zeros at the right of the decimal point and the decimal point itself.
2. The operation stops at the decimal column.

Expanded Serial Input/Output Adapter

A large-scale data processing system (such as the IBM 1410/7010) can transmit data to and from an IBM 1440 Data Processing System. The large-scale system must have a control adapter feature, priority feature, and processing overlap feature installed, while the 1440 system must have the *Expanded Serial I/O Adapter* feature installed. Also, a prerequisite for the 1440 system is the indexing and store-address register feature and the bit-test feature.

Throughout this section, the large-scale data processing system is referred to as the *primary system*; the IBM 1440 Data Processing System is referred to as the *secondary system.*

See Figure G-35 for a list of instructions that can be interrupted.

Primary System Operation

The primary system can initiate four types of operations:

1. Read operation
2. Write operation
3. Control operation
4. Sense operation.

Any of these can cause an interrupt in the secondary system.

Read Operation

When the primary system initiates a read operation, it sends a read command signal to the secondary system. This signal sets the primary read latch ON in the secondary system (latch remains on until the operation ends). When the secondary system program tests the primary read latch and finds it on, the program branches into a subroutine that sends data from the secondary system to the primary system.

Write Operation

When the primary system initiates a write operation, it sends a write command signal to the secondary system. This signal sets the primary write latch ON in the secondary system (latch remains on until the operation ends). When the secondary system program tests the primary write latch and finds it on, the program branches into a subroutine that receives data in the secondary system from the primary system.

Control Operation

When the primary system initiates a control operation, it sends a control command signal to the secondary system. This signal sets the primary control latch ON in the secondary system (latch remains on until the operation ends). When the secondary system program tests the primary control latch and finds it on, the program branches into a subroutine that receives control data in the secondary system from the primary system. The data must be analyzed by the secondary system to determine what it must do under primary system control.

Sense Operation

When the primary system initiates a sense operation, it sends a sense command signal to the secondary system. This signal sets the primary sense latch ON in the secondary system (latch remains on until the operation ends). When the secondary system program tests the primary sense latch and finds it on, the program branches into a subroutine that sends status data from the secondary system to the primary system. The status operation is usually initiated by the primary system when the secondary system initiates an unusual-end signal or an attention signal. This status data sent to the primary system is usually only four characters in length and indicates what condition initiated the unusual-end signal. Some of the conditions that can initiate the signal are:

1. program check
2. message length check
3. no storage available.

Primary System Signals

The primary system communicates with the secondary system by using seven signals:

1. read command
2. write command
3. control command
4. sense command
5. end response
6. operational out line
7. stop.

Figure H-20 shows a summary of the primary system signals.

Read Command

The read command signal is sent to the secondary system by the primary system to initiate a primary system read operation. The read command signal sets ON the primary read latch in the secondary system.

The status of the primary read latch is checked with a BRANCH IF INDICATOR ON instruction — B (III) 3.

Signal	Function
Read-Command Signal	Sent to the secondary system to initiate a primary system-read operation. Turns ON primary read latch, and the latch status is checked with a B (III) 3 instruction.
Write-Command Signal	Sent to the secondary system to initiate a primary system-write operation. Turns ON primary write latch, and the latch status is checked with a B (III) 4 instruction.
Control-Command Signal	Sent to the secondary system to initiate a primary system-control operation. Turns ON primary control latch, and the latch status is checked with a B (III) 7 instruction.
Sense-Command Signal	Sent to the secondary system to initiate a primary system-sense operation. Turns ON primary sense latch, and the latch status is checked with a B (III) 6 instruction.
End-Response Signal	Sent to the secondary system as an acknowledgement signal after receiving an end or unusual end signal. Turns OFF any primary latch in the secondary system that is ON.
Operational Out Line	Tells the secondary system if primary system is operating. Can be program-tested with a B (III) 8 instruction.
Stop Signal	Tells the secondary system that primary system has terminated the data-transmission operation.

Figure H-20. Primary System Signal and Line Condition Summary

Write Command

The write command signal is sent to the secondary system by the primary system to initiate a primary system write operation. The write command signal sets ON the primary write latch in the secondary system.

The status of the primary write latch is checked with a BRANCH IF INDICATOR ON instruction — B (III) 4.

Control Command

The control command signal is sent to the secondary system by the primary system to initiate a primary system control operation. The control command signal sets ON the primary control latch in the secondary system.

The status of the primary control latch is checked with a BRANCH IF INDICATOR ON instruction — B (III) 7.

Sense Command

The sense command signal is sent to the secondary system by the primary system to initiate a primary system sense operation. The sense command signal sets ON the primary sense latch in the secondary system.

The status of the primary sense latch is checked with a BRANCH IF INDICATOR on instruction — B (III) 6.

End Response

The end response signal is sent to the secondary system by the primary system after the primary system accepts either the end or the unusual-end signal sent to it from the secondary system. The end response signal turns off any primary latch (primary read, primary write, primary control, primary sense) that is on, and terminates the end or unusual-end condition in the secondary system.

Operational-Out Line

An active operational-out line tells the secondary system that the primary system is operating. If the operational-out line becomes inactive during a secondary system read, write, control, or sense operation, the secondary system data transfer is terminated. The status of this line can be program-tested in the secondary system by executing a BRANCH IF INDICATOR ON instruction — B (III) 8.

Stop

The stop signal is used to inform the secondary system that the primary system has terminated the data-transmission operation in process. The secondary system then terminates its own read or write operation and proceeds to the next instruction. Depending on the program procedure, the secondary system sends either an end or an unusual-end signal to the primary system as a recognition signal of the stop.

Secondary System Instructions

The secondary system makes use of three types of instructions to communicate with the primary system:

1. SIGNAL CONTROL instructions.
2. BRANCH IF INDICATOR ON instructions.
3. READ AND WRITE instructions.

These instructions cannot be successfully chained.

Signal-Control Instructions

The SIGNAL CONTROL instructions are used by the secondary system to:

1. perform a particular function in the secondary system, or,
2. inform the primary system of a particular secondary-system condition.

Figure H-21 shows a summary of all SIGNAL CONTROL instructions.

Send Attention Signal

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	A

INSTRUCTIONS	DESCRIPTION
K A or K III A	Originates an attention signal that is sent to the primary system.
K E or K III E	Originates an end signal that is sent to the primary system.
K F or K III F	Originates an unusual-end signal that is sent to the primary system.
K > or K III >	Permits secondary system program interruption by any one of the four primary system-operation commands.
K < or K III <	Prevents secondary system program interruption.

Figure H-21. Signal Control Instruction Summary

Function. This instruction sends an attention signal to the primary system, which sets ON an attention indicator in the primary system, and indicates that the secondary system wants to transfer data with the primary system. When the primary system tests the attention indicator, it also sends an attention response signal back to the secondary system, and the attention signal is reset.

The secondary system does not suspend system operation after executing this instruction, but proceeds to the next instruction.

Word Marks. Word marks are not affected.

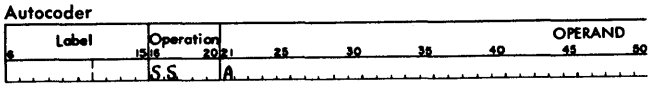
Timing. T = .0333 ms.

Note. For proper operation, this instruction should be followed by one of the primary latch test instructions (primary read, write, sense, or control).

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Abb	Abb

Example. Send an attention signal to the primary system (Figure H-22).



Assembled Instruction: K A

Figure H-22. Send Attention Signal

Send Attention Signal and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	A

Function. This instruction is similar to the SEND ATTENTION SIGNAL instruction, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

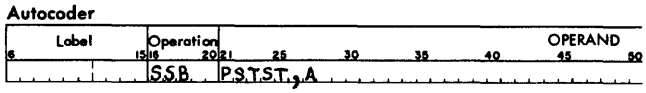
Timing.

Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Send an attention signal to the primary system and branch to core-storage location 0385 (area labeled PSTST); Figure H-23.



Assembled Instruction: K 385 A

Figure H-23. Send Attention Signal and Branch

Send End Signal

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	E

Function. This instruction sends an end signal to the primary system and is usually used to signify that a normal end condition resulted from the previous data transfer between the primary and secondary system. The signal causes the primary system to automatically terminate its operation, and proceed to the next instruction. Either this instruction or the UNUSUAL-END instruction must be executed to end the data transmission.

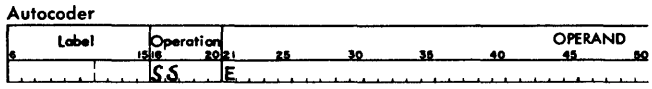
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Ebb	Ebb

Example. Send an end signal to the primary system (Figure H-24).



Assembled Instruction: K E

Figure H-24. Send End Signal

Send End Signal and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	E

Function. This instruction is similar to the SEND END SIGNAL instruction, except that the next instruction is taken from the I-address.

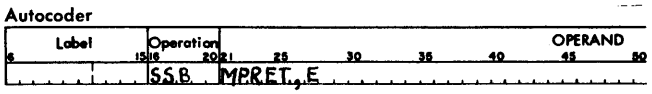
Word Marks. Word marks are not affected.

Timing.
Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Send an end signal to the primary system and branch to core-storage location 0853 (area labeled MPRET); Figure H-25.



Assembled Instruction: K 853 E

Figure H-25. Send End Signal and Branch

Send Unusual-End Signal

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	F

Function. This instruction sends an unusual-end signal to the primary system, and is usually used to signify that some condition has occurred in the secondary system and should be investigated. The signal causes the primary system to automatically terminate its operation, and the secondary system proceeds to the next instruction.

The primary system normally replies to this signal with a secondary system sense operation, which makes it possible for the secondary system to inform the primary system, in more detail, of the condition that caused the unusual end. Either this instruction or the END instruction must be executed to end the data transmission.

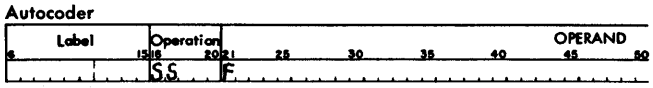
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	Fbb	Fbb

Example. Send an unusual-end signal to the primary system (Figure H-26).



Assembled Instruction: K F

Figure H-26. Send Unusual-End Signal

Send Unusual-End Signal and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	F

Function. This instruction is similar to the SEND UNUSUAL-END SIGNAL instruction, except that the next instruction is taken from the I-address.

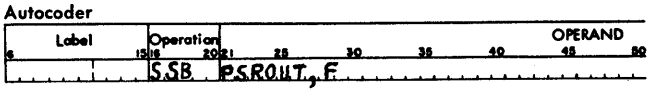
Word Marks. Word marks are not affected.

Timing.
Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Send an unusual-end signal to the primary system and branch to core-storage location 0538 (area labeled PSROUT); Figure H-27.



Assembled Instruction: K 538 F

Figure H-27. Send Unusual-End Signal and Branch

Enable Interrupt

Instruction Format:

Mnemonic	*Op Code*	*d-character*
SS	K	>

Function. This instruction makes it possible for the primary system to interrupt secondary system operation with any one of the four primary system opera-

tion commands (read, write, control, sense), unless the secondary system has already been interrupted and that interrupt is being processed. See Figure G-35 for a list of instructions that can be interrupted.

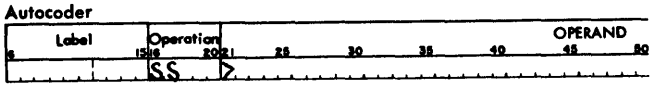
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	>bb	>bb

Example. Permit interruption of the secondary system operation by the primary system (Figure H-28).



Assembled Instruction: K>

Figure H-28. Enable Interrupt

Enable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address,*	*d-character*
SSB	K	III	>

Function. This instruction is similar to the ENABLE INTERRUPT instruction, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

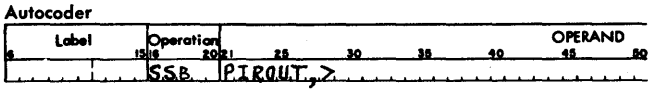
Timing.

Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Permit interruption of the secondary system operation by the primary system and branch to core-storage location 0385 (area labeled PIROUT); Figure H-29.



Assembled Instruction: K 385 >

Figure H-29. Enable Interrupt and Branch

Disable Interrupt

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	<

Function. This instruction prevents any secondary system program interruption by the primary system. Secondary system program interruption cannot occur until the ENABLE INTERRUPT instruction is executed.

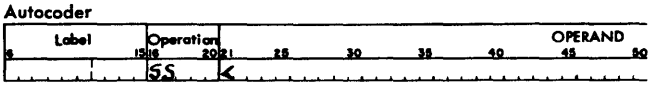
Word Marks. Word marks are not affected.

Timing. T = .0333 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	<bb	<bb

Example. Do not permit interruption of the secondary system operation by the primary system (Figure H-30).



Assembled Instruction: K <

Figure H-30. Disable Interrupt

Disable Interrupt and Branch

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SSB	K	III	<

Function. This instruction is similar to the DISABLE INTERRUPT instruction, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

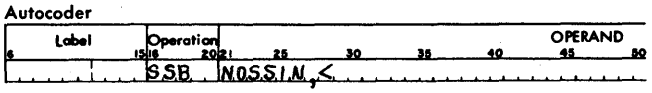
Timing.

Branch (without indexing):
T = .0666 ms.
Branch (with indexing):
T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Branch (without indexing):	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Example. Do not permit interruption of the secondary system operation by the primary system and branch to core-storage location 0853 (area labeled NOSSIN); Figure H-31.



Assembled Instruction: K 853 <

Figure H-31. Disable Interrupt and Branch

Branch if Indicator on Instructions

The BRANCH IF INDICATOR ON instructions — B (III) n are used by the secondary system to check for primary system conditions and secondary system conditions. When a tested condition is present, the program branches to a previously written subroutine. The subroutine begins at the address specified in the BRANCH IF INDICATOR ON instruction.

Figure H-32 shows a summary of all BRANCH IF INDICATOR ON instructions.

Branch if Transmission Error Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	1

INSTRUCTION	DESCRIPTION
B (III) 1	Causes a branch to the specified I - address whenever the transmission - error indicator in the secondary system is ON, when tested.
B (III) 3	Causes a branch to the specified I - address whenever the primary read indicator in the secondary system is ON, when tested.
B (III) 4	Causes a branch to the specified I - address whenever the primary write indicator in the secondary system is ON, when tested.
B (III) 6	Causes a branch to the specified I - address whenever the primary sense indicator in the secondary system is ON, when tested.
B (III) 7	Causes a branch to the specified I - address whenever the primary control indicator in the secondary system is ON, when tested.
B (III) 8	Causes a branch to the specified I - address whenever the status of the operational out line specifies that the primary system is operating.

Figure H-32. Branch If Indicator On Instruction Summary

Function. This instruction tests the condition of the transmission error indicator in the secondary system. This indicator turns ON whenever an A-register error is detected while the secondary system is accepting data from the primary system (primary system write or control operation). When the indicator turns on, it resets the A-register error condition OFF. The system does not stop.

The indicator is also turned on if the primary system stops operating during a read, write, control, or sense operation. If the indicator is ON, when tested, the program branches to the subroutine that starts at the I-address specified in the instruction, and this branch turns off the transmission error indicator.

Word Marks. Word marks are not affected.

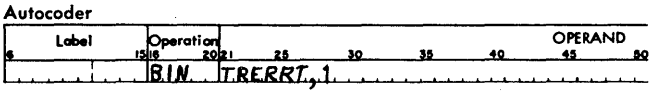
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	1bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the secondary-system transmission error indicator. If the indicator is ON, branch to core-storage location 0385 (area labeled TRERRT); Figure H-33.



Assembled Instruction: B 385 1

Figure H-33. Branch If Transmission Error Indicator On

Branch if Primary-Read Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	3

Function. This instruction tests the status of the primary read indicator located in the secondary system. This indicator turns on whenever the primary system

sends a read command signal to the secondary system, and remains on until the operation ends (end response signal received from primary system). If the indicator is ON, when tested, the program branches to the subroutine that starts at the I-address specified in the instruction.

Word Marks. Word marks are not affected.

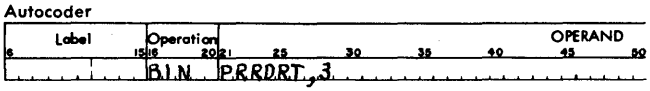
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	3bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the secondary-system primary read indicator. If the indicator is ON, branch to core-storage location 0853 (area labeled PRRDRT); Figure H-34.



Assembled Instruction: B 853 3

Figure H-34. Branch If Primary-Read Indicator On

Branch if Primary-Write Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	4

Function. This instruction tests the status of the primary write indicator located in the secondary system. This indicator turns on whenever the primary system sends a write command signal to the secondary system, and remains on until the operation ends (end response signal received from primary system). If the indicator is ON, when tested, the program branches to the subroutine that starts at the I-address specified in the instruction.

Word Marks. Word marks are not affected.

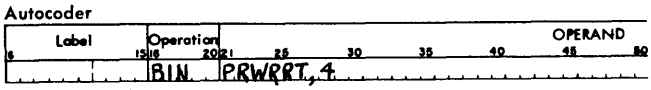
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	4bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the secondary-system primary-write indicator. If the indicator is ON, branch to core-storage location 0538 (area labeled PRWRRT); Figure H-35.



Assembled Instruction: B 538 4

Figure H-35. Branch If Primary-Write Indicator On

Branch if Primary-Sense Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	6

Function. This instruction tests the status of the primary sense indicator located in the secondary system. This indicator turns on whenever the primary system sends a sense command signal to the secondary system, and remains on until the operation ends. If the indicator is ON, when tested, the program branches to the subroutine that starts at the I-address specified in the instruction.

The subroutine, among other things, should contain four characters of status data that specify the secondary system's present status. Refer to Figure H-36 for the characters and some typical conditions.

Word Marks. Word marks are not affected.

Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

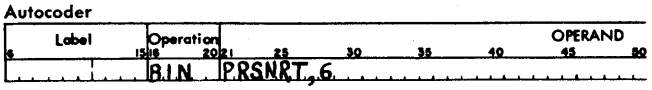
Character	Conditions
1st	Program check Data check Other conditions further specified by 2nd character
2nd	Message - length check No available storage area System does not want data
3rd	Secondary system detects an incoming parity check
4th	Secondary system ready to send a data message to primary system Secondary system input storage area is available

Figure H-36. Sense Data Character Examples

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	6bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the secondary-system primary-sense indicator. If the indicator is ON, branch to core-storage location 0385 (area labeled PRSNRT); Figure H-37.



Assembled Instruction: B 385 6

Figure H-37. Branch If Primary-Sense Indicator On

Branch if Primary Control Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	7

Function. This instruction tests the status of the primary control indicator located in the secondary system. This indicator turns on whenever the primary system sends a control command signal to the secondary system, and remains on until the operation ends. If the indicator is ON, when tested, the program branches to the subroutine that starts at the I-address specified in the instruction.

Word Marks. Word marks are not affected.

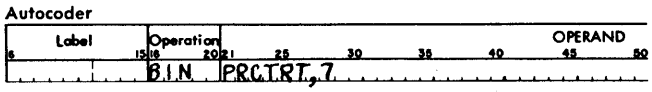
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	7bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the secondary-system primary control indicator. If the indicator is ON, branch to core-storage location 0853 (area labeled PRCTRT); Figure H-38.



Assembled Instruction: B 853 7

Figure H-38. Branch If Primary Control Indicator On

Branch if Operational Out

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	8

Function. This instruction tests the status of the operational line which tells the secondary system when the primary system is operating. If the primary system is operating when tested, the program branches to the subroutine that starts at the I-address specified in the instruction.

Any signals sent by the secondary system when the primary system is not operating are ignored by the primary system.

Word Marks. Word marks are not affected.

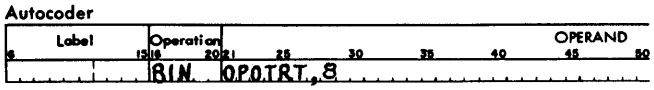
Timing.

No Branch: T = .0666 ms.
Branch (without indexing): T = .0666 ms.
Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	8bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the operational line. If the line is OUT (primary system operating), branch to core-storage location 0385 (area labeled OPOTRT); Figure H-39.



Assembled Instruction: B 385 8

Figure H-39. Branch If Operational Out

Instruction	Description
M (%01) (BBB) R	Data sent from primary system is received by secondary system without word-mark control.
L (%01) (BBB) R	Data sent from primary system is received by secondary system with word-mark control.
M (%01) (BBB) W	Data is sent from secondary system to primary system without word-mark control.
L (%01) (BBB) W	Data is sent from secondary system to primary system with word-mark control.

Figure H-40. Read and Write Instruction Summary

Read and Write Instructions

The READ and WRITE instruction – M or L (%01) (BBB) R or W, initiates the data transmission operation between the primary system and the secondary system, in the specified mode.

These instructions cannot be successfully chained.

The parts of the instruction and their uses are:

M or L. The M or L operation code specifies whether the data transmission is performed in the move or load mode. In both the move and load mode, up to 7 bits per character (CBA8421) are involved in the data transmission. If the load mode is specified, a word-separator character precedes each word-mark-associated character, and each word-separator character.

%01. The A-address (%01) specifies that the secondary system is going to operate with a primary system.

BBB. The B-address specifies the high-order position of the message involved in the data transmission.

R or W. The d-character R specifies a read operation. This d-character is used when the primary system is sending the data to the secondary system. The d-character W is used when the secondary system is sending data to the primary system.

Figure H-40 shows a summary of the READ and WRITE instructions.

Read from Primary

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%01	BBB	R

Function. This instruction specifies a data transmission from the primary system to the secondary system. The data transmission is terminated in one of two ways:

1. A group mark with a word mark sensed in the secondary system ends the operation, and the program then proceeds to the next instruction.
2. An end condition sensed in the primary system (no group mark with a word mark sensed in the secondary system) forces the secondary system to end the operation, and the secondary system proceeds to the next instruction.

Word Marks. Word marks are not stored when operating in the *move* mode (M operation code).

Primary System		*Secondary System*		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*After*	*Read Op Code*
M	ABC	ABC	DEF	ABC	M
M	A‹B	A‹B	DEF	A‹B	M

Note. If the primary system sends data, with word-mark control, to the secondary system, the word marks are transmitted as word-separator characters (‹). For example:

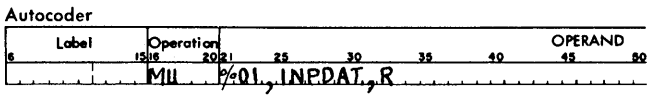
Primary System			Secondary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*After*	*Read Op Code*
⌄ L	⌄ AB	⌄AB	DEF	⌄AB	M
⌄ L	A⌄	A⌄⌄	DEF	A⌄⌄	M

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%61	B + message length + 1

Example. Read data from the primary system and place in core storage, beginning at location 0942 (area is labeled INPDAT); Figure H-41.



Assembled Instruction: M %01 942 R

Figure H-41. Read from Primary

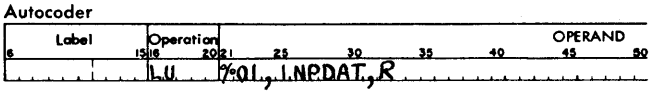
Primary System			Secondary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*After*	*Read Op Code*
⌄ M	⌄ ABC	ABC	DEF	ABC	L
⌄ M	⌄ A⌄B	A⌄B	DEF	ABF	L

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%61	B + message length + 1

Example. Read data from the primary system, with its associated word marks, and place in core storage, beginning at location 0294 (area is labeled INPDAT); Figure H-42.



Assembled Instruction: L %01 294 R

Figure H-42. Read from Primary with Word Marks

Read from Primary with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%01	BBB	R

Function. This instruction is similar to the READ FROM PRIMARY instruction except that word marks in the message area of core storage are removed, and word marks sent from the primary system are written in core storage.

Word Marks. Word marks transmitted from the primary system are written in core storage.

Notes. When the primary system sends data, with word-mark control, to the secondary system, the word marks are transmitted as word-separator characters (⌄). A word-separator character is transmitted as two word-separator characters. For example:

Primary System			Secondary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*After*	*Read Op Code*
⌄ L	⌄ ABC	⌄ABC	DEF	ABC	L
⌄ L	A⌄B	⌄A⌄⌄B	DEF	A⌄B	L

When the primary system sends data, without word-mark control, to the secondary system, the word marks are not transmitted. A word-separator character is transmitted and converted to a word mark. For example:

Write to Primary

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%01	BBB	W

Function. This instruction specifies a data transmission from the secondary system to the primary system. The data transmission is terminated in one of two ways:

1. A group mark with a word mark sensed in the secondary system ends the operation, and the program then proceeds to the next instruction.

2. An end condition sensed in the primary system (no group mark with a word mark sensed in the secondary system) forces the secondary system to end the operation, and the secondary system proceeds to the next instruction.

Word Marks. Word marks are not sent to the primary system when secondary system is operating in the *move* mode (M operation code).

Notes. When the primary system is accepting data, without word-mark control, the word marks in the specified primary system core-storage area are not affected. The data is read into the specified primary system core-storage area as sent from the secondary system. For example:

Secondary System			Primary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*Core Storage After*	*Read Op Code*
M	ABC	ABC	DEF	ABC	M
M	‹AB	‹AB	DEF	‹AB	M

When the primary system is accepting data, with word-mark control, the word marks in the specified primary system core-storage area are erased. Any word-separator characters (‹) transmitted from the secondary system may be permanently altered by the time they are written in the primary system core-storage area. For example:

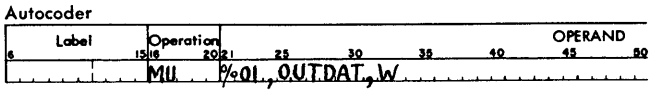
Secondary System			Primary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*Core Storage After*	*Read Op Code*
M	ABC	ABC	DEF	ABC	L
M	‹ABC	‹ABC	DEF	‹BC	L

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%61	B + message length + 1

Example. Send data to the primary system from the core-storage area labeled OUTDAT (first position of the data located in 0942); Figure H-43.



Assembled Instruction: M %01 942 W

Figure H-43. Write to Primary

Write to Primary with Word Marks

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%01	BBB	W

Function. This instruction is similar to the WRITE TO PRIMARY instruction, except that word marks are transmitted to the primary system with the associated data.

Word Marks. Word marks are sent to the primary system when the secondary system is operating in the *load* mode (L operation code).

Notes. When the primary system is accepting data, without word-mark control, the word marks in the specified primary system core-storage area are not affected. The data is read into the specified primary system core-storage area as sent from the secondary system. A word-separator character is transmitted as two word-separator characters.

Secondary System			Primary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*Core Storage After*	*Read Op Code*
L	AB	‹AB	DEF	‹AB	M
L	‹A	‹‹A	DEF	‹‹A	M

When the primary system is accepting data, with word-mark control, the word marks in the specified primary system core-storage area are erased. A word-separator character is transmitted as two word-separator characters. Any word separator character (‹) transmitted from the secondary system may be permanently altered by the time they are written in the primary system core-storage area. For example:

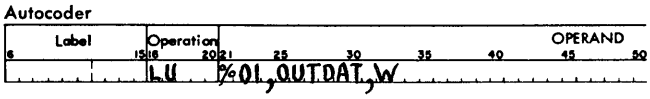
Secondary System			Primary System		
Write Op Code	*Core-Storage Contents*	*Transmitted As*	*Core Storage Before*	*Core Storage After*	*Read Op Code*
L	ABC	‹ABC	DEF	ABC	L
L	‹AB	‹‹A‹B	DEF	‹AB	L

Timing. T = .0999 ms + transmission time.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%61	B + message length + 1

Example. Send data to the primary system with its associated word marks, from the core-storage area labeled OUTDAT (first position of the data located in 0429); Figure H-44.



Assembled Instruction: L %01 429 W

Figure H-44. Write to Primary with Word Marks

Feature Operation

With the signals and instructions just described, the specific type of data transmission desired is set up. To illustrate one type of data transmission operation, assume that the primary system wants to receive data from the secondary system. The program procedure shown in Figure H-45 and the accompanying writeup are presented only as an example that shows the use of the various signals and instructions in an operation. This example should not be considered the optimum procedure for this type of operation.

1. The primary system signals the secondary system that it wants to read from the secondary system by sending it a read command signal.
2. The read command signal sets ON the primary read latch in the secondary system. If program

interruption can occur (prior execution of K > or K (III) > instruction), the secondary system *immediately branches* to the interrupt subroutine, which includes the program shown in Figure H-45.

3. The secondary system program tests the various primary indicators and finds the primary read indicator set ON. If all indicators had been turned off, the main program execution would continue – step 3A.

4. With the primary read indicator ON, the secondary system is checked to see if it wants to ignore the primary system read command.

4A. If the secondary system does want to ignore the read command, a SEND UNUSUAL-END SIGNAL instruction is executed, which signals the primary system that some condition has occurred in the secondary system and the condition should be investigated. The program then branches back to the point where the primary sense indicator is checked to see if it was ON.

4B. The primary system recognizes and accepts the unusual-end signal, and generates an end-response signal that turns off the primary read indicator in the secondary system.

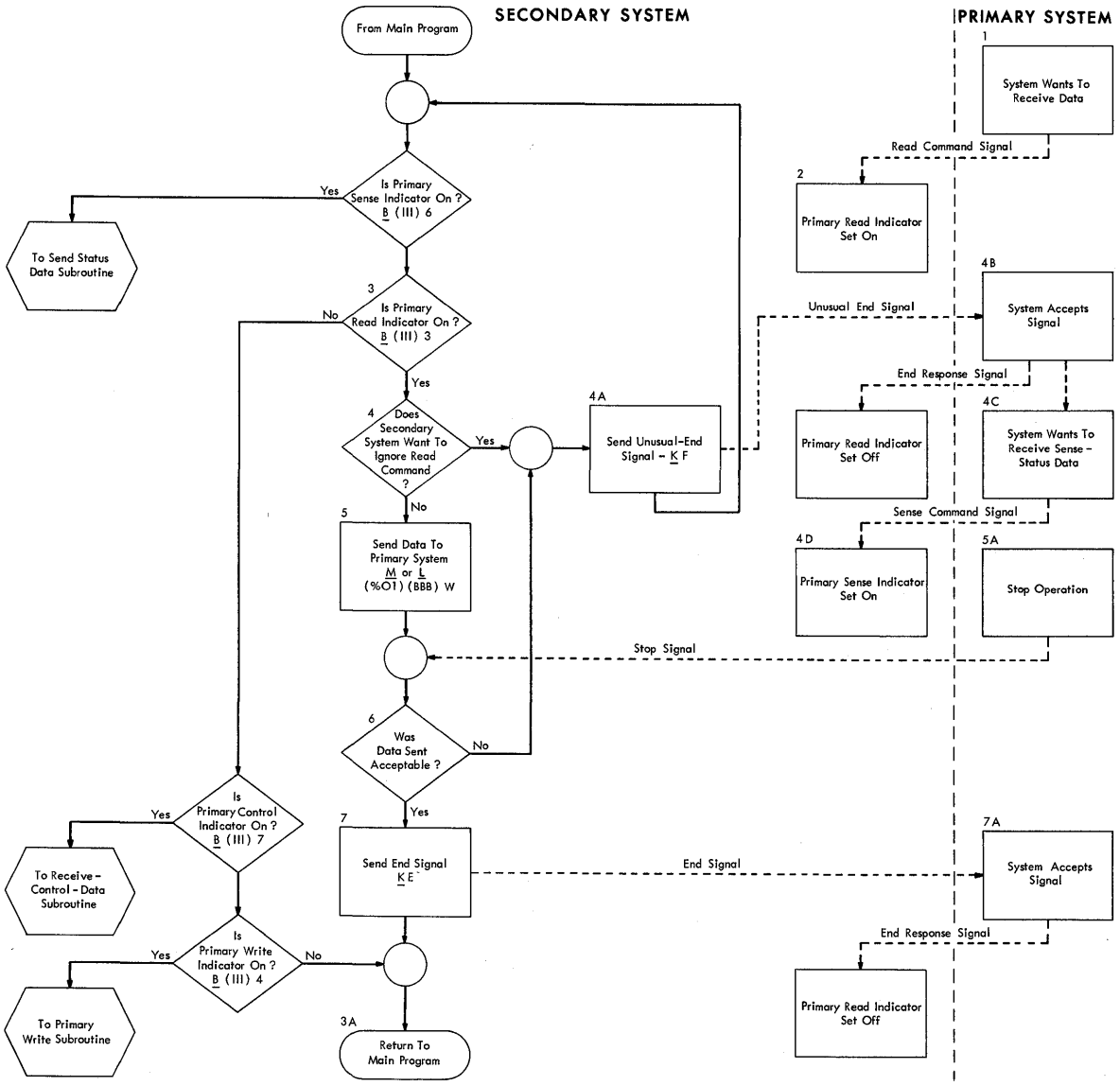


Figure H-45. Data Transmission Operation Schematic

4C. The primary system signals the secondary system that it wants to receive status data from the secondary system by sending it a sense command signal.

4D. The sense command signal sets ON the primary sense indicator in the secondary system.

5. If the secondary system wants to accept the read command, the secondary system executes a WRITE instruction – M (%01) (BBB) W.

The actual data transmission begins and continues until one of the systems ends the operation. The secondary system ends the operation when a group mark with a word mark is encountered in core storage.

5A. The primary system could end the operation by using the stop line. This would cause the secondary system to end its write operation; and, depending on program procedure, send either an end or an unusual-end signal to the primary system as a recognition signal of the stop. The end or unusual-end signals are accepted and the procedures discussed elsewhere in the example writeup are carried out (steps 6, 4A-4D or 7).

6. After the data transmission ends, the data is checked to see if it was acceptable. If the data sent was not acceptable, then steps 4A-4D are repeated at this time.

7. If the data sent was acceptable, a SEND END SIGNAL instruction is executed, which sends an end signal to the primary system.

7A. When the primary system recognizes and accepts the end signal, it generates an end-response signal that turns off the primary read indicator in the secondary system.

3A. With the completion of the data transmission operation, the next sequential instruction in the main program is executed.

Notes:

The conditions that specify the terminating system are:

1. The writing system terminates the data transmission when the read input area is longer than the write output area.
2. Both the reading and writing systems terminate the data transmission when the read input area is the same size as the write output area.
3. Both the reading and writing systems terminate the data transmission when the read input area is one core-storage position shorter than the write output area.
4. The reading system terminates the data transmission when the read input area is shorter than the write output area by more than one core-storage position.

Indexing and Store Address Register

This feature provides the IBM 1440 Data Processing System with greater program flexibility, by making address indexing and address storing more automatic.

Indexing

The indexing portion of the indexing and store address register special feature provides three 3-position index locations (registers) that can be used to modify addresses automatically. These three index registers are part of core storage and can be used as normal storage positions when not being used as index register locations. The core-storage addresses assigned and the index register numbers are:

Index Register Numbers	*Core-Storage Positions*
1	087-089
2	092-094
3	097-099

Factors contained in the index registers can be initialized and modified in several ways. The index factor can be placed in the index register by normal programming (ADD or MOVE instructions, for instance), and the factor can be changed (add or subtract operations, for example). In these instances, a word mark must be set in the high-order position of the index register prior to inserting or changing the index factor.

The two-address STORE B-ADDRESS REGISTER instruction is usually easier to use. The ADD and SUBTRACT instructions can cause unwanted zone-bits to appear in the resultant factor. Figure H-46 illustrates how index register 1 can be initialized to zeros, index register 2 incremented by 10, and index register 3 decremented by 3.

Autocoder

Label	Operation	OPERAND
	SBR	X1,0
	SBR	X2,10+X2
	SBR	X3,15997+X3

Assembled Instruction: H 089 000
H 094 0J0
H 099 IIG

Note: Assuming the Autocoder equates the 1440 index registers to the actual machine addresses: X1 = 089, X2 = 094, and X3 = 099.

Figure H-46. Initializing, Incrementing, and Decrementing Index Registers

Both the A-address and/or the B-address can be modified by the factor contained in any one of the three index registers; however, only core-storage address can be modified.

The A- and/or B-address specifies which index register is to be used by a combination of A- and B-bits in the tens position of the address. The bit combinations and the registers they specify are:

Bit Combination	*Index Register Numbers*	*Zone Punch*
A-bit, No B-bit	1	Zero
B-bit, No A-bit	2	Eleven
A-bit, B-bit	3	Twelve

When the tens position of an A- and/or B-address contains one of these zone-bit combinations, the address is referred to as being tagged.

OP	*Positions Tagged* ↓	↓
OP	AAA	BBB

Note: After any arithmetic operation that affects the indexing factor, zones that appear in the units or tens position of any index location must be removed. (The modify-address instruction is not, in this sense, an arithmetic function.)

The modification of the A- and/or B-address occurs in their respective address registers. For instance, if the A-address is indexed, the indexing occurs in the A-address register. This means the original instruction in storage is in no way changed or modified.

1. The A-address and B-address are analyzed for indexing as they are moved into the address registers.
2. The contents of the proper index location (indexing factor) is added to the contents of the address register and develops the effective address there, when indexing is indicated.
3. Three or four additional cycles are required for each address indexed. (The fourth cycle is required when the new address has been modified so that it is in the next highest group of 4,000 storage positions. The zone-bit configuration in the units position must reflect the new group of positions.)

Increasing an Address

To increase a core-storage address using the indexing feature, the contents of the index location is added to the selected address register. Figure H-47 illustrates various methods of address modification using the index locations.

		INSTRUCTION IN STORAGE	INDEX LOCATION 1	2	3	EFFECTIVE INSTRUCTION
1. INDEX THE B-ADDRESS	BEFORE	M 080 1A7	010	025	050	
	AFTER	M 080 1A7	010	025	050	M 080 167
2. INDEX A- AND B-ADDRESS	BEFORE	M 0S0 1J7	010	025	050	
	AFTER	M 0S0 1J7	010	025	050	M 030 142
3. INDEX A- AND B-ADDRESS	BEFORE	M JB0 8C0	010	025	050	
	AFTER	M JB0 8C0	010	025	050	M J70 880

Figure H-47. Indexing

Decreasing an Address

To decrease an address, the 16,000's complement of the amount to be subtracted from the address must be stored in the index location.

Example.

Decreasing Required: Decrease a B-address by 10
Indexing Factor (Complement): 16,000 − 10 (15,990)

The 15,990 converts to the three digit factor I9? (Figure H-48).

Using the modulus 16 rules, the arithmetic overflow adds an A-bit in the hundreds position (both the hundreds and units positions already contain A- and B-bits, the combination of which indicates a 15,000-15,999 block address). The addition of the A-bit increases the value of the zone bits to 16 which, according to modulus 16, has an address value of 0 (000-999 block address). Therefore, the new address is 927. With the indexing feature, even though there was an overflow, the arithmetic overflow indicator is not turned on.

Store Address Register

The store address register portion of the indexing and store address register special feature make it possible to store the contents of the A- and B-address registers. Thus, the A- and B-addresses of program instructions can be modified directly in cases where variable length records are being processed. This facility also makes it easier to re-enter the main program from a subroutine. Because the address of the next instruction in sequence can be retained, program re-entry is simplified.

	INSTRUCTION IN STORAGE	INDEX LOCATION 1	2	3	EFFECTIVE INSTRUCTION
BEFORE	L 123 9I7	I9?			
AFTER	L 123 9I7	I9?			L 123 927

937 + I9? = 927 (with overflow)
I = (AB9)
? = (AB0)

Figure H-48. Converting Address

A subroutine is a set of program instructions that are executed, if a particular condition arises during the main routine. For example, if an unequal compare occurs during processing, the program branches to a subroutine in which a special set of instructions handles the condition.

Each time a subroutine is used, some method must be employed to link it with the main program. The function of the STORE A-ADDRESS REGISTER, and STORE B-ADDRESS REGISTER instructions is to establish subroutine linkage so that upon leaving the sequence of the main program it is possible to execute the steps of the subroutine, and return to the main program where the sequence was interrupted.

Store A-Address Register

Instruction Format.

Mnemonic	*Op Code*	*A-address*
SAR	Q	AAA

Function. This instruction stores the contents of the A-address register from the previous operation, in the 3-position field that has its units position defined by the A-address of the STORE A-ADDRESS REGISTER instruction.

Word Marks. Word marks are not affected.

Timing. T = .0111 (L_I + 5 or 6*) ms.

* Plus 5 or 6, depending on the presence or absence of zone bits in the units position of the address being stored.

Address Registers After Operation.

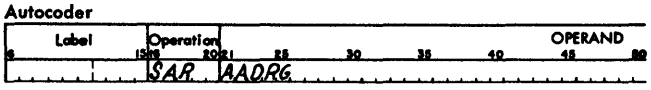
I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-3	Ap

Chaining. If this instruction is chained (operation code only) to the preceding operation, the following will result:

M 617 xxx	Base instruction. (Assume 1-position field.)
Q 613	Normal STORE A-ADDRESS REGISTER operation. The address factor 616 is stored in locations 611, 612, and 613.
Q	Chained STORE A-ADDRESS REGISTER instruction. The address factor 610 will be stored in locations 608, 609, and 610.
NSI	

If the A-address factor is required in more than one place, follow the base instruction (which leaves the address factor to be stored) with one STORE A-ADDRESS REGISTER instruction, followed by as many STORE B-ADDRESS REGISTER instructions as might be required to satisfy the program requirements.

Example. Store the contents of the A-address register in area labeled AADRG (0625); Figure H-49.



Assembled Instruction: Q 625

Figure H-49. Store A-Address Register

Store B-Address Register (One Address)

Instruction Format.

Mnemonic	*Op Code*	*A-address*
SBR	H	AAA

Function. This instruction stores the contents of the B-address register resulting from the previous operation, in the 3-position field that has its units position defined by the A-address of the STORE B-ADDRESS REGISTER instruction.

Word Marks. Word marks are not affected.

Timing. $T = .0111\ (L_I + 4 \text{ or } 5^*)$ ms.

* Plus 4 or 5, depending on the presence or absence of zone bits in the units position of the address being stored.

Note. When indexing is installed in the 1440, the functioning of all branch commands is altered to simplify subroutine linkage. With these alterations, each time a branch occurs as a result of one of these commands, the address of the next sequential instruction in the main routine is inserted in the B-address register.

Although the subroutine may be entered from many distant points in the main program, this use of the SBR operation makes the subroutine linkage complete.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-3	Bp

IBM

Form X24-1350-2
Printed in U.S.A.

Program
Programmed by
Date

INTERNATIONAL BUSINESS MACHINES CORPORATION
AUTOCODER CODING SHEET
IBM 1401-1410-1440-1460

Identification 76 80
Page No. 1 2 of

Line	Label	Operation	OPERAND
0 1			MAIN PROGRAM STEPS.
0 2		B	MULTRU BRANCH TO MULTIPLY SUBROUTINE.
0 3		A	FLD1,FLD2 FIRST MAIN-PROGRAM STEP AFTER
0 4			MULTIPLY SUBROUTINE.
0 5			OTHER MAIN-PROGRAM STEPS.
0 6	MULTRU	SBR	MULAST+3 SETUP BRANCH-BACK LINKAGE.
0 7			MULTRU STEPS.
0 8	MULAST	B	0 BRANCH-OUT INSTRUCTION (BRANCHES TO
0 9			MAIN-PROGRAM STEP FOLLOWING
1 0			B MULTRU INSTRUCTION).
1 1			

Assembled Instructions:	B	495	Branch to multiply subroutine.
	A	880 990	First main-program step after multiply subroutine. This instruction-address is stored in the branch-out instruction: MULAST +3.
	~~		Main-program steps.
	H	654	Store address of instruction following B MULTRU in I-address of last multiply-subroutine step.
	~~		Multiply subroutine steps.
	B	000	Completes linkage between MULTRU subroutine and main program.

Figure H-50. Store B-Address Register (One Address)

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. The main routine branches to a multiply subroutine labeled MULTRU (0495). This example shows the last step in the main routine, and the first and last steps of the multiply routine, and illustrates subroutine linkage (Figure H-50). The last instruction (labeled MULAST), plus three, contains the address of the next instruction in the main routine.

Store B-Address Register (Two Addresses)

Instruction Format.

Mnenomic	*Op Code*	*A-address*	*B-address*
SBR	H	AAA	BBB

Function. This instruction stores the present contents of the B-address register in the 3-position field that has its units position defined by the A-address of the STORE B-ADDRESS REGISTER instruction. The B-address register contains the number specified by the B-register portion of the STORE B-ADDRESS REGISTER instruction.

Word Marks. Word marks are not affected.

Timing. $T = .0111\ (L_I + 4 \text{ or } 5^*)$ ms.

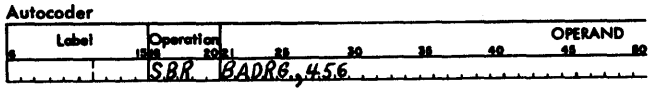
* Plus 4 or 5, depending on the presence or absence of zone bits in the units position of the address being stored.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-3	B

Chaining. This instruction can be chained to the preceding operation (if that instruction left usable address-register contents) by supplying only the operation code.

Example. Store contents (0456) of the B-address register (any 3-character factor) in the area labeled BADRG (0123); Figure H-51.



Assembled Instruction: H 123 456

Figure H-51. Store B-Address Register (Two Addresses)

Multiply-Divide

This feature makes it possible to perform direct multiplication and division in the IBM 1440 Data Processing System.

Multiply

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
M	@	AAA	BBB

Function. The multiplicand (data located in the A-field) is repetitively added to itself in the B-field. The B-field contains the multiplier in the high-order positions, and enough additional positions (low order) to allow for the development of the product. At the end of the multiply operation, the units position of the product is located at the B-address. The multiplier is destroyed in the B-field as the product is developed. Therefore, if the multiplier is needed for subsequent operations, it must be retained in another storage area.

The multiply-divide feature for the 1440 system has additional circuitry that automatically eliminates readdressing machine cycles when recomplementing is required during the operation.

Rules:

1. The product is developed in the B-field. The length of the B-field is determined by adding *1* to the sum of the number of digits in the multiplicand and multiplier fields.

Example:

```
  1246      4-digit multiplicand
× 543       3-digit multiplier
          + 1
          ---
            8 positions must be allowed in
              the B-field.
```

2. A word mark must be associated with the high-order positions of both the multiplier and multiplicand fields.
3. A- and B-bits need not be present in the units positions of the multiplier and multiplicand fields. The absence of zone bits in these positions indicates a positive sign. At the completion of the multiply operation the B-field has zone bits in the units position of the product only. The multiply operation uses algebraic sign control (Figure H-52).

Multiplier Sign	+	+	−	−
Multiplicand Sign	+	−	+	−
Sign of Product	+	−	−	+

Figure H-52. Algebraic Sign Control for Multiplication

4. Zone bits that appear in the multiplicand field are undisturbed by the multiply operation. Zone bits in the units position of the multiplicand are interpreted for sign control.

Timing. The average time required for a multiply operation is:

$T = .1110 + 2L_C + 5L_CL_M + 7L_M$ ms.

L_C = length of multiplicand field.
L_M = length of multiplier field.

A chart of approximate timing is included in the section on *Multiply and Divide Timing.*

Notes. The first addition within the multiply operation inserts zeros in the product field from the storage location specified by the B-address up to the units position of the multiplier.

The A-address register and the B-address register indicate positions within the A- and B-fields on which operations are currently being performed.

Word Marks. A word mark must be associated with the high-order positions of the multiplier and multiplicand fields.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_C	B-L_P

L_P = Length of product field.
L_C = Length of multiplicand field.

Chaining. This instruction cannot be successfully chained.

Example. Multiply:

Label	*Location of Data Word*	*Contents of Data Word*	*Description*
MULCAN	0502	1246	Multiplicand
MULIER	0065	543	Multiplier
PRODCT	0610		Product

The size of the product field is 4 + 3 + 1 = 8. The multiplier is placed in the three high-order positions of the PRODCT area (0603, 0604, and 0605). At the completion of the multiply operation, load the product in the area labeled OUT2 (0178). The units positions of the multiplier and multiplicand fields may be signed (Figure H-53).

Autocoder

```
Label     Operation     OPERAND
          ZA            MULIER,PRODCT-5
          M             MULCAN,PRODCT
          MLCWA         PRODCT,OUT2
```

Assembled Instruction: ? 065 605
@ 502 610
L 610 178

Figure H-53. Multiply

Divide

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*
D	%	AAA	BBB

Function. This instruction divides the data (dividend) in the low-order positions of the B-field by the divisor located in the A-field, and develops the quotient in the high-order positions of the B-field. The remainder is left in the low-order positions of the B-field.

Rules:

1. The quotient is developed in the B-field. The length of the B-field is determined by adding 1 to the sum of the number of digits in the divisor and dividend fields.

Example:

543 ⟌ 1246

4-digit dividend
3-digit divisor
\+ 1
8 positions must be allowed in the B-field.

2. A word mark must be associated with the high-order position of the A-field.
3. In all cases either A- and B-bits (plus sign) or B-bit (minus sign) must appear in the units position of the dividend field. The divisor may be either signed or unsigned. If there are no bits in the units position of the divisor, the machine assumes the divisor factor is positive. The divide operation uses algebraic sign control (Figure H-54).
4. The dividend is loaded in the low-order positions of the B-field (Figure H-55) by a ZERO AND ADD instruction to ensure that zeros are present in the high-order positions of the B-field.
5. The B-address in the DIVIDE instruction specifies the high-order position of the dividend.

   At the completion of division:

Divisor Sign	+	+	−	−
Dividend Sign	+	−	+	−
Quotient Sign	+	−	−	+
Remainder Sign	+	−	+	−

Figure H-54. Algebraic Sign Control for Division

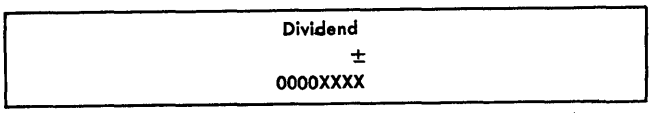


Figure H-55. Dividend in B-Field

   a. The quotient is in the high-order positions of the B-field. The location of the units position of the quotient, is the address of the units position of the dividend, minus the length of the divisor, minus one.
   b. The remainder is in low-order positions of the B-field.
   c. The sign of the quotient is over the units position of the quotient field.
   d. Because only one quotient digit can be developed at a time, it is important to address the high-order position of the dividend (B-address of the DIVIDE instruction). This ensures that the first divide operation results in a single high-order quotient digit. A dividend improperly addressed can cause an arithmetic overflow if the result of the first divide operation is greater than 9.

Note: A divide operation refers to the process of developing each quotient digit. If the quotient field is not large enough, no overflow is indicated. The machine does not check for this programming error. Division by zero results in an arithmetic overflow condition. Figure H-56 shows the result of a divide operation.

Extra zeros can be added to the dividend prior to a divide operation when a larger quotient is required. For each additional quotient digit desired, place one zero to the right of the dividend as shown in Figure H-57. Note that in this example, the units position of the quotient is *not* located in the position previously described in Item 5a.

Word Marks. A word mark must define the high-order position of the divisor.

Timing. Average time required for the execution of a divide operation is calculated:

$T = .0999 + 7L_RL_Q + 8L_Q$ ms.
L_Q = length of the quotient field.
L_R = length of the divisor field.

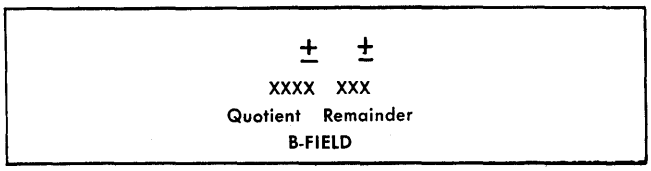


Figure H-56. Location of the Results of a Divide Operation

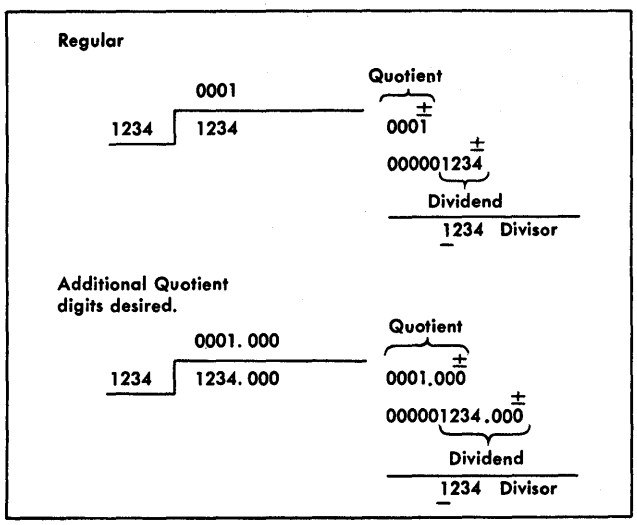


Figure H-57. Additional Quotient Digits

A chart of approximate timings is included in the section on *Multiply and Divide Timing.*

Note. The quotient field is not cleared before actual division begins.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	A-L_R	Tens position of quotient. If divisor has all zeros, the B-address register stands at the units position of the dividend, minus the length of the divisor, minus the length of the dividend, minus 1.

Chaining. This instruction cannot be successfully chained.

Example. Figure H-58 is a symbolic example for DIVIDE.

Label	*Locaton of Data Word*	*Data Word*	*Description*
DIVEND	0502	1246	Dividend
DIVSOR	0065	543	Divisor
QUOT	0985		Quotient

Autocoder

Label	Operation	OPERAND
	ZA	DIVEND,QUOT
	D	DIVSOR,QUOT-3

Assembled Instruction: ? 502 985
% 065 982

Figure H-58. Divide

Multiply and Divide Subroutine

These are subroutines for multiply and divide operations, discussed here to illustrate programming methods and to aid in programming machines not equipped with the multiply-divide special feature. These are not the only methods of performing these operations; they are typical methods. These sample (actual-machine language) programs are easily converted to Autocoder language to fit the particular application.

Multiply Subroutine

The block diagram in Figure H-59 illustrates the logic used to develop the multiply subroutine discussed here. The subroutine provides for a maximum of a 9-digit multiplier, 11-digit multiplicand, and a 20-digit product, and uses positive factors.

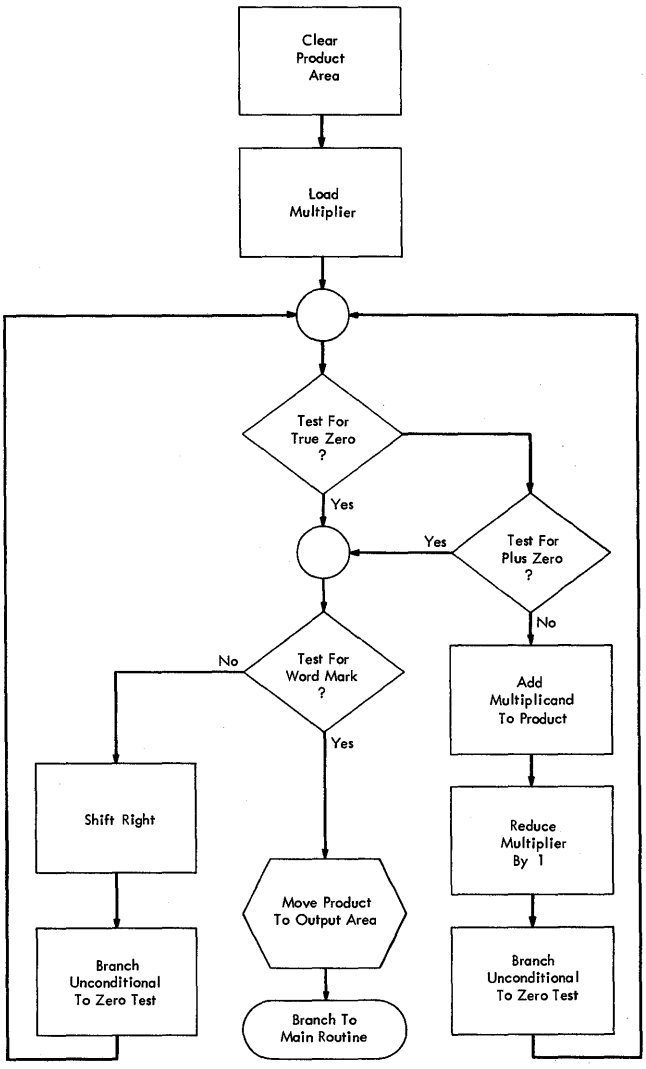


Figure H-59. Multiply Flow Chart

The subroutine is written in actual language (Figure H-60), and occupies the 900 block of storage. A multiplier area is provided in storage positions 901-909, and the product area is assigned in storage positions 910-929. The multiplicand can be located anywhere.

Any program that uses this subroutine must include a step that moves the multiplier address (XXX) to location 937 and the multiplicand address (YYY) to location 960.

At the completion of the multiply subroutine, the program instruction step 12 is a branch to the main program or stop instruction.

The routine starts in storage position 930. The product is found in 929 for a 9-digit multiplier, 928 for 8-digit, 927 for 7-digit, 926 for 6-digit, etc.

Note: The multiply subroutine results in blanks instead of zeros in the low-order position of a product when the multiplier contains low-order zeros. To correct this situation, set the product area to zeros.

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION FORM X24-6437 PRINTED IN U.S.A.

PROGRAM CHART
IBM 1401-1440-1460

Program: Multiply Subroutine Programmer: Date:

Step No.	Inst. Address	Instruction O	A/I	B	Remarks	Effective No. of Characters Inst.	Data	Total
1	930	/	92 9		Clear Product Area	4		
2	934	L	XXX	909	Load Multiplier	7		
3	941	B	975	909 0	Test 909 for true zero	8		
4	949	B	975	909 ?	No true zero - test for plus zero	8		
5	957	A	YYY	921	No zero - add Multiplicand to Product	7		
6	964	S	994	909	Reduce Multiplier by 1	7		
7	971	B	941		Branch to zero test	4		
8	975	V	995	909 1	Test 909 for Word Mark	8		
9	983	L	928	929	No Word Mark - Shift Right	7		
10	990	B	941		Branch to zero test	4		
11	994	1			Constant "1"	1		
12	995	B	ZZZ		Multiplication complete - Branch back to Program			
			XXX		Location of Multiplier			
			YYY		Location of Multiplicand			
			ZZZ		Address of next Program Stop			
					9 digit Multiplier			
					11 digit Multiplicand			
					20 digit Product			

Figure H-60. Multiply Subroutine (Actual)

Divide Subroutine

The restrictions placed on this subroutine (Figures H-61 and H-62) are:

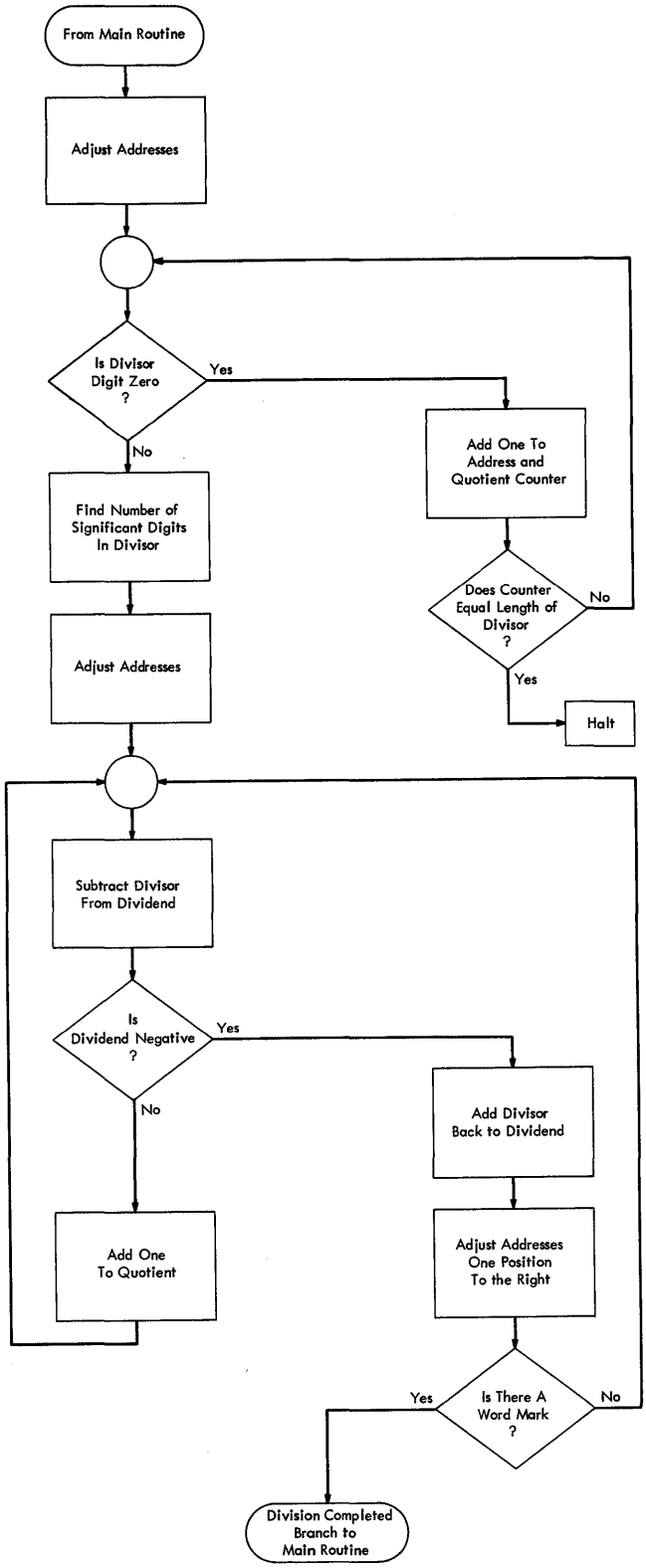


Figure H-61. Divide Flow Chart

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

PROGRAM CHART

IBM 1401-1440-1460

FORM X24-6437
PRINTED IN U.S.A.

Program: Divide Routine Example Programmer: _____ Date: _____

Step No.	Inst. Address	Instruction OP	A/I			B			d	Remarks	Effective No. of Characters Inst.	Data	Total
	516	M	5	0	7	5	2	9		Store Address of Word Mark Position (High Order) of Divisor			
	523	B	6	6	2	Y	Y	Y	0	Branch if Divisor Digit Equals Zero			
	531	S	5	1	2	5	1	5		(Length of Divisor) - (Number of High Order Zeros)			
	538	A	5	1	5	5	0	1		Adjust Dividend Address			
	545	A	5	1	5	5	0	4		Modify Addresses			
	552	S	5	1	3	5	0	1					
	559	S	5	1	3	5	0	4					
	566	Y	7	5	5	5	0	1		Clear zone from low-order position to prepare for address assignment.			
	573	Y	7	5	5	5	0	4		Clear zone from low-order position to prepare for address assignment.			
	580	M	5	0	1	6	4	2		Set Modified Addresses into Divide Routine			
	587	M	5	0	1	6	4	9					
	594	M	5	0	1	6	9	8					
	601	M	5	0	1	7	0	5					
	608	M	5	0	1	7	1	9					
	615	M	5	0	4	6	5	7					
	622	M	5	1	0	6	3	9		Set Divisor Address			
	629	M	5	1	0	6	9	5					
	636	S	Z	Z	Z	W	W	W		Subtract Divisor from Dividend			
	643	V	6	9	2	W	W	W	K	Branch if Negative Result			
	651	A	5	1	3	X	X	X		Add One to Quotient			
	658	B	6	3	6					Repeat Subtraction			
	662	A	5	1	3	5	2	9		Add One to YYY Address			
	669	A	5	1	3	5	1	2		Increase Counter by One			
	676	C	5	1	2	5	1	5		If Equal, Divisor Equals Zero			
	683	B	5	2	3				/	Branch Unequal			
	688	.	7	6	0					Halt - cannot Divide by Zero			
	692	A	Z	Z	Z	W	W	W		Add Divisor to Dividend			
	699	Y	7	5	5	W	W	W		Move blank zone to word mark position of dividend			
	706	A	5	1	3	7	1	9					
	713	V	7	6	0	W	W	W	1	Test for End of Divide			
	721	A	5	1	3	6	4	2		Modify Addresses to Develop Next Quotient Digit			
	728	A	5	1	3	6	4	9		" " " " " "			
	735	A	5	1	3	6	5	7		" " " " " "			
	742	A	5	1	3	6	9	8		" " " " " "			
	749	A	5	1	3	7	0	5		" " " " " "			
	756	B	6	3	6					Return to Divide Calculations			
	760									Divide Complete			

DATA FOR DIVISION SUBROUTINE

Location of Data Word	Data Word	Description of Data
501	WWW	Address of word mark position (high order) of dividend
504	XXX	Address of word mark position (high order) of quotient
507	YYY	Address of word mark position of divisor
510	ZZZ	Divisor Address
512	00	Counter for number of zeros in divisor
513	1	Constant
515	NN	Length of the divisor

Figure H-62. Divide Subroutine (Actual)

1440 Multiply Times Based on Multiply Subroutine (milliseconds)												
Number of Positions in Multiplicand →		1	2	3	4	5	6	7	8	9	10	11
Number of Positions in Multiplier	1	3.387	3.503	3.609	3.725	3.831	3.937	4.053	4.169	4.275	4.391	4.507
	2	6.851	7.073	7.295	7.517	7.739	7.961	8.183	8.405	8.627	8.849	9.071
	3	10.364	10.692	11.030	11.358	11.696	12.024	12.362	12.690	13.028	13.356	13.693
	4	13.915	14.359	14.083	15.247	15.691	16.135	16.579	17.023	17.467	17.910	18.354
	5	17.515	18.074	18.625	19.184	19.734	20.294	20.844	21.404	21.954	22.513	23.064
	6	21.162	21.828	22.494	23.160	23.826	24.492	25.158	25.823	26.489	27.155	27.821
	7	24.858	25.640	26.412	27.194	27.966	28.747	29.519	30.301	31.073	31.855	32.627
	8	28.593	29.481	30.369	31.256	32.144	33.032	33.920	34.808	35.695	36.583	37.471
	9	32.366	33.370	34.364	35.367	36.361	37.365	38.359	39.362	40.356	41.360	42.354

Figure H-63. IBM 1440 Multiply Times (Based on Multiply Subroutine)

1. The dividend and quotient fields must be of equal length.
2. The dividend and divisor must both be positive.
3. The divisor must have no zone for its positive indication. This is necessary only if the divisor could be zero.
4. The divisor cannot contain more than nine leading zeros.
5. All fields must be located completely below address 999.
6. At the completion of the subroutine, the address of the units position of the quotient can be found in the B-address of the instruction located in 651.
7. The remainder is left in the dividend field.
8. A word mark must be located immediately to the right of the units position of the dividend.
9. The quotient area must be preset to zeros or blanks to develop the correct quotient. If the area is not zeroed or blanked, the quotient is added to whatever is there. The positions added depend on the number of leading zeros in the divisor.
10. The information shown in *Data for Division Subroutine* (Figure H-62), except the constant I in location 513, must be set initially for each desired execution of the divide subroutine. The two addresses in locations 507 and 510, associated with the divisor, are not altered. Thus, they do not have to be reinitialized if the divisor is contained in the same area.

Multiply and Divide Timings

The four timing charts give the approximate timings of multiply (Figures H-63 and H-64) and divide (Figures H-65 and H-66) operations. Two of the charts are based on the timings when a subroutine written in actual language is used. The other two charts are based on the timings required when the system is equipped with the special feature for multiply and divide.

1440 Multiply Times With Special Feature (milliseconds)												
Number of Positions in Multiplicand →		1	2	3	4	5	6	7	8	9	10	11
Number of Positions in Multiplier	1	.266	.335	.403	.472	.540	.609	.677	.746	.814	.883	.951
	2	.400	.533	.666	.799	.932	1.065	1.199	1.332	1.465	1.598	1.731
	3	.532	.722	.911	1.100	1.289	1.478	1.668	1.857	2.046	2.235	2.424
	4	.666	.910	1.154	1.398	1.643	1.888	2.132	2.376	2.620	2.864	3.108
	5	.799	1.099	1.399	1.699	1.999	2.300	2.600	2.900	3.200	3.500	3.800
	6	.932	1.287	1.642	1.998	2.353	2.708	3.063	3.418	3.773	4.128	4.483
	7	1.065	1.476	1.888	2.299	2.710	3.121	3.532	3.943	4.354	4.765	5.176
	8	1.198	1.665	2.131	2.597	3.063	3.529	3.995	4.461	4.927	5.393	5.859
	9	1.331	1.854	2.376	2.898	3.420	3.942	4.464	4.986	5.508	6.030	6.552
	10	1.465	2.042	2.619	3.196	3.773	4.350	4.927	5.504	6.081	6.659	7.236

Figure H-64. IBM 1440 Multiply Times (with Special Feature)

1440 Divide Times Based on Divide Subroutine (millseconds)											
Number of Positions in Quotient →		1	2	3	4	5	6	7	8	9	10
Number of Positions in Divisor	1	8.204	13.223	18.242	21.965	25.487	29.008	32.529	36.050	39.572	43.093
	2	8.560	13.413	18.266	22.107	25.794	29.482	33.169	36.856	40.544	44.231
	3	8.915	13.601	18.287	22.253	26.107	29.961	33.816	37.670	41.524	45.378
	4	9.270	13.790	18.310	22.396	26.416	30.436	34.456	38.476	42.497	46.517
	5	9.625	13.978	18.331	22.541	26.729	30.916	35.103	39.290	43.477	47.664
	6	9.980	14.167	18.354	22.684	27.037	31.390	35.744	40.097	44.450	48.803
	7	10.335	14.355	18.376	22.830	27.350	31.870	36.390	40.910	45.430	49.950
	8	10.690	14.544	18.399	22.973	27.659	32.345	37.031	41.717	46.403	51.089
	9	11.045	14.733	18.420	23.119	27.971	32.824	37.677	42.530	47.383	52.236
	10	11.401	14.927	18.444	23.261	28.280	33.299	38.318	43.337	48.356	53.375

Figure H-65. IBM 1440 Divide Times (Based on Divide Subroutine)

1440 Divide Times With Special Feature (milliseconds)											
Number of Positions in Quotient →		1	2	3	4	5	6	7	8	9	10
Number of Positions in Divisor	1	.266	.433	.599	.766	.932	1.099	1.265	1.432	1.598	1.765
	2	.345	.589	.833	1.077	1.321	1.565	1.809	2.054	2.298	2.542
	3	.423	.744	1.065	1.387	1.708	2.029	2.351	2.672	2.993	3.315
	4	.500	.899	1.299	1.698	2.098	2.497	2.897	3.296	3.696	4.095
	5	.577	1.055	1.532	2.010	2.488	2.965	3.443	3.921	4.398	4.876
	6	.655	1.210	1.765	2.320	2.875	3.430	3.984	4.539	5.094	5.649
	7	.732	1.366	1.999	2.632	3.265	3.898	4.531	5.164	5.797	6.430
	8	.811	1.521	2.231	2.941	3.652	4.362	5.072	5.782	6.493	7.203
	9	.888	1.676	2.464	3.252	4.039	4.828	5.615	6.404	7.191	7.980
	10	.966	1.832	2.697	3.563	4.428	5.294	6.160	7.025	7.891	8.756

Figure H-66. IBM 1440 Divide Times (with Special Feature)

Scan Disk Feature

The scan-disk special feature provides an automatic search of 1311 and/or 1301 (Models 11, 12, 21, 22) disk data for a specific identifier or condition predetermined by the program.

Instructions applying to the 1301 and 1311 cannot be successfully chained.

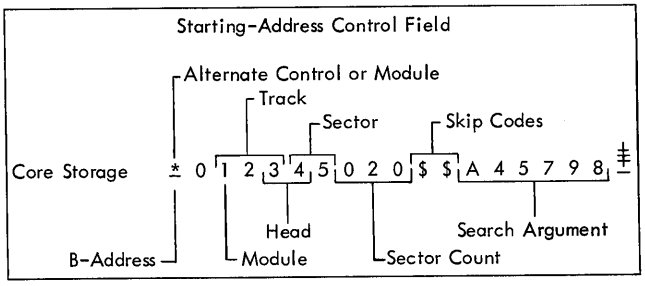


Figure H-67. Record in Core Storage for Scan Disk Operation

Scan Disk

Instruction Format.

Mnemonic	*Op Code*	*A-Address*	*B-address*	*d-character*
SDL	M	%F7	BBB	W
SDLW	L	%F7	BBB	W
SDE	M	%F8	BBB	W
SDEW	L	%F8	BBB	W
SDH	M	%F9	BBB	W
SDHW	L	%F9	BBB	W

Function. This instruction compares a specified search argument in core storage (factor B) to the records within a specified group of sectors in disk storage (factor A).

The A-address units position controls the operation. A 7 in the units position specifies a scan operation that stops when the search argument in core storage is either less than (B<A), or equal to (B=A), a record in the specified section of disk storage. An 8 specifies a scan operation that stops when the search argument in core storage is equal to (B=A) a record in the specified section of disk storage. A 9 specifies a scan operation that stops when the search argument in core storage is either higher than (B>A), or equal to (B=A), a record in the specified section of disk storage. (The operation also stops when the end of the cylinder is reached, or when the sector count reaches zero.)

The B-address of the instruction specifies the high-order position of the disk-control field in core storage that specifies the starting address in disk storage. The record area associated with the disk control field contains the search argument. The search argument must be placed in the same positions of the core-storage record as it appears in the disk-storage record. Skip codes ($) are used in those positions of the core-storage record that are not a part of the search argument (Figure H-67). The search argument can be variable in length, but must be no longer than 99 characters. The last character (100th) of the record cannot be included as part of the search argument. The units positions of the search argument should be followed by a group mark with a word mark to signal the end of the search argument.

Scanning begins at the disk record specified by the B-address and ends:

1. When the specified comparison is found. The sector-count field may, or may not, be all zeros at this time.
2. When the operation reaches the end of a cylinder. The sector-count field may, or may not, be all zeros at this time.
3. When the sector-count field is reduced to all zeros.

Word Marks. A group mark with a word mark must be set one position to the right of the last character of the search argument.

Timing. T = .0999 + $2N_S$ + disk rotation. 400 ms is the maximum time for scanning one 1311 cylinder (200 sectors), 1,332 ms is the maximum time for scanning one 1301 cylinder (800 sectors).

Notes.

1. The result of the scan is determined by testing the high, low, or equal compare indicators with the BRANCH IF INDICATOR ON instruction.
2. The scan operation can be performed only on disk records written in sector format.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B+6	B+11+L_F

L_F = length of factor B.

Example. Scan disk storage for an equal compare beginning at sector-address 012510 and continue scanning until the record with part number A24537 is found. The disk-control field is located in the high-order positions of the area of core storage labeled SCANAR (0966-0974); Figure H-68.

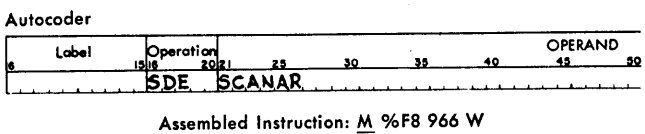


Figure H-68. Scan Disk Equal

Seek Overlap Feature (1311 Models 1 and 2)

The Seek Overlap Feature provides the flexibility to allow a seek operation to be overlapped with one 1311 read or write operation, plus any number of other seek operations on the other 1311 Drives.

Note: The Seek Overlap function is standard on all 1301 Model 11, 12, 21 or 22's that are attached to the 1440.

On the systems which have 1301's and the 1311, under certain conditions, the Seek Overlap function may be obtained by proper programming, even when the Seek Overlap Feature is not installed on the 1311's. Thus, if the Seek instructions are issued for the 1301's before they are given for the 1311, then we effectively have Seek Overlap.

Sense Switches

Sense switch A (last-card test) is standard on the 1440 system. Sense Switches B through G are available as a special feature. These manually modify programs to follow alternate routines, depending upon varying conditions or requirements. See the *Branch if Indicator On* section, and Figure B-15.

Track Record (1301)

The track record special feature provides for reading or writing an entire disk track with or without the track address. A single, 6-digit address is used, followed by 2,543 characters in the move mode and 2,261 characters in the load mode. Track records can be used for storing programs, tables, blocked records, and other data requiring a single large storage block.

When this feature is installed on the system, it provides the track-record function to all the attached 1301 units.

Instructions applying to the 1301 cannot be successfully chained.

Read Disk-Track Record

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDTR	M	%F2	BBB	R
RDTRW (word marks)	L	%F2	BBB	R

Function. This instruction causes data to be read from a disk track into core storage. The digit 2 in the A-address (%F2) specifies that a track-record operation is to be performed. Data is read from the disk track (2,543 characters in *move* mode or 2,261 characters in *load* mode). The additional characters read are accounted for by using the normal gap between disk sectors and the sector-address positions. Reading from the disk is stopped by a group mark with a word mark in core storage.

Reading from the track begins following the address specified by the core-sector address. This address is located at the beginning of the track, directly after the index pulse.

The core-sector address field in core storage is not modified, but the sector-count field in core storage is reduced by one as the track is read. The sector-count field must be set at 001 before the operation begins so that reducing it by one can signal an end-of-operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage reserved for data read from the disk track.

The R in the d-character position signifies that this is a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the track record. If a group mark with a word mark is detected before reading of the track is completed, the wrong-length record and any-disk condition indicators turn on and reading stops. The position of the group mark with a word mark is determined by adding 2,554 to the B-address.

Timing. T = 33.3999 ms + disk rotation.*

*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Notes. Track-record read operations can be performed only on a track written with a track record operation.

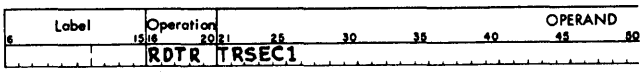
Before reading starts, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators are turned on, and the data in storage cannot be read from the disk.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + 2543 or B + 11 + 2261

Example. Read disk track 012540 in core storage beginning at location 0976 (area is labeled TRSEC1). The high-order position of the disk-control field is located in the ten positions preceding the label (0966-0975); Figure H-69.

Autocoder



Assembled Instruction: M %F2 966 R

Figure H-69. Read Disk-Track Record

Read Disk-Track Record with Address

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDTA	M	%F@	BBB	R
RDTAW (word marks)	L	%F@	BBB	R

Function. This instruction is similar to the READ DISK-TRACK RECORD instruction except that the @ in the A-address (%F@) specifies that the address of the track record in disk storage is also read into core storage with the data on the disk track. Data is read from the disk track (2,549 characters in *move* mode or 2,267 characters in *load* mode). The additional characters read are accounted for by using the normal gap between disk sectors and the sector-address positions. Reading from the disk is stopped by a group mark with a word mark in core storage.

When a disk-track record operation is initiated, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are equal, reading begins immediately following the index pulse on the disk track. (The index pulse signals the system that the beginning of a track is about to come under the access assembly.) The track-record address in the high-order position of the disk data field in core storage is written in the first sector-address position after the index pulse.

The core-sector address field in core storage is not modified, but the sector-count field in core storage is reduced by one as the track is read. The sector-count field must be set at 001 before the operation begins, so that reducing it by one can signal an end-of-operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage reserved for the track-record address and data read from the disk track.

The R in the d-character position signifies that this is a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the disk track. If a group mark with a word mark is detected before reading of the track is completed, the wrong-length record and any-disk condition indicators turn on and reading stops. The position of the group mark with a word mark is determined by adding 2560 to the B-address.

Timing. T = 33.3999 ms + disk rotation.*

*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Notes. Track-record read operations can be performed only on a track written with a track-record instruction.

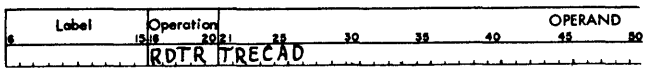
Before reading starts, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators are turned on, and the data in storage cannot be read from the disk.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2549 or B + 11 + 2267

Example. Read the address and data from disk track 012540 into core storage beginning at location 0476 (area is labeled TRECAD). The high-order position of the disk-control field is located in the ten positions preceding the label (0466-0475); Figure H-70.

Autocoder



Assembled Instruction: M %F@ 466 R

Figure H-70. Read Disk-Track Record with Address

Write Disk-Track Record

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDTR	M	%F2	BBB	W
WDTRW (word marks)	L	%F2	BBB	W

Function. This instruction causes data from core storage to be written on a disk track. The digit 2 in the A-address (%F2) specifies that a track-record operation is to be performed. An entire disk track is written from the data in core storage (2,543 characters in *move* mode or 2,261 characters in *load* mode). The additional characters are accounted for by writing in what is normally the gap between disk sectors and the sector address positions. Writing of the disk track is stopped by sensing a group mark with a word mark in core storage and the end of track.

Writing begins at the track address specified by the core-sector address field. This address is located at the beginning of the track, directly after the index pulse.

The core-sector address field in storage is not modified, but the sector-count field in core storage is reduced by one as the track is written. Set the sector-count field to 001 before the operation begins so that reducing it by one can signal an end-of-operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage where the data to be written on the disk track is stored.

The W in the d-character position signifies that this is a write operation.

Word Marks. A group mark with a word mark must be one position to the right of the last character of the data in core storage. The writing of data stops when the end of track is reached on the disk and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end of track, the remainder of the disk track is filled with valid blanks (C-bits), and the wrong-length record and any-disk condition indicators are turned on. The position of the group mark with a word mark is determined by adding 2554 to the B-address.

Timing. T = 33.3999 ms + disk rotation.*

*35 ms is maximum for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Notes.

1. Before writing starts, an automatic check is made of the core-sector address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators are turned on, and the data in storage cannot be written on the disk.
2. A WRITE DISK CHECK instruction must be performed following a write operation unless an error occurred during the write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.
3. If the data in core storage contains characters with word marks and the write operation is performed in the move mode, only the CBA 8421 portion of the character is written on the disk (the word mark is ignored).
4. Disk tracks adjacent to, but not above or below, a disk track written with the WRITE DISK-TRACK RECORD instruction must be either unused or set up as a *track record.* Tracks five and seven are adjacent to track six; track six (on disk surfaces 1 and 3) is "above or below" track six on disk surface 2. If the adjacent tracks are written using WRITE DISK SECTOR or WRITE DISK SECTOR WITH ADDRESSES instructions, interference occurs to the track-record data stored in what is normally the gap between sectors.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + 2543 or B + 11 + 2261

Example. Write a disk-track record from the data in the core-storage area labeled TRSEC1 (the first position of data is at 0976). The high-order position of the disk-control field is located in the ten positions preceding the label (0966-0975); Figure H-71.

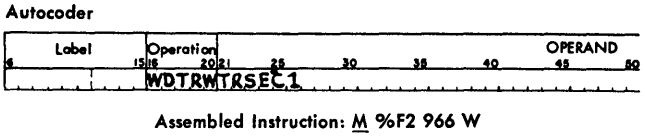


Figure H-71. Write Disk-Track Record

Write Disk-Track Record with Address

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDTA	M	%F@	BBB	W
WDTAW (word marks)	L	%F@	BBB	W

Function. This is similar to the WRITE DISK TRACK RECORD instruction except that the @ in the A-address (%F@) specifies that the address of the track record in core storage is also written on the disk.

When a disk-track record operation is initiated, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are equal, writing begins immediately following the index pulse on the disk track. (The index pulse signals the system that the beginning of a track is about to come under the access assembly.) The track-record address in the high-order position of the disk data field is written in the first sector-address position after the index pulse.

An entire disk track is written from the data in core storage (2,549 characters in *move* mode or 2,267 characters in *load* mode). The additional characters are accounted for by writing in what is normally the gap between disk sectors and the sector-address positions. Writing of the disk track is stopped by sensing a group mark with a word mark in core storage and the end of track.

The core-sector address field in storage is not modified, but the sector-count field in core storage is reduced by one as the track is written. The sector-count field should be set to 001 before the operation

begins, so that reducing it by one can signal an end-of-operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field. It also specifies the area in storage where the address and data to be written on the disk track are stored.

The W in the d-character position signifies that this is a write operation.

Word Marks. A group mark with a word mark must be one position to the right of the last character of data in core storage. The writing of data stops when the end of track is reached on the disk and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end-of-track, the remainder of the disk track is filled with valid blanks (C-bits), and the wrong-length record, and any-disk condition indicators are turned on. Processing is interlocked until the end of the sector. The position of the group mark with a word mark is determined by adding 2560 to the B-address.

Timing. T = 33.3999 ms + disk rotation.*

*35 ms is maximum time for disk rotation.
18.4 ms is average time for disk rotation.
1.7 ms is minimum time for disk rotation.

Notes.

1. Before writing starts, an automatic check is made of the core-sector address in storage with one of the sector addresses on the pack. If the address is not found, the unequal-address compare and any-disk condition indicators are turned on, and the data in storage cannot be written on the disk.
2. A WRITE DISK CHECK instruction must be performed following a write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.
3. If the data in core storage contains characters with word marks and the write operation is performed in the move mode, only the CBA 8421 portion of the character is written on the disk (the word mark is ignored).
4. Disk tracks adjacent to, but not above or below, a disk track written with the WRITE DISK TRACK RECORD or WRITE DISK TRACK RECORD WITH ADDRESS instructions must be either unused or set up as a *track record*. Tracks five and seven are adjacent to track six; track six (on disk surfaces 1 and 3) is "above or below" track six on disk surface 2. If the adjacent tracks are written using WRITE DISK SECTOR or WRITE DISK SECTOR with ADDRESSES instruction, interference occurs to the track-record data stored in what is normally the gap between sectors.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2549 or B + 11 + 2267

Example. Write a disk track record with its new address from the data in the core-storage area labeled TRECAD (the first position of the address is at 0476). The high-order position of the disk-control field is located in the ten positions preceding the label (0466-0475); Figure H-72.

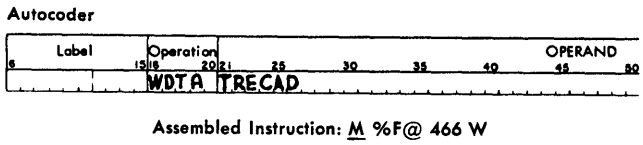


Figure H-72. Write Disk-Track Record with Address

Track Record (1311)

The track record special feature provides for reading or writing an entire disk track with or without the track address. A single, 6-digit address is used, followed by 2,980 characters in the *move* mode and 2,682 characters in the *load* mode. Track records can be used for storing programs, tables, blocked records, and other data requiring a single large storage block.

When this feature is installed on the system, it provides the track-record function to all the attached 1311 drives.

Instructions applying to the 1311 cannot be successfully chained.

Read Disk-Track Record

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDTR	M	%F2	BBB	R
RDTRW (word marks)	L	%F2	BBB	R

Function. This instruction causes data to be read from a disk track into core storage. The digit 2 in the A-address (%F2) specifies that a track-record operation is to be performed. Data is read from the disk track (2,980 characters in *move* mode or 2,682 characters in *load* mode). The additional characters read are accounted for by using the normal gap between disk sectors and the sector-address positions. Reading from the disk is stopped by a group mark with a word mark in core storage. Reading from the track begins following the address specified by the core-sector address. This address is located at the beginning of the track, directly after the index pulse.

The core-sector address field in core storage is not modified, but the sector-count field in core storage is reduced by one as the track is read. The sector-count field must be set at 001 before the operation begins so that reducing it by one can signal an end of operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field, and the area in storage reserved for data read from the disk track.

The R in the d-character position signifies that this is a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the track record. If a group mark with a word mark is detected before reading of the track is completed, the wrong-length record and any-disk condition indicators turn on and reading stops. The position of the group mark with a word mark is determined by adding 2,991 (*move* mode) or 2,693 (*load* mode) to the B-address.

Timing. T = 40.0999 ms + disk rotation.*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

Notes.

1. Track-record read operations can be performed only on a track written with a track-record instruction.
2. Before reading starts, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators turn on, and the data in storage cannot be read from the disk.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + 2980 or B + 11 + 2682

Example. Read disk track 012540 in core storage beginning at location 0976 (area is labeled TRSEC1). The high-order position of the disk-control field is located in the ten positions preceding the label (0966-0975); Figure H-73.

Autocoder

```
Label          Operation          OPERAND
6          15  16  20 21  25  30  35  40  45  50
               RDTR  TRSEC1
```

Assembled Instruction: M %F2 966 R

Figure H-73. Read Disk-Track Record

Read Disk-Track Record with Address

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RDTA	M	%F@	BBB	R
RDTAW (word marks)	L	%F@	BBB	R

Function. This instruction is similar to the READ DISK-TRACK RECORD instruction except that the @ in the A-address (%F@) specifies that the address of the track record in disk storage is also read into core storage with the data on the disk track. Data is read from the disk track (2,986 characters in *move* mode or 2,688 characters in *load* mode). The addi-

tional characters read are accounted for by using the normal gap between disk sectors and the sector-address positions. Reading from the disk is stopped by a group mark with a word mark in core storage.

When a disk-track record operation is initiated, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are equal, reading begins immediately following the index pulse on the disk track. (The index pulse signals the system that the beginning of a track is about to come under the access assembly). The track-record address in the high-order position of the disk data field in core storage is written in the first sector-address position after the index pulse.

The core-sector address field in core storage is not modified, but the sector-count field in core storage is reduced by one as the track is read. The sector-count field must be set at 001 before the operation begins so that reducing it by one can signal an end of operation (000 in sector-count field).

The B-address specifies the high-order position in core storage reserved for the track-record address and data read from the disk track.

The R in the d-character position signifies that this is a read operation.

Word Marks. A group mark with a word mark must be one position to the right of the last position reserved in core storage for the disk track. If a group-mark with a word mark is detected before reading of the track is completed, the wrong-length record and any-disk condition indicators turn on and reading stops. The position of the group-mark with a word mark is determined by adding 2,997 (*move* mode) or 2,699 (*load* mode) to the B-address.

Timing. T = 40.0999 ms + disk rotation.*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

Notes.

1. Track-record read operations can be performed only on a track written with a track-record instruction.
2. Before reading starts, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators turn on, and the data in storage cannot be read from the disk.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2986 or B + 11 + 2688

Example. Read the address and data from disk track 012540 into core storage beginning at location 0476 (area is labeled TRECAD). The high-order position of the disk-control field is located in the ten positions preceding the label (0466-0475); Figure H-74.

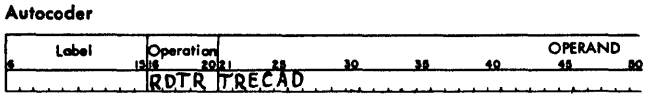


Figure H-74. Read Disk-Track with Address

Write Disk-Track Record

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDTR	M	%F2	BBB	W
WDTRW (word marks)	L	%F2	BBB	W

Function. This instruction causes data from core storage to be written on a disk track. The digit 2 in the A-address (%F2) specifies that a track-record operation is to be performed. An entire disk track is written from the data in core storage (2,980 characters in *move* mode or 2,682 in *load* mode). The additional characters are accounted for by writing in what is normally the gap between disk sectors and the sector-address positions. Writing of the disk track is stopped by sensing a group mark with a word mark in core storage and the end of track.

Writing begins at the track address specified by the core sector address field. This address is located at the beginning of the track, directly after the index pulse.

The core-sector address field in storage is not modified, but the sector-count field in core storage is reduced by one as the track is written. Set the sector-count field to 001 before the operation begins so that reducing it by one can signal an end of operation (000 in sector-count field).

The B-address specifies the high-order position in core storage of the disk control field, and the area in storage where the data to be written on the disk track is stored.

The W in the d-character position signifies that this is a write operation.

Word Marks. A group mark with a word mark must be one position to the right of the last character of the data in core storage. The writing of data stops when the end of track is reached on the disk, and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end of track, the remainder of the disk track is filled with C-bits (blanks), and the wrong-length record and any-disk condition indicators turn on. The position of the group mark with a word mark is determined by adding 2,991 (*move* mode) or 2,693 (*load* mode) to the B-address.

Timing. T = 40.0999 ms + disk rotation.*

*42 ms is maximum time for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

Notes.

1. Before writing starts, an automatic check is made of the core-sector address in storage with the record address on the disk. If the addresses are not the same, the unequal-address compare and any-disk condition indicators turn on, and the data in storage cannot be written on the disk.
2. A WRITE DISK CHECK instruction must be performed following a write operation unless an error occurred during the write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.
3. If the data in core storage contains characters with word marks and the write operation is performed in the *move* mode, only the CBA 8421 portion of the character is written on the disk (the word mark is ignored).
4. Disk tracks adjacent to, but not above or below, a disk track written with the WRITE DISK-TRACK RECORD instruction must be either unused or set up as a *track record.* Tracks five and seven are adjacent to track six; track six (on disk surfaces 1 and 3) is "above or below" track six on disk surface 2.

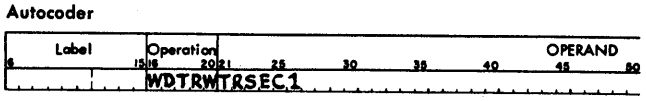
   If the adjacent tracks are written using WRITE DISK SECTOR or WRITE DISK SECTOR WITH ADDRESS instructions, interference occurs to the track-record data stored in what is normally the gap between sectors.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 6	B + 11 + 2980 or B + 11 + 2682

Example. Write a disk-track record from the data in the core-storage area labeled TRSEC1 (the first position of data is at 0976). The high-order position of the disk-control field is located in the ten positions preceding the label (0966-0975); Figure H-75.

Autocoder



Assembled Instruction: M %F2 966 W

Figure H-75. Write Disk-Track Record

Write Disk-Track Record with Address

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
WDTA	M	%F@	BBB	W
WDTAW (word marks)	L	%F@	BBB	W

Function. This instruction is similar to the WRITE DISK-TRACK RECORD instruction except that the @ in the A-address (%F@) specifies that the address of the track record in core storage is also written on the disk.

When a disk-track record operation is initiated, an automatic check is made of the record address in storage with the record address on the disk. If the addresses are equal, writing begins immediately following the index pulse on the disk track. (The index pulse signals the system that the beginning of a track is about to come under the access assembly.) The track-record address in the high-order position of the disk data field in core storage is written in the first sector-address position after the index pulse.

An entire disk track is written from the data in core storage (2,986 characters in *move* mode or 2,688 in *load* mode). The additional characters are accounted for by writing in what is normally the gap between disk sectors and the sector-address positions. Writing of the disk track is stopped by sensing a group mark with a word mark in core storage and the end of track.

The core-sector address field in storage is not modified, but the sector-count field in core storage is reduced by one as the track is written. The sector-count field should be set to 001 before the operation begins, so that reducing it by one can signal an end of operation (000 in the sector-count field).

The B-address specifies the high-order position in core storage of the disk-control field. It also specifies the area in storage where the address and data to be written on the disk track are stored.

The W in the d-character position signifies that this is a write operation.

Word Marks. A group mark with a word mark must be one position to the right of the last character of data in core storage. The writing of data stops when the end of track is reached on the disk and a group mark with a word mark is sensed in core storage. If the group mark with a word mark is sensed before the end of track, the remainder of the disk track is erased. Because even a valid blank must have a C-bit, a parity error condition results. The disk

error, wrong-length record, and any-disk condition indicators are turned on. Processing is interlocked until the end of the sector. The position of the group mark with a word mark is determined by adding 2,997 (*move* mode) or 2,699 (*load* mode) to the B-address.

Timing. T = 40.0999 ms + disk rotation.*

*42 ms is maximum for disk rotation.
22 ms is average time for disk rotation.
2 ms is minimum time for disk rotation.

Notes.

1. Before writing starts, an automatic check is made of the core-sector address in storage with sector address on the pack. If the address is not found, the unequal-address compare and any-disk condition indicators turn on, and the data in storage cannot be written on the disk.
2. A WRITE DISK CHECK instruction must be performed following a write operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.
3. If the data in core storage contains characters with word marks and the write operation is performed in the *move* mode, only the CBA 8421 portion of the character is written on the disk (the word mark is ignored).
4. Disk tracks adjacent to, but not above or below, a disk track written with the WRITE DISK-TRACK RECORD or WRITE DISK-TRACK RECORD WITH ADDRESS instructions must be either unused or set up as a track record. Tracks five and seven are adjacent to track six; track six (on disk surfaces 1 and 3) is "above or below" track six on disk surface 2. If the adjacent tracks are written using WRITE DISK SECTOR or WRITE DISK SECTOR WITH ADDRESSES instructions, interference occurs to the track-record data stored in what is normally the gap between sectors.

   The write-address key on disk-storage-drive zero must be on to perform this operation.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	B + 9	B + 11 + 2986 or B + 11 + 2688

Example. Write a disk-track record with its new address from the data in the core storage area labeled TRECAD (the first position of the address is at 0476). The high-order position of the disk-control field is located in the ten positions preceding the label (0466-0475); Figure H-76.

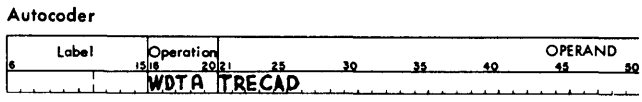


Figure H-76. Write Disk-Track Record with Address

Baudot Character	Enters Storage as:	Generates Translate-Table Address	Baudot Character	Enters Storage as:	Generates Translate-Table Address
T	2	X02	Z	CB2	X42
5 (FIGS-T)	C21	X03	"(FIGS-Z)	B21	X43
CR	4	X04	D	CB4	X44
CR (FIGS)	C41	X05	$(FIGS-D)	B41	X45
0	C42	X06	B	B42	X46
9 (FIGS-0)	421	X07	?(FIGS-B)	CB421	X47
SPACE	8	X10	S	CB8	X50
FIGS-SPACE	C81	X11	Bell (FIGS-S)	B81	X51
H	C82	X12	Y	B82	X52
#(FIGS-H)	821	X13	6(FIGS-Y)	CB821	X53
N	C84	X14	F	B84	X54
,(FIGS-N)	841	X15	!(FIGS-F)	CB841	X55
M	842	X16	X	CB842	X56
.(FIGS-M)	C8421	X17	/(FIGS-X)	B8421	X57
LF	A	X20	A	CBA	X60
FIGS-LF	CA1	X21	-(FIGS-A)	BA1	X61
L	CA2	X22	W	BA2	X62
)(FIGS-L)	A21	X23	2(FIGS-W)	CBA21	X63
R	CA4	X24	J	BA4	X64
4 (FIGS-R)	A41	X25	'(FIGS-J)	CBA41	X65
G	A42	X26	FIGS	Deleted From Input by Adapter	
&(FIGS-G)	CA421	X27			
I	CA8	X30	U	BA8	X70
8 (FIGS-I)	A81	X31	7(FIGS-U)	CBA81	X71
P	A82	X32	Q	CBA82	X72
0 (FIGS-P)	CA821	X33	1(FIGS-Q)	BA821	X73
C	A84	X34	K	CBA84	X74
:(FIGS-C)	CA841	X35	((FIGS-K)	BA841	X75
V	CA842	X36	LTRS	Deleted From Input by Adapter	
;(FIGS-V)	A8421	X37			
E	B	X40	BLANK	C	X00
3(FIGS-E)	CB1	X41	FIGS-BLANK	1	X01

(Note: X in table address represents any hundreds-position digit.)

Figure H-77. Baudot Code to Generate Translate-Table Address

Translate

This special feature provides the data processing system with the capability of fast, flexible translation of codes to and from the code of the system.

The feature uses stored-program instructions to initiate the code translation and subsequent record movement. One translate instruction translates a complete record, moving left to right as it replaces each record character with a character from a translate table in core storage until a group mark with a word mark is detected in the field being translated.

Each code translation requires a table in storage beginning at an even-hundreds address for TRANSLATE-WITH-WORD-MARKS and at *any* hundreds address for TRANSLATE-WITHOUT-WORD-MARKS. The number of various code translations that can be handled at one time in a system is limited only by the core storage available for tables.

The LOAD RECORD instruction (included in the translate feature) moves characters and word marks from an A-field to a B-field, moving left to right up to and including an A-field group mark with a word mark. Original B-field word marks are cleared.

Instructions applying to the translate feature cannot be successfully chained.

BCD Character	Generates Translate-Table Address	BCD Character	Generates Translate-Table Address
BLANK	X00	–	X40
1	X01	J	X41
2	X02	K	X42
3	X03	L	X43
4	X04	M	X44
5	X05	N	X45
6	X06	O	X46
7	X07	P	X47
8	X10	Q	X50
9	X11	R	X51
0	X12	!	X52
#	X13	$	X53
@	X14	*	X54
:	X15	)	X55
>	X16	;	X56
√‾	X17	△	X57
¢	X20	&	X60
/	X21	A	X61
S	X22	B	X62
T	X23	C	X63
U	X24	D	X64
V	X25	E	X65
W	X26	F	X66
X	X27	G	X67
Y	X30	H	X70
Z	X31	I	X71
≠	X32	?	X72
,	X33	.	X73
%	X34	¤	X74
=	X35	(	X75
'	X36	<	X76
"	X37	‡	X77

Note: Word marks with the BCD characters will generate the same sequence of addresses at (X+1)00, (X+1)01, etc.

Figure H-78. BCD Character to Generate Translate-Table Address

Translate with Word Marks

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
TRW	T	AAA	B00	>

The TRANSLATE WITH WORD MARKS instruction consists of T (C A 2 1 WM) for the operation code, a 3-character A-address representing the initial address of the record to be translated, a 3-character B-address representing the initial address of the translate table, and a d-character with a bit configuration of 8-4-2.

The record to be translated must end with a group mark with a word mark. The initial address of the translate table is restricted to any available *even*-hundreds address such as 200, 400, 600, 800, or 1000. Two table sizes can be accommodated: a 78-character table provides as many as 64 usable positions, and a 156-character table provides as many as 128 usable positions. (Figures H-77 and H-78 show that the digits 8 and 9 are not used in the units or hundreds position of any generated address of the translate table.) The 156-character table consists of the 78-character table, beginning at an even-hundreds address (for example 200-277), combined with another 78-character table, beginning at the next sequential hundreds address (300-377).

Characters and word marks from the translate table (initial address specified by B-address) replace the characters and word marks in the record being translated, beginning at the address specified by the A-address of the translate instruction.

The TRANSLATE WITH WORD MARKS instruction interprets word marks in the A-address field as DATA or SHIFT bits, and the word marks actually take part in the translate function.

The translate instructions are interruptible.

The B-field address (initial translate-table address) cannot be indexed. The B-address register contains blanks in the units and tens position at the completion of a translate operation. Therefore, the use of a STORE B-ADDRESS REGISTER instruction immediately following a translate operation causes the system to interlock.

Translate without Word Marks

Mnemonic	*Op Code*	*A-address*	*B-address*
TR	T	AAA	B00

The TRANSLATE WITHOUT WORD MARKS instruction format is the asme as the TRANSLATE WITH WORD MARKS instruction, but without a d-character. It functions the same, with these exceptions:

1. One table size can be accommodated: the 78-character table, which provides as many as 64 usable positions.
2. A-field word marks do not take part in the actual translation, and are *not* altered by the translation.

General Description of Translate

The program assigns locations and contents in the translate table depending on the desired translation. Each position of the translate table contains the BCD character to which a particular character is to be translated. For each character to be translated, the translate feature automatically selects the appropriate position of the translate table. The contents of that position replaces the character in the A-field. The contents of the translate table are undisturbed.

The translate-table characters must contain odd parity. The input/output device, channel or adapter (for example, the IBM 1448 Transmission Control Unit) performs any necessary parity conversion for input and output. The tables also must contain required SHIFT bits for shifted codes. The adapter detects SHIFT-bit transitions, generates the required SHIFT character, and removes the SHIFT bit.

Figure H-77 is an example of how a code (Baudot telegraph code in this case) enters the system and the addresses that are generated (assuming Baudot 1-2-3 4-5 bits respectively are equal to BCD bits B-A-8-4-2 with the 1-bit designating figures shift.

Figure H-78 shows the BCD character and the table address generated.

Translation of each character is accomplished in a 3-cycle sequence (first A-cycle, B-cycle, and second A-cycle). Refer to Figure H-79 while reading the sequence described here. Here again Baudot code is used as an example.

Although the examples show the function of the translate with incoming data, it functions the same way for outgoing data using another translate table.

First A-Cycle

The first A-cycle generates the appropriate translate-table address specified by the character to be translated.

The translate instruction acts upon all the characters of the A-field. Figure H-79 shows how the instruction T 900 600 (as an example) translates one character. In this case the first character of the A-field is used as the example. The contents of position 900 has a bit configuration of 2-4-A (Baudot-code G).

The first A-cycle moves this first character in the A-field to the B-register. From the B-register, the operation moves the 1-2-4 bits of this character (with proper parity) to the *units* position of the storage-address register together with the A-B bits from the units position of the B-address register. A-B bits in the units position of the generated address designate core-storage blocks over 3999. In Figure H-79 the 2-4 bits produce a 6 in the *units* position of the storage-address register. If the 1-2-4 bits are all blanks (no-bits), the operation generates an 8-2-C bit configuration (zero) into the *units* position of the storage-address register.

The first A-cycle also moves the 8-A-B bits (interpreted as 1-2-4 bits) of the same character (with proper parity) to the *tens* position of the storage-address register. In Figure H-79 the A-bit, which becomes a 2-bit, produces a 2 in the *tens* position of the storage-address register. Again, if the 8-A-B bits are all blanks, the operation generates an 8-2-C bit configuration (zero) into the *tens* position of the storage-address register.

If the instruction is a TRANSLATE WITH WORD MARKS, a word-mark bit in the B-register is interpreted as a 1-bit and is combined with the bits already selected for transfer to the *hundreds* position of the storage-address register. This adds 1 to the *hundreds* position of the storage-address register and thus generates an address from the second half of a 156-character table. If the table's base address was 600 (as in Figure H-79) the generated address would be 726. Because in the example the B-register has no word mark, the generated address is 626.

A TRANSLATE WITHOUT WORD MARKS instruction blocks a word mark in the B-register so it takes no part in the generation of the address.

B-Cycle

The B-cycle of the operation reads a character or character with word marks out of a specified translate-table address in storage into the A-register. It uses the address generated in the first A-cycle. The translate table is unaltered.

Second A-Cycle

The second A-cycle of the operation again reads out the character being translated and replaces it in core

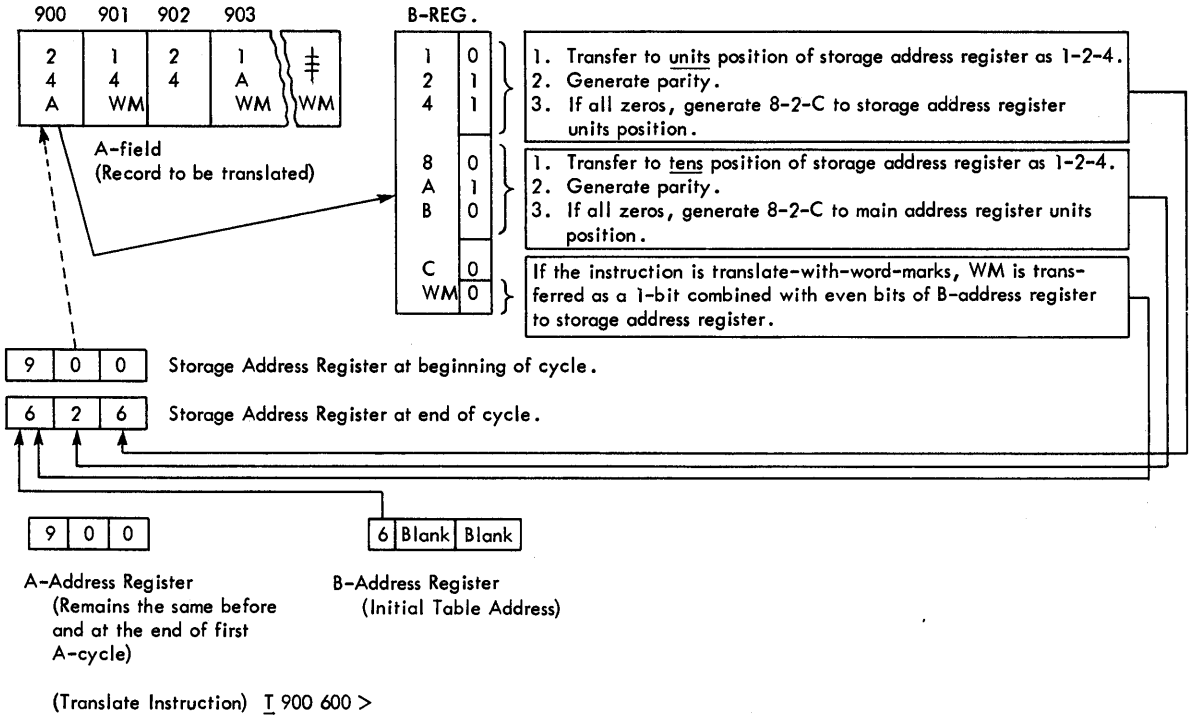


Figure H-79. First A-Cycle of Translate Sequence

storage with the contents of the A-register, which contains the character read out of the translate table on the B-cycle. The original A-field character is destroyed.

A TRANSLATE WITH WORD MARKS instruction destroys original A-field word marks but moves word marks from the translate table to the A-field. A TRANSLATE WITHOUT WORD MARKS instruction regenerates original A-field word marks into the A-field and moves translate-table word marks to the A-field.

Figures H-80 and H-81 show how the translate instruction affects processor-storage areas. No particular codes are specified.

Timing.

The formula for the translate-operation execution time (T) for the 1440 is:

$$T = .0111\ (L_I + 2 + 3N)\ \text{ms.}$$

N = The number of characters in A-field to be translated.
L_I = 7 for TRANSLATE WITHOUT WORD MARKS.
8 for TRANSLATE WITH WORD MARKS.

Example: N = 100-character record
T = 1.8 ms

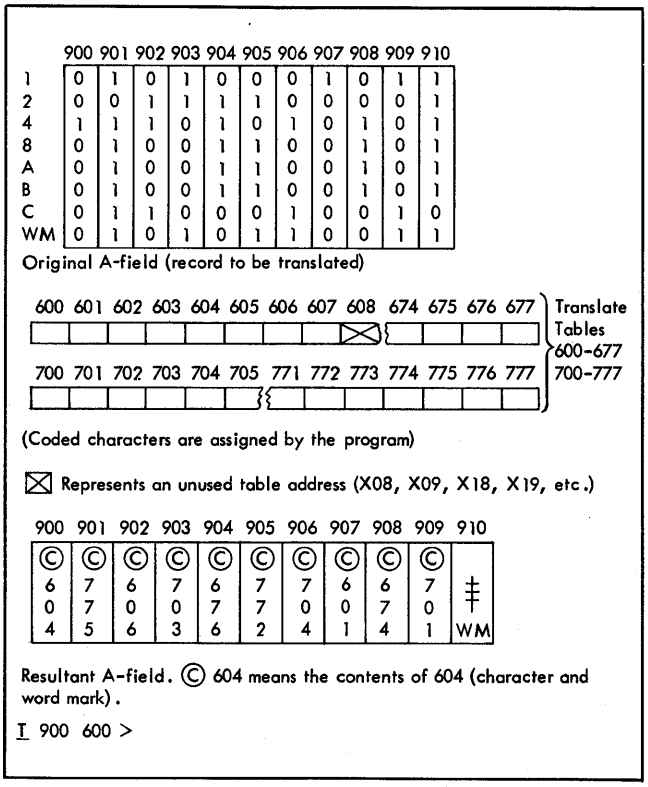


Figure H-80. Translate with Word Marks

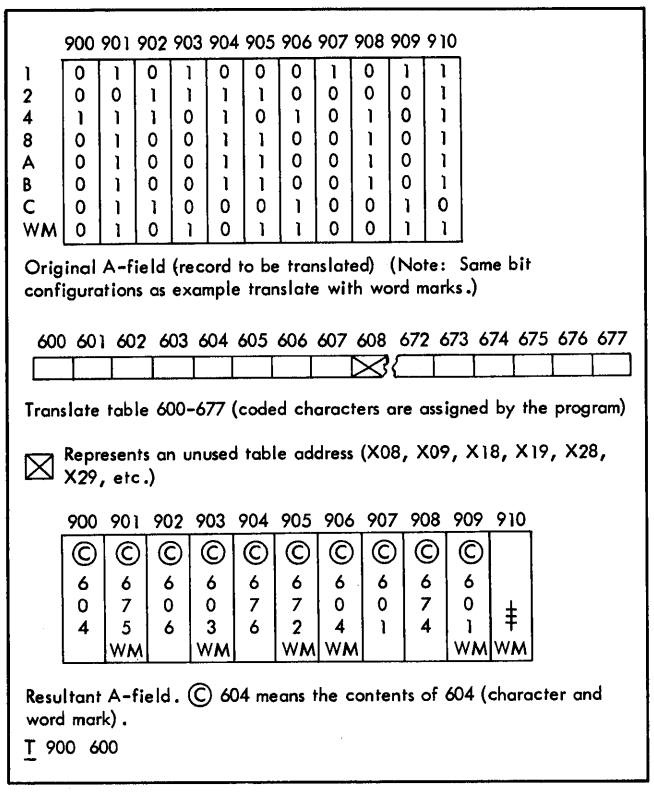


Figure H-81. Translate without Word Marks

Load Record

The translate feature includes a special move instruction that moves characters and associated word marks in a record from one storage area to another.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MRCWG	P	AAA	BBB	>

The LOAD RECORD instruction consists of P (B 4 2 1 WM) for the operation code, a 3-character A-address representing the address of the record to be moved, a 3-character B-address representing the address of the field to which the record is moved, and a d-character with a bit configuration of 8-4-2.

The operation moves characters and word marks in the A-field to the B-field, moving from low-numbered to high-numbered storage position up to and including an A-field group mark with a word mark, which stops the operation. Original B-field word marks are cleared.

IBM 1009 Data Transmission Unit Special Features

IBM 1009 Data Transmission Control Unit Buffer Feature

Substantially improved 1009 operation is provided by this buffer feature. Refer to *IBM 1009 Data Transmission Unit,* Form A24-1039, and to the *IBM 1009 Data Transmission Unit* section of this publication. While the buffer is being loaded with data from the transmission line, or unloading data to the transmission line, the processor is freed for other operations.

The buffer feature provides a 400-character core-storage buffer that is divided into four 100-character blocks. Blocks of 100 characters are transferred between buffer storage and the processor by a single MOVE or LOAD instruction. After a block of 100 characters has been moved into the core-storage I/O area, the processor (receive mode) interrogates indicators to determine whether more data is stored and available for transfer to the processor.

After a block of 100 characters has been moved into the 1009 buffer from the processor core storage (transmit mode), the program routine can interrogate the buffer to determine whether it can accommodate another block of 100 characters. If additional data is available in the 1009 buffer (receive mode), or additional space is available in the 1009 buffer (transmit mode), the program routine can initiate another MOVE or LOAD instruction. The B-address of the LOAD or MOVE instruction should be increased by 100 before the instruction is executed. This procedure is followed until a group mark with word mark (end-of-message) is detected in the processor core storage (transmit mode) or an end-of-message is detected in 1009 buffer storage (receive mode).

The processor program must determine the validity of the data received by testing the appropriate indicator. The data is processed if it is valid. If the data is invalid, it is automatically retransmitted a maximum of two more times. A counter in the 1009 buffer keeps track of the number of transmissions made during an error routine; however, the retransmission of data is under program control. If the data is still invalid after three transmissions, the transmission of data ceases, and an alarm sounds indicating operator intervention is required.

The buffer feature also provides the 1009 with the ability to answer automatically and establish a telephone connection for transmission of data and disconnect at the end of transmission without operator intervention. The direction of the transmission can automatically controlled through appropriate programming.

Console Panel, IBM 1009 Buffer Feature

The console panel remains the same except for the transmit/receive switch. This switch now has three positions: transmit, receive, and automatic. The switch must be set to AUTOMATIC if the auto-answer, auto-disconnect, and/or automatic control of transmission direction features are used.

IBM 1009 Automatic Mode Operation

Procedure for making retransmission connection:

1. The operator places the call after loading the program in the processor and placing the 1009 in a *ready* status (power on, binary-BCD mode switch set to desired mode, test-normal switch set to NORMAL, and transmit-receive-automatic switch set to AUTOMATIC).
2. After placing the 1009 in a ready status and completing the line connection, press the data key on the data set, and cradle the handset.
3. Press the auto key on the data set if future calls are to be answered automatically.
4. Press the start key on the processor.
5. The K E instruction in the program routine sets the transmission direction to transmit.
6. The U %D1 E instruction initializes the transmission of the message.

Procedure for making the receiving connection:

1. Load the program in the processor and place the 1009 in a ready status (power on, binary-BCD mode switch set to desired mode, test-normal switch set to NORMAL, and transmit-receive-automatic switch set to AUTOMATIC).
2. Press the auto key on the data set.
3. Press the start key on the processor console.
4. The K D instruction in the program routine sets the direction of transmission to receive. The 1009 automatically answers a call and receives an initial inquiry signal from the sender. The 1009 acknowledges this inquiry under control of the U %D1 D processor instruction. After the inquiry signal is acknowledged, the first message is received in 1009 buffer storage.

Ending the Operation

The IBM 1009, under program control, automatically performs all necessary disconnect functions. When an end-of-file signal is received. the receiving station can, under program control, either reverse direction and proceed in a transmit mode, reestablish the receive mode, or ignore the end-of-file signal. Ignoring the end-of-file signal automatically disconnects the transmission line after about 30 seconds.

The receive station can also send an end-of-file signal to the transmitting station manually or under program control. The transmitting station acknowledges this signal by sending an end-of-file signal to the receive station, which terminates the transmission on both ends.

IBM 1009 Data Transmission Unit Line Speed

CPS	*Speed Select Switch Setting*	*MS/CHAR*
75	600	13.3
93.75	750	10.65
125	1000	8.0
150	1200	6.67
187.5	1500	5.7
250	2000	4.0
300	2400	3.33
375	3000	2.66
500	4000	2.0
0-600	EXT.*	LINE

*Speed depends upon type of data set used

Maximum Processor Time Required for Data Movement

Blocks of 100 characters (with or without word marks) are transferred by each MOVE or LOAD instruction.

Timing (without indexing).

T =	.0999 ms	(Instruction time)
+	.0111	(First character)
+	L	(Line-character speed in ms/character).
+	1.0989	(Subsequent 99 characters of the block. Each character requires .0111 ms. When fewer characters are transferred, reduce this factor accordingly).
	1.2099 ms	(plus line speed) for each 100-character block.

Timing (with indexing).

T =	.0999 ms	(Instruction time)
+	.0111	(First character)
+	.0333	(Indexing cycle time)
+	L	(Line-character speed in ms/character).
+	1.0989	(Subsequent 99 characters of the block. Each character requires .0111 ms. When fewer characters are transferred, reduce this factor accordingly).
	1.2432 ms	(plus line speed) for each 100-character block.

Additional and Modified Instructions for 1009 Buffer Feature

Several processor instructions are expanded to provide program control for the IBM 1009 Data Transmission Unit with buffer feature.

Instructions applying to the 1009 cannot be successfully chained.

Initialize a Message Transmission (XMIT)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%D1	E

Function. This instruction initiates a start-of-message signal if the 1009 is in a send-run condition (transmit/receive switch is set to TRANSMIT). If the 1009 is in a receive-run condition (transmit/receive switch set to RECEIVE) the instruction causes the processor to interlock and an alarm to sound, signaling that operation intervention is necessary. This instruction is also used when the transmit, receive, and automatic switch is set to the automatic position (buffer feature installed) and a K E (set direction to TRANSMIT) instruction has been issued.

The A-address specifies the 1009, and the d-character specifies the start-transmission operation.

Word Marks. Word marks are not affected.

Timing. T = .0666 ms.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	Ebb

Initialize a Reply from Receiver (RCV)

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*d-character*
CU	U	%D1	D

Function. The receiving 1009 signals the transmitting station that it is ready to receive and indicates the status of the previous message (see BRANCH IF INDICATOR ON instruction).

Word Marks. Word marks are not affected.

Timing. T = .0666 ms.

Note: The transmit/receive switch on the receiving 1009 should be set to RECEIVE. If it is set to TRANSMIT, the processor is interlocked and an alarm is sounded to signal the operator. This instruction is also used when the transmit, receive, and automatic switch is set to the automatic position (buffer feature installed) and a K D instruction has been issued.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	Dbb

Move Character to the Transmitting 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%D1	BBB	W

Function. The transmitting processor sends 100 characters to the 1009 buffer, starting with the position in core storage specified by the B-address. The d-character, W, specifies a transmit operation.

Word Marks. Word marks are not affected.

Timing. T = 1.210 + 1L ms (without indexing)

T = 1.243 + 1L ms (with indexing)

Note. If a group mark with word mark is sensed in processor core storage, an end-of-message transmit condition is recognized. See *Maximum Processor Time Required for Movement of Data* section.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B + 100 or GM + 1

Move Character from the Receiving 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
MU	M	%D1	BBB	R

Function. This instruction transfers 100 characters in the receiving 1009 buffer to the receiving processor core-storage location starting with the position specified by the B-address. The d-character specifies a receive operation.

Word Marks. Word marks are not affected.

Timing. T = 1.210 + 1L ms (without indexing)

T = 1.243 + 1L ms (with indexing)

Note. When the 1009 recognizes the end-of-message condition, the receiving processor gets an end-of-message receive signal and inserts a group mark in the core-storage location immediately beyond the location containing the last character of the message. See *Maximum Process Time Required for Movement of Data.*

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B + 100 or GM + 1

Load Character to the Transmitting 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%D1	BBB	W

Function. The transmitting processor sends 100 characters, starting with the location specified by the B-address, to the transmitting 1009 buffer. The d-character, W, specifies a transmit operation.

Word Marks. If a word mark is associated with a character, it is placed with the character in 1009 buffer storage during one transfer cycle. When the character is placed on the transmission line, the word mark is converted to a word separator. Placing the character and word separator on the transmission line takes two transmission cycles.

Timing. T = 1.210 + 1L ms (without indexing)

T = 1.243 + 1L ms (with indexing)

Note. A group mark with word mark in processor core storage signals an end-of-message transmit condition. See *Maximum Process Time Required for Movement of Data.*

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	%41	B + 100 or GM + 1

Load Character from the Receiving 1009

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
LU	L	%D1	BBB	R

Function. This instruction transfers 100 characters in the receiving 1009 buffer to receiving processor core storage starting with the location specified by the B-address. The d-character, R, signals a receive operation.

Word Marks. If a word mark is associated with a character, it is received as a word separator followed by its associated data character, but converted to a word mark and placed with its associated character in the receiving 1009 buffer storage. This operation takes two transmission cycles. The character and associated word mark are then transferred to processor core storage during one transfer cycle.

Timing. T = 1.210 + 1L ms (without indexing)

T = 1.243 + 1L ms (with indexing)

Note. When the 1009 recognizes an end-of-message condition, the receiving processor interprets an end-of-message receive signal and inserts a group mark in the core-storage location immediately beyond the location containing the last character of the message. See *Maximum Process Time Required for Movement of Data.*

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	B-*Add. Reg.*
NSI	%41	B + 100 or GM + 1

Branch if Indicator On

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	d

Function. This instruction tests the indicator specified by the d-character. If the indicator is on, the program branches to the I-address for the next instruction. If it is off, the program continues with the next sequential instruction.

d-character	*Indicator*	*Station*
1	1009 Ready	RCV or XMIT
2	Buffer Service	RCV or XMIT
3	Reply Good	XMIT
4	Reply Bad	XMIT
5	Receive Error	RCV
6	Attention 1009	RCV or XMIT
7	Receive EOM or Initial Inquiry	RCV
8	Receive EOF	RCV or XMIT

Indicators

B III 1. This indicator turns on when the 1009 is in a run condition. If the 1009 is not in a run condition, the program should stop, or loop until the run condition is established.

B III 2. This indicator, when on, indicates to the processor that the 1009 is in a buffer-available condition when in either the transmit or receive mode. This indicator turns on the first time when 100 characters have been placed in the receive 1009 buffer or when a U %D1 E instruction has been issued when in a transmit mode.

B III 3. This indicator turns on if the signal sent to the transmitting station by the U %D1 D instruction specified that a good transmission occurred. The transmitting processor should test this indicator and branch to the routine for the next message if it is on. If the indicator is not on, the program should advance to test the transmission-error indicator.

B III 4. The reply-bad indicator turns on if the signal sent to the transmitting station by the U %D1 D instruction specified that a transmission error occurred. The transmitting processor should test this indicator and branch to an error subroutine if an error occurred.

B III 5. This indicator, when on, indicates to the processor that the condition for acknowledgment that has been set in the 1009 is error-reply. This indicator turns on when the first error character is detected in a message and remains on until the next U %D1 D instruction is issued.

B III 6. This indicator turns on when any one of the following indicators turns on: Branch 2, Branch 3, Branch 4, Branch 7, and Branch 8. When the Branch 6 indicator is on, the 1009 requires program attention.

B III 7. This indicator, when on, indicates to the processor that the conditions for an acknowledgment have been set (reply-good or reply-bad) in the 1009. This indicator turns on when the processor receives an end-of-message signal from the 1009 and remains on until the next U %D1 D instruction is issued. This also comes on in response to the initial two-character inquiry sequence (transmit leader/inquiry) sent by the transmitting 1009 prior to the start of transmission.

B III 8. This indicator, when on, indicates to the processor that the 1009 has received an EOF (end-of-file) signal from the remote terminal.

Word Marks. Word marks are not affected.

Timing.

No branch, or branch without indexing: T = .0666 ms.
Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch:	NSI	BI	dbb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing):	NSI	BI	NSI

Suppress 3-Second Alarm

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		A
SSB	K	III	A

Function. This instruction prevents the 3-second alarm from sounding during a delay (such as tape rewind). Normal alarm functions will be restored when any subsequent instruction addresses the 1009. This instruction can be given when a delay in processing can be foreseen.

Word Marks. Word marks are not affected.

Timing.

Suppress 3-Second Alarm: T = .0333 ms.
Suppress 3-Second Alarm and Branch (without indexing: T = .0666 ms.
Suppress 3-Second Alarm and Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Suppress 3-Second Alarm	NSI	Abb	Abb
Suppress 3-Second Alarm and Branch (without indexing):	NSI	BI	blank
Suppress 3-Second Alarm and Branch (with indexing):	NSI	BI	NSI

Set Direction to Receive

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		D
SSB	K	III	D

Function. This instruction is incorporated in the processor receive program routine to set the line direction to RECEIVE. This instruction is effective only if the transmit, receive, and automatic switch is set to AUTOMATIC.

Word Marks. Word marks are not affected.

Timing.

Set Direction to Receive: T = .0333 ms.
Set Direction to Receive
and Branch (without indexing): T = .0666 ms.
Set Direction to Receive
and Branch (with indexing): T = .077 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Set Direction to Receive:	NSI	Dbb	Dbb
Set Direction to Receive and Branch (without indexing):	NSI	BI	blank
Set Direction to Receive and Branch (with indexing):	NSI	BI	NSI

Set Direction to Transmit

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		E
SSB	K	III	E

Function. This instruction is incorporated in the processor transmit program routine to set the line direction to TRANSMIT. This instruction is effective only if the transmit, receive, and automatic switch is set to AUTOMATIC.

Word Marks. Word marks are not affected.

Timing.

Set Direction to Transmit: T = .0333 ms.
Set Direction to Transmit
and Branch (without indexing): T = .0666 ms.
Set Direction to Transmit
and Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Set Direction to Transmit:	NSI	Ebb	Ebb
Set Direction to Transmit and Branch (without indexing):	NSI	BI	blank
Set Direction to Transmit and Branch (with indexing):	NSI	BI	NSI

Send End-of-File (EOF)

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		F
SSB	K	III	F

Function. This instruction is incorporated in the processor transmit or receive program routine to initiate an EOF (end-of-file) code signal to the remote 1009.

Word Marks. Word marks are not affected.

Timing.

Send End-of-File: T = .0333 ms.
Send End-of-File
and Branch (without indexing): T = .0666 ms.
Send End-of-File
and Branch (with indexing)): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Send End-of-File	NSI	Fbb	Fbb
Send End-of-File and Branch (without indexing):	NSI	BI	blank
Send End-of-File and Branch (with indexing):	NSI	BI	NSI

Operate in Load Mode

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
SS	K		L
SSB	K	III	L

Function. This instruction is placed ahead of the U %D1 D instruction in the processor receive program routine if the receiving 1009 is operating in the load mode. This instruction causes all word separators to be converted to word marks as they are received at the receiving 1009 and before they are placed in the receiving 1009 buffer. The receive station operates in a load mode until an end-of-message signal is received.

Timing.

Operate in Load Mode: T = .0333 ms.
Operate in Load Mode
and Branch (without indexing): T = .0666 ms.
Operate in Load Mode
and Branch (with indexing): T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
Operate in Load Mode:	NSI	Lbb	Lbb
Operate in Load Mode and Branch (without indexing):	NSI	BI	blank
Operate in Load Mode and Branch (with indexing):	NSI	BI	NSI

Program Instructions

The existing 1012 I/O instructions are used for read tape feature operations. (See *IBM 1012 Tape Punch* section of this publication.) For example, with the read/punch switch set to READ, a PUNCH BLANK instruction followed by a READ instruction causes the information at the read station to be transmitted to the processor and the tape to be moved one character position. A PUNCH BLANK instruction followed by a READ instruction is required to read each character.

The same considerations for starting and stopping the tape for punching, apply for reading as well. The read operation, or timing purposes, is the same as punching with READ BACK CHECK. See *IBM 1012 Tape Punch*, Form A26-5776-0.

IBM 1012 Tape Punch Special Features

IBM 1012 Tape Punch, Read Feature

The addition of the read special feature to the IBM 1012 allows the 1012 to operate as a reader (*input to* a using system) or as a punch (*output from* a using system). The mode of operation is controlled by a read/punch switch that is installed in the blank position next to the rewind switch.

When the read/punch switch is set to READ, the 1012 Tape Punch is modified as follows:

- All punching is inhibited.
- The reel sense arm's function is inhibited. The trailing end of the tape is allowed to feed from the supply reel without causing the machine to stop.
- The supply arm's lower-limit switch is bypassed. This allows the trailing end of the tape to pass the read station without causing the machine to stop.
- The feed switch, when pressed, causes the tape to advance without being punched with feed codes.

IBM 1403 Special Features

The IBM 1403 Printer special features add to the flexibility of the output applications for particular jobs, and also reduce the time required. Only those 1403 special features which require programmed instructions are listed here. Refer to *IBM 1403 Printer*, Form A24-3073.

Selective Tape Listing Feature

The selective tape listing feature can be attached to either of the Models 2 and 3 of the IBM 1403 Printer so that output results of data processed on a system can be printed on adding-machine style paper tapes (Figure H-82).

Operation

An IBM 1403 Printer with the selective tape listing feature installed continues to operate at regular 1403 speeds. Each tape is individually linespaced, one line at a time (no skipping or ejecting is possible). Tape is spaced by using modified 1403 CONTROL CARRIAGE Op code (F), when used with a d-character of A through H, which signals a single linespace for the corresponding tape. (The space operation takes place after the next print operation.) The modifier characters and the tapes they control are:

A – Tape 1	E – Tape 5
B – Tape 2	F – Tape 6
C – Tape 3	G – Tape 7
D – Tape 4	H – Tape 8

When a double-width tape is used, two tape linespace instructions are given, using the d-characters corresponding to the positions occupied by the double-width tape. If additional linespacing is wanted, a tape linespace instruction (CONTROL CARRIAGE Op code and the specific d-character and a PRINT instruction) are given. The print operation is a dummy print operation, and the print area in core storage should be clear so that nothing is printed. The linespace operation occurs after the print operation.

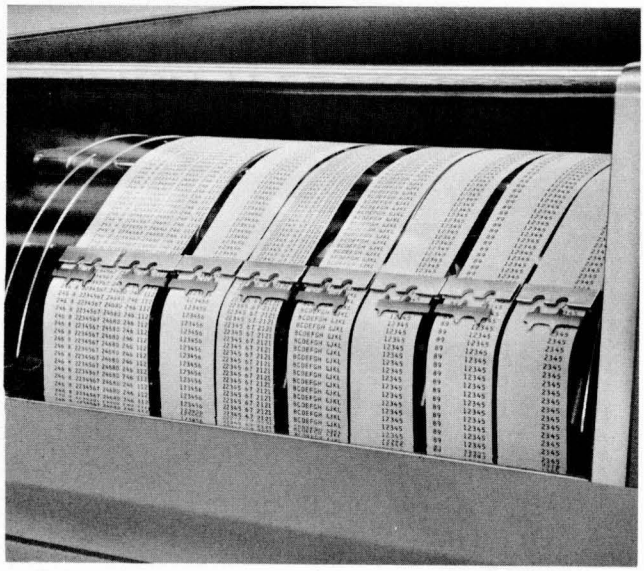


Figure H-82. Selective Tape Listing Feature Mechanism

To equalize the ribbon wear, the customer can vary the location of the master tape. This can be done by using the same width tape in another location and altering the program (changing the d-character to the character that corresponds to the new location).

An end-of-tape condition, sensed at the tape-spool tray, stops the printing operation and turns on the 1403 end-of-forms light.

When programming selective tape listing, the program should not select more than four tape-feeds simultaneously.

Control Carriage Selective Tape Listing Feature

Instruction Format.

Mnemonic	*Op Code*	*d-character*
CC	F	d

Function. Space a specific tape (as defined by the d-modifier character). Each such instruction causes a single tape to space up one line. Multiple instructions (with the correct d-characters) are required for spacing more than one tape.

Word Marks. Word marks are not affected.

Timing. See *1403 Printer Timings* section.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	dbb	dbb

Chaining. This instruction cannot be successfully chained.

Example. Space up tapes 1, 2, 7, and 8 after the next print operation. Tapes 7 and 8 might represent a single double-width tape (Figure H-83).

Autocoder

Label	Operation	OPERAND
	CC	A
	CC	B
	CC	H
	CC	G

Assembled Instruction: F A
F B
F G
F H

Figure H-83. Control Carriage, Selective Tape Listing Feature

IBM 1442, Models 1, 2, and 4, Special Features

Card Image Feature

This feature is available for the IBM 1442 Card Read-Punch, Models 1 and 2, and the IBM 1442 Card Reader, Model 4, with the IBM 1440 System. The card-image feature provides the circuitry to convert binary-coded cards into BCD codes, and BCD codes into binary-coded cards.

This feature also permits processing cards with multiple significant-digit punching in a single column.

When reading in card-image code, the validity check is suspended because all charcaters are considered valid. However, validity checking is still in effect for card columns designated as BCD (see *Word Marks* section), so if the I/O check stop switch is ON, the system will stop on a reader (or punch) validity error, or if the I/O check-stop switch is OFF, a BRANCH IF ERROR instruction may be used to enter a subroutine. Cards with interspersed, conventional punched codes and binary-coded data can be read.

Note: When this feature is installed on the first IBM 1442 attached to an IBM 1440 system, it also functions on the second 1442 installed on the system.

This feature permits reading punched data into the IBM 1441 unit without the normal translation from the standard IBM punched-card code to BCD code. In BCD mode, each card column of data is stored in two adjacent positions of core storage. Similarly, the data in two adjacent core positions can be punched into one card column (Figure H-84).

Instructions applying to the 1442 cannot be successfully chained.

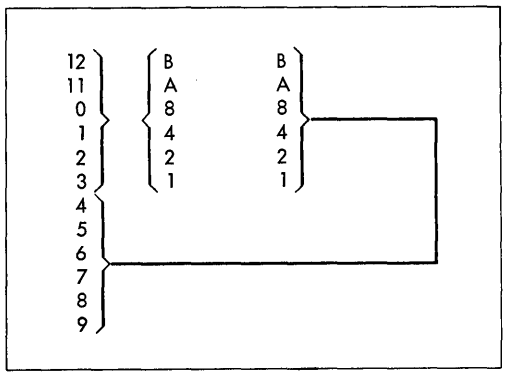


Figure H-84. Card Image Representation

Read Card Image

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
RCB	M	%Gn	BBB	R

Function. This instruction causes a read-card-image operation to be performed by the presence of a 9 or 0 (zero) in the n position of the A-address. The 9 also selects the number-1 unit on the system; a 0 (zero) in this position selects the number-2 unit when two units are on the same system.

The B-address is the address of the core storage where column-1 information is to be stored.

The instruction terminates when a group mark with word mark is sensed at location $(B + L_B)$, where L_B is the number of card columns to be read into the processing-unit core storage.

Word Marks. Special significance is assigned to a word mark during the execution of a CARD IMAGE READ instruction. The word mark is used to signal a change in the mode of operation. For example, in the card-image-read operation in the normal reading mode, each card column read is translated and stored in single core-storage locations until a word mark is detected in core storage. This signals the reading mode to change. Translation from the standard IBM punched-card code is suspended. Suspension causes the card data beginning at the word mark to be stored in two adjacent core-storage positions for each column read. Detection of another word mark causes the reading to revert to the normal reading mode.

Note. If the word marks are incorrectly placed, undetected invalid characters can be stored. The word mark must be associated with the upper character of a card-image set. Any given card column must be stored in *either* binary-coded-decimal or card-image mode.

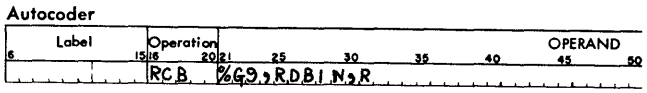
Timing.

Model 1 $T = 21.0999 + 1.3\ (L_B + 1)$ ms
Model 2 $T = 15.0999 + 1.0\ (L_B + 1)$ ms

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B+L_B+1$

Example. Transfer the data in card image form from card read-punch 1 to the area in core storage labeled RDBIN (0303); Figure H-85.



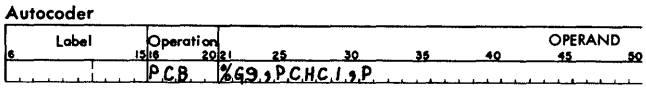
Assembled Instruction: M %G9 303 R

Figure H-85. Read Card in Card Image Mode

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B+L_B+1$

Example. Punch the data in card image form on card read-punch 1 beginning in the area labeled PCHCI (0303) and ending with a group mark with a word mark (Figure H-86).



Assembled Instruction: M %G9 303 P

Figure H-86. Punch-and-Stop in Card Image Mode

Punch Card Image and Stop

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
None	M	%Gn	BBB	P

Function. This instruction causes a punch-card-image operation to be performed by the presence of a 9 or 0 (zero) in the n position of the A-address. The 9 also selects the number-1 unit on the system; a 0 (zero) in this position selects the number-2 unit when two units are on the same system.

The B-address is the location of the data in storage to be punched. The data in core storage is transferred in ascending sequence to the card punch beginning at B-address until a group mark with word mark is sensed at $(B+L_B)$. This information is punched in successive columns of the card at the punch station. A punching operation following another punch operation (no intervening card feed) causes the data at the B-address to be punched in the next successive card column. The program must be written so as not to exceed 80-columns of punching per card. If 80 columns are exceeded, data is lost.

The operation is terminated by a group mark with word mark in the rightmost position of the field.

Word Marks. The word marks within the data being punched are neither considered nor affected.

Timing.

Model 1.

$T = 216.3499 + 12.5\ (L_B)$ ms

Model 2,

$T = 163.2299 + 6.25\ (L_B)$ ms

Punch Card Image and Feed

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
PCB	M	%Gn	BBB	G

Function. This instruction is used to transfer data from core storage into the card read-punch for punching into a card. When punching ends, the card is ejected from the punch station and selected into a stacker.

This instruction causes a punch-card-image operation to be performed by the presence of a 9 or 0 (zero)in the n position of the A-address. The 9 also selects the number-1 unit on the system; a 0 (zero) in this position selects the number-2 unit when two units are on the same system.

The data stored in the core-storage position specified by the B-address is transferred and punched in the card column registered beneath the punching mechanism. The rest of the data located in the adjacent core-storage positions is transferred, column-by-column, and punched in the adjacent card columns until a group mark with word mark in core storage is sensed. The number of characters punched in the card depends upon the B-field length that is established in core storage. The B-field length can be from 1 to 80 positions, plus one position for the group mark with word mark. (Characters in excess of 80 all punch in column 81, and are lost.) When the punching operation ends, the card is ejected from the punch station and selected into a stacker.

The card located at the read station advances during this operation, also, but the data in the card is not transferred into core storage. A card from the hopper is also advanced and registered at the read station during the punch-and-feed operation.

Word Marks. Word marks are not affected. A group mark with word mark is needed to end the operation.

Timing.

Model 1.

$T = 216.3499 + 12.5\ (L_B)$ ms

Model 2,

$T = 163.2299 + 6.25\ (L_B)$ ms

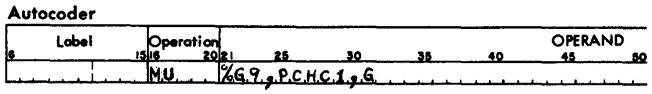
A period of 210 ms elapses before another card read-punch operation can be executed.

Note. When a punch-and-feed operation follows either a read-card or punch-and-feed operation, the card at the punch station is registered in column 1, and punching begins in column 1. When a punch-and feed-operation follows a punch-and-stop operation, the card at the punch station is the card that was punched during the previous operation; punching begins in the column adjacent to the last column punched.

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	$B + L_B + 1$

Example. Punch the data in card-image form on card read-punch 1 beginning in the area labeled PCHC1 (0303) and ending with a group mark with word mark, and then eject the card (Figure H-87).



Assembled Instruction: M %G9 303 G

Figure H-87. Punch-and-Feed in Card Image Mode

Selective Stacker

This feature provides a second stacker for the IBM 1442 Model 1, so that cards can be selected under program control for special applications. The IBM 1442 Models 2 and 4, have two stackers each as standard equipment.

Select Stacker

Instruction Format.

Mnemonic	*Op Code*	*d-character*
SS	K	2

Function. This instruction directs the card at the punch station into stacker 2. Unless stacker 2 has been selected before the operation that ejects the card (READ OR PUNCH FEED), the ejected card is directed to stacker 1.

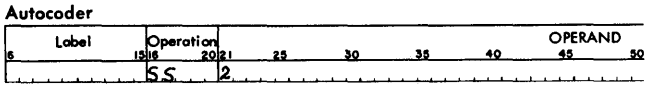
Word Marks. Word marks are not affected.

Timing. T = .0333 ms

Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	2bb	2bb

Example. Enter selected card into pocket 2 (Figure H-88).



Assembled Instruction: K 2

Figure H-88. Selective Stacker

Punch-Column Skip Feature

This special feature increases card output by allowing the punch portion of the attached card read-punch(es) to space over a specified number of card columns without interlocking the system. The punch-column-skip operation is initiated by executing a PUNCH COLUMN SKIP instruction.

Punch-Column Skip

Instruction Format.

Mnemonic	*Op Code*	*A-address*	*B-address*	*d-character*
PSK	M	%Gn	nnn	C

Function. This instruction initiates the skip operation. A-address of %Gn specifies one of the two card read-punches. The first 1442 attached to the system

is designated by a 1 in the n position. The second 1442 attached to the system is designated by a 2 in the n position.

The B-address is a 3-position number that specifies the number of card columns to be spaced through the punch station. For example, if a punch-column-skip operation of 40 columns is specified, the 3-position B-address would be 040.

The C d-character specifies a punch-column-skip operation.

Word Marks. Word marks are not affected.

Timing. $T = .0111\ (81 - L_B) + .1110$ ms

Time available to the processor is:

$T = 3.13 + (6.25 \times$ number of columns skipped) ms for IBM 1442, Model 2.

$T = 6.25 + (12.5 \times$ number of columns skipped) ms for IBM 1442, Model 1.

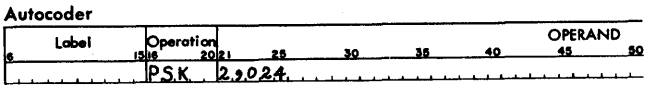
Address Registers After Operation.

I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
NSI	BBB	081

Chaining. This instruction cannot be successfully chained.

The stored program is released for further processing as soon as the instruction is executed in the processing unit, so that the processing of subsequent instructions and the skip operation are taking place at the same time.

Example. Skip 24 columns on card read-punch 2 (Figure H-89).



Assembled Instruction: M %G2 024 C

Figure H-89. Punch Column Skip

IBM 1447 Special Features

Branch If Buffer Busy

Instruction Format.

Mnemonic	*Op Code*	*I-address*	*d-character*
BIN	B	III	□

Function. This branch indicator is turned on when the inquiry-request (Q) indicator is on, or when the 1447 buffer is occupied.

Note: Refer to *IBM 1447 Console,* Form A24-3031, for special feature considerations and additional information.

Word Marks. Word marks are not affected.

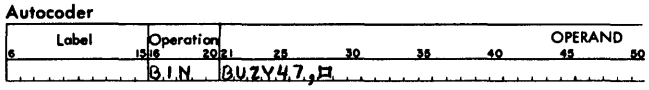
Timing.

No branch, or branch without indexing:
T = .0666 ms
Branch with indexing: T = .0777 ms.

Address Registers After Operation.

	I-Add. Reg.	*A-Add. Reg.*	*B-Add. Reg.*
No Branch	NSI	BI	□bb
Branch (without indexing)	NSI	BI	blank
Branch (with indexing)	NSI	BI	NSI

Example. Test the buffer-busy indicator and branch to a subroutine labeled BUZY47 (0554) if the test is successful; Figure H-90.



Assembled Instruction: B 554 □

Figure H-90. Branch if Buffer-Busy Indicator On

Appendix

COLL SEQ	DEFINED CHARACTER	CARD CODE	BCD CODE	CHARACTER SET 13	39	52 A	52 H	63
0	Blank		C	X	X	X	X	X
1	. Period	12-3-8	B A 8 2 1	X	X	X	X	X
2	□ Lozenge	12-4-8	C B A 8 4			□	)	X
3	[Left Bracket	12-5-8	B A 8 4 1					X
4	< Less Than	12-6-8	B A 8 4 2					X
5	≡ Group Mark	12-7-8	C B A 8 4 2 1					X
6	& Ampersand	12	C B A			&	+	X
7	$ Dollar Sign	11-3-8	C B 8 2 1		X	X	X	X
8	* Asterisk	11-4-8	B 8 4	X		X	X	X
9	] Right bracket	11-5-8	C B 8 4 1					X
10	; Semicolon	11-6-8	C B 8 4 2					X
11	Δ Delta	11-7-8	B 8 4 2 1					X
12	- Hyphen	11	B	X		X	X	X
13	/ Diagonal	0-1	C A 1			X	X	X
14	, Comma	0-3-8	C A 8 2 1		X	X	X	X
15	% Percent Mark	0-4-8	A 8 4			%	(	X
16	Y Word Separator	0-5-8	C A 8 4 1					X
17	\ Left Oblique	0-6-8	C A 8 4 2					X
18	# Segment Mark	0-7-8	A 8 4 2 1					X
19	ↄ Substitute Blank	2-8	A			X	X	X
20	# Number Sign	3-8	8 2 1			#	=	X
21	@ At Sign	4-8	C 8 4			@	'	X
22	: Colon	5-8	8 4 1			X	X	X
23	> Greater Than	6-8	8 4 2					X
24	√ Radical	7-8	C 8 4 2 1					X
25	? (Plus Zero)	12-0	C B A 8 2			X	X	X
26	A	12-1	B A 1		X	X	X	X
27	B	12-2	B A 2		X	X	X	X
28	C	12-3	C B A 2 1		X	X	X	X
29	D	12-4	B A 4		X	X	X	X
30	E	12-5	C B A 4 1		X	X	X	X
31	F	12-6	C B A 4 2		X	X	X	X
32	G	12-7	B A 4 2 1		X	X	X	X
33	H	12-8	B A 8		X	X	X	X
34	I	12-9	C B A 8 1		X	X	X	X
35	! (- zero)	11-0	B 8 2			X	X	X
36	J	11-1	C B 1		X	X	X	X
37	K	11-2	C B 2		X	X	X	X
38	L	11-3	B 2 1		X	X	X	X
39	M	11-4	C B 4		X	X	X	X
40	N	11-5	B 4 1		X	X	X	X
41	O	11-6	B 4 2		X	X	X	X
42	P	11-7	C B 4 2 1		X	X	X	X
43	Q	11-8	C B 8		X	X	X	X
44	R	11-9	B 8 1		X	X	X	X
45	⫩ Record Mark	0-2-8	A 8 2			X	X	X
46	S	0-2	C A 2		X	X	X	X
47	T	0-3	A 2 1		X	X	X	X
48	U	0-4	C A 4		X	X	X	X
49	V	0-5	A 4 1		X	X	X	X
50	W	0-6	A 4 2		X	X	X	X
51	X	0-7	C A 4 2 1		X	X	X	X
52	Y	0-8	C A 8		X	X	X	X
53	Z	0-9	A 8 1		X	X	X	X
54	0 (Zero)	0	C 8 2	X	X	X	X	X
55	1	1	1	X	X	X	X	X
56	2	2	2	X	X	X	X	X
57	3	3	C 2 1	X	X	X	X	X
58	4	4	4	X	X	X	X	X
59	5	5	C 4 1	X	X	X	X	X
60	6	6	C 4 2	X	X	X	X	X
61	7	7	4 2 1	X	X	X	X	X
62	8	8	8	X	X	X	X	X
63	9	9	C 8 1	X	X	X	X	X

Figure I-1. 1440 Character Code Chart in Collating Sequence

Declarative and Assembler-Control Statements

DECLARATIVE OPERATIONS	
Mnemonic Op Code	Description
DA	Define Area
DC	Define Constant (No Word Mark)
DCW	Define Constant With Word Mark
DS	Define Symbol
DSA	Define Symbol Address
EQU	Equate

ASSEMBLER CONTROL OPERATIONS			
Mnemonic	Description	Mnemonic	Description
CTL	Control	ORG	Origin
END	End	XFR	Transfer
ENT	Enter New Coding Mode	SFX	Suffix
		JOB	Job
EX	Execute	INSER	Insert
LTORG	Literal Origin	DELET	Delete
LIST	Resume Listing		
SPCE	Space n Lines		
ULST	Stop Listing		

Figure I-2. Declarative and Assembler Control Operation

Figure I-2 lists all the declarative and assembler-control mnemonic operation codes that are valid for the disk Autocoder language.

Figure I-3 shows all valid control-carriage instruction d-modifier characters.

Figure I-4 illustrates all valid (1440 Autocoder) select-stacker instruction d-modifier characters.

Note: Several miscellaneous input/output devices that can be attached to the 1440 system require special SS mnemonic d-characters. These are coded in the operand portion of the instruction statement. Refer to *Programming Note, 1440 Autocoder Assembly* section.

d	Immediate skip to	d	Skip after print to
1	Channel 1	A	Channel 1
2	Channel 2	B	Channel 2
3	Channel 3	C	Channel 3
4	Channel 4	D	Channel 4
5	Channel 5	E	Channel 5
6	Channel 6	F	Channel 6
7	Channel 7	G	Channel 7
8	Channel 8	H	Channel 8
9	Channel 9	I	Channel 9
0	Channel 10	?	Channel 10
#	Channel 11	•	Channel 11
@	Channel 12	□	Channel 12
d	**Immediate space**	**d**	**After print-space**
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

Figure I-3. Control Carriage d-Characters

SELECT STACKER (1442, 1444)				
Unit	(Device)	d	Feed	Stacker Pocket
1	(1442)	2	Read/Punch	2
2	(1442)	0	Read/Punch	2
3	(1444)	#	Punch	2

Note: See Programming Note, 1440 Autocoder Assembly section.

Figure I-4. Select-Stacker d-Characters

Imperative Statements

Figure I-5 is an imperative-statement reference chart that lists all valid mnemonic imperative operation codes. The information given for each mnemonic is:

1. The mnemonic description.
2. The machine-language operation code.
3. The operand sequence. This entry represents the valid set of operands to be used with the mnemonic. Deviations from the specified operand sequences are detected, and the appropriate error message is printed.

The following symbols are used to describe the operand sequence:

Symbol	*Meaning*
RD	Declared field. An actual, symbolic, or asterisk address, or an area-defining literal. Address-adjustment and indexing are permitted.
D	Constant or declared field. An actual, symbolic, or asterisk address, or a literal. Address-adjustment and indexing are permitted.
XC	X-control field. Address of a unit, such as %U1, used to address a unit of system feature. Address-adjustment and/or indexing are not permitted, but XC may be a symbolic address if it has been equated to the desired I/O device.
n	Single numeric character.
S	Symbolic address. Address-adjustment and/or indexing are not permitted.
d	d-modifier character. Used to modify the operation code.
,	Operand separator.
/	Optional operand separator. For example, n/XC/S means that either a single numeric character, an X-control field, or a symbolic address may be used for the operand.
1	Single alphabetic character.

Mnemonic	Description	Op Code	Operand Sequence	Chain	X-Control Field	d-Character
			ARITHMETIC OPERATIONS			
A	Add	A	D,RD	F		
D	Divide*	%	D,RD	No		
M	Multiply*	@	D,RD	No		
S	Subtract	S	D,RD	F		
ZA	Zero and Add	?	D,RD	F		
ZS	Zero and Subtract	!	D,RD	F		
			DATA CONTROL OPERATIONS			
MIZ	Move and Insert Zeros*	X	D,RD	No		
MCE	Move Characters and Edit	E	D,RD	No		
MCS	Move Characters and Suppress Zeros	Z	D,RD	No		
MRCWG	Move Characters and Word Marks to Group Mark-Word Mark in A-Field*	P	D,RD	F		>
MLCWA	Move Characters and Word Marks to Word Mark in A-Field	L	D,RD	HF		
MRCM	Move Characters to Record Mark or Group Mark-Word Mark	P	D,RD	F		
MLC	Move Characters to Word Mark in A- or B-Field	M	D,RD	HF		
MLNS	Move Single Numeric	D	D,RD	F		
MLZS	Move Single Zone	Y	D,RD	F		
			LOGIC OPERATIONS			
BWZ	Branch if 0-Zone (A-bit, no B-bit)	V	RD,D,d	F		S**
BWZ	Branch if 11-Zone (B-bit, no A-bit)	V	RD,D,d	F		K**
BWZ	Branch if 12-Zone (AB-bits)	V	RD,D,d	F		B**
BIN	Branch if Access Busy	B	RD,d	No		**
BIN	Branch if Access Inoperable	B	RD,d	No		N**
BIN	Branch if Any Disk Error	B	RD,d	No		Y**
BAV	Branch if Arithmetic Overflow	B	RD	No		Z
BBE	Branch if Bit Equal*	W	RD,D,d	F		**
BC9	Branch if Carriage Channel 9	B	RD	No		9
BCV	Branch if Carriage Overflow (12)	B	RD	No		@
BCE	Branch if Character Equal	B	RD,D,d	F		**
BIN	Branch if Disk Error	B	RD,d	No		V**
BEF	Branch if End of File or End of Reel	B	RD	No		K
BE	Branch if Equal Compare (B = A)	B	RD	No		S
BH	Branch if High Compare (B>A)	B	RD	No		U
BIN	Branch if Inquiry Clear*	B	RD,d	No		* **
BIN	Branch if Inquiry Request*	B	RD,d	No		Q**
BLC2	Branch if Last Card (Reader Unit 2)	B	RD	No		&
BLC	Branch if Last Card (Sense Switch A)	B	RD	No		A
BL	Branch if Low Compare (B<A)	B	RD	No		T
BM	Branch if Minus (11-Zone)	V	RD,d	F		K
BWZ	Branch if No Zone (No A- or B-Bit)	V	RD,D,d	F		2**
BPB	Branch if Printer Busy	B	RD	No		P
BPCB	Branch if Printer Carriage Busy	B	RD	No		R (1403 only)
BIN	Branch if Printer Error (I/O Check Stop Switch Off)	B	RD,d	No		÷**
BIN	Branch if Process Check (Check Stop Switch Off)	B	RD,d	No		%**
BIN	Branch if Punch Error (I/O Check Stop Switch Off)	B	RD,d	No		!**
BIN	Branch if Reader Error (I/O Check Stop Switch Off)	B	RD,d	No		?**
BSS	Branch if Sense Switch A	B	RD,d	No		A**
BSS	Branch if Sense Switch (B-G)*	B	RD,d	No		(B-G)**

* Special Feature

** d-Character must be placed in operand when coding in Autocoder.

Figure I-5. Imperative Operations (Part 1 of 3)

Mnemonic	Description	Op Code	Operand Sequence	Chain	X-Control Field	d-Character
			LOGIC OPERATIONS (CONT.)			
BER	Branch if Tape Transmission Error	B	RD	No		L
BIN	Branch if Unequal Address Compare (Disk)	B	RD,d	No		X**
BU	Branch if Unequal Compare (B≠A)	B	RD	No		/
BW	Branch if Word Mark	V	RD,D,d	F		1
BWZ	Branch if Word Mark or 0-Zone	V	RD,D,d	F		T**
BWZ	Branch if Word Mark or 11-Zone	V	RD,D,d	F		L**
BWZ	Branch if Word Mark or 12-Zone	V	RD,D,d	F		C**
BWZ	Branch if Word Mark or No Zone	V	RD,D,d	F		3**
BIN	Branch if Wrong-Length Record (Disk)	B	RD,d	No		W**
B	Branch Unconditional	B	RD	No		
C	Compare	C	D,D	F		
			MISCELLANEOUS OPERATIONS			
CC	Carriage Control	F	d	No		**
CS	Clear Storage	/	RD	F		
CS	Clear Storage and Branch	/	RD,RD	No		
CW	Clear Word Mark	□	RD,RD	F		
H	Halt	.	D,D	F		
H	Halt and Branch	.	RD			
MA	Modify Address*	#	D,RD	F		
NOP	No Operation	N	XC/RD,d	HF		**
SAR	Store A-Address Register*	Q	RD,D	HF		
SBR	Store B-Address Register*	H	RD,D	HF		
SS	Select Stacker	K	d	No		**
SSB	Select Stacker and Branch (Serial I/O device only)	K	RD,d	No		**
SW	Set Word Mark	,	RD,RD	F		
TR	Translate*	T	D,RD	No		
TRW	Translate with Word Marks*	T	D,RD	No		>
			MAGNETIC TAPE OPERATIONS			
BSP	Backspace Tape	U	n/XC/S	No	%Un	B
CU	Control Unit	U	XC/S,d	No	%In	**
RT	Read Tape	M	n/XC/S,RD	No	%Un	R
RTB	Read Tape Binary	M	n/XC/S,RD	No	%Bn	R
RTW	Read Tape with Word Marks	L	n/XC/S,RD	No	%Un	R
RWD	Rewind Tape	U	n/XC/S	No	%Un	R
RWU	Rewind and Unload Tape	U	n/XC/S	No	%Un	U
SKP	Skip and Blank Tape	U	n/XC/S	No	%Un	E
WT	Write Tape	M	n/XC/S,RD	No	%Un	W
WTB	Write Tape Binary	M	n/XC/S,RD	No	%Bn	W
WTM	Write Tape Mark	U	n/XC/S	No	%Un	M
WTW	Write Tape with Word Marks	L	n/XC/S,RD	No	%Un	W
			I/O DEVICE OPERATIONS			
CU	Control Unit	U	XC/S,d	No	%In	**
LU	Load Unit	L	XC/S,RD,d	No	%In	**
MU	Move Unit	M	XC/S,RD,d	No	%In	**
P	Punch a Card and Feed	M	n/XC/S,RD	No	%Gn	G
PCB	Punch Column Binary and Feed*	M	n/XC/S,RD	No	%Gn	G
PS	Punch a Card and Stop	M	n/XC/S,RD	No	%Gn	P
PSK	Punch Skip*	M	n/XC/S,RD	No	%Gn	C
R	Read a Card	M	n/XC/S,RD	No	%Gn	R
RCB	Read Column Binary*	M	n/XC/S,RD	No	%Gn	R

* Special Feature

** d-Character must be placed in operand when coding in Autocoder.

Figure I-5. Imepartive Operations (Part 2 of 3)

Mnemonic	Description	Op Code	Operand Sequence	Chain	X-Control Field	d-Character
	I/O DEVICE OPERATIONS (CONT.)					
RCP	Read Console Printer	M	RD	No	%T0	R
RCPW	Read Console Printer with Word Marks	L	RD	No	%T0	R
W	Write a Line	M	RD	No	%Y1	W
WCP	Write Console Printer	M	RD	No	%T0	W
WCPW	Write Console Printer with Word Marks	L	RD	No	%T0	W
WS	Write and Suppress Space	M	RD	No	%Y1	S
	DISK OPERATIONS					
CU	Control Unit	U	XC/S,d	No	%In	**
RD	Read Disk Sector(s)	M	RD	No	%F1	R
RDW	Read Disk Sector(s) with Word Marks	L	RD	No	%F1	R
RDTR	Read Disk Track Record*	M	RD	No	%F2	R
RDTRW	Read Disk Track Record with Word Marks*	L	RD	No	%F2	R
RDTA	Read Disk Track Record with Address*	M	RD	No	%F@	R
RDTAW	Read Disk Track Record with Address and Word Marks*	L	RD	No	%F@	R
RDT	Read Disk Track Sectors with Addresses	M	RD	No	%F6	R
RDTW	Read Disk Track Sectors with Addresses and Word Marks	L	RD	No	%F6	R
RDCO	Read Disk with Sector Count Overlay	M	RD	No	%F5	R
RDCOW	Read Disk with Sector Count Overlay with Word Marks	L	RD	No	%F5	R
SD	Seek Disk	M	RD	No	%F0	R
SDE	Scan Disk Equal*	M	RD	No	%F8	W
SDEW	Scan Disk Equal with Word Marks*	L	RD	No	%F8	W
SDH	Scan Disk High, Equal*	M	RD	No	%F9	W
SDHW	Scan Disk High, Equal with Word Marks*	L	RD	No	%F9	W
SDL	Scan Disk Low, Equal*	M	RD	No	%F7	W
SDLW	Scan Disk Low, Equal with Word Marks*	L	RD	No	%F7	W
WDC	Write Disk Check	M	RD	No	%F3	W
WDCW	Write Disk Check with Word Marks	L	RD	No	%F3	W
WD	Write Disk Sector(s)	M	RD	No	%F1	W
WDW	Write Disk Sector(s) with Word Marks	L	RD	No	%F1	W
WDTR	Write Disk Track Record*	M	RD	No	%F2	W
WDTRW	Write Disk Track Record with Word Marks*	L	RD	No	%F2	W
WDTA	Write Disk Track Record with Address*	M	RD	No	%F@	W
WDTAW	Write Disk Track Record with Address and Word Marks*	L	RD	No	%F@	W
WDTW	Write Disk Track Sectors	L	RD	No	%F6	W
WDT	Write Disk Track Sectors with Addresses	M	RD	No	%F6	W
WDCO	Write Disk with Sector Count Overlay	M	RD	No	%F5	W
WDCOW	Write Disk with Sector Count Overlay with Word Marks	L	RD	No	%F5	W

* Special Feature

** d-Character must be placed in operand when coding in Autocoder.

Figure I-5. Imperative Operations (Part 3 of 3)

4. The code that indicates whether chaining (deletion of one or more operands) is permitted (see *Chaining* section).

Code	*Meaning*
F	Full chaining only – both operands must be deleted for a logical chaining operation.
HF	Half or full chaining – either the last or both operands deleted for a logical chaining operation.
NO	No chaining – no operands deleted.

Note: Except for the move, load, store, branch, and I/O operations, the presence of an A-field address and the absence of the B-field address causes the A-field to modify itself. Thus, a SUBTRACT instruction with only an A-field address will cause the field specified to be subtracted from itself. *This is in no sense chaining.* The move and load operations having only an A-field address *are* considered chaining. Refer to the particular instruction for additional information.

5. The X-control field, if required.

6. The d-character, if required. Figures I-3 and I-4 list the Control Carriage (CC) and Select Stacker (SS) mnemonics.

Note: See *Programming Note, 1440 Autocoder Assembly* section.

Index of Branch Instructions and d-Modifiers

This alphabetic listing of 1440 branch instructions contains the following information:

- The feature or device to which the instruction applies.
- The instruction name.
- The autocoder mnemonic.
- The branch op-code.
- The branch d-modifier character.
- The page number in this manual where the instruction is defined.

Note: Although the various branches associated with the select-stacker mnemonic (SSB) are considered unconditional, they are also listed here.

Direct Data Channel Feature

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if End of Transmission	BIN	B	2	H-4
Branch if Indicator On	BIN	B	d	H-4
Branch if Process Check	BIN	B	1	H-4
Branch if Read Data	BIN	B	7	H-4
Branch if Read Request	BIN	B	3	H-4
Branch if System A Stopped	BIN	B	8	H-4
Branch if Write Data	BIN	B	6	H-4
Branch if Write Request	BIN	B	4	H-4
Read Request and Branch	SSB	K	C	H-2
Reset and Branch	SSB	K	E	H-3
Write Request and Branch	SSB	K	D	H-3

Expanded Serial Input/Output Adapter Feature

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Operational Out	BIN	B	8	H-22
Branch if Primary Control Indicator On	BIN	B	7	H-22
Branch if Primary Read Indicator On	BIN	B	3	H-20
Branch if Primary Sense Indicator On	BIN	B	6	H-21
Branch if Primary Write Indicator On	BIN	B	4	H-21
Branch if Transmission Error Indicator On	BIN	B	1	H-20
Disable Interrupt and Branch	SSB	K	<	H-19
Enable Interrupt and Branch	SSB	K	>	H-19
Send Attention Signal and Branch	SSB	K	A	H-17
Send End Signal and Branch	SSB	K	E	H-17
Send Unusual End Signal and Branch	SSB	K	F	H-18

IBM 1009 Data Transmission Unit

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Attention	BIN	B	6	H-55
Branch if Buffer Service	BIN	B	2	H-55
Branch if Indicator On	BIN	B	d	G-5, H-54
Branch if Rec. EOM or Init. Inq.	BIN	B	7	H-55
Branch if Receive EOF	BIN	B	8	H-55
Branch if Receive Error	BIN	B	5	H-55
Branch if Reply Bad	BIN	B	4	H-55
Branch if Reply Good	BIN	B	3	H-55
Branch if 1009 Ready	BIN	B	1	H-55

Instruction	Mnemonics	Op Code	d-Char.	Page
Operate in Load Mode and Branch	SSB	K	L	H-56
Send EOF and Branch	SSB	K	F	H-56
Set Direction to Rec. and Branch	SSB	K	D	H-56
Set Direction to Trans. and Branch	SSB	K	E	H-56
Suppress 3-Second Alarm and Branch	SSB	K	A	G-6, H-55

IBM 1011 Paper Tape Punch

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Input/Output Indicator On	BIN	B	1	E-10
Branch if Paper Tape Reader Ready	BIN	B	2	E-11

IBM 1012 Tape Punch

Instruction	Mnemonic	Op Code	d-Char.	Page
Backspace Tape and Branch	SSB	K	A	E-13
Branch if In Backspace Oper.	BIN	B	1	E-14
Branch if Tape Punch Not Ready to Accept Data	BIN	B	3	E-14
Branch if Tape Punch Not Ready to Read	BIN	B	4	E-15
Branch if Tape Punch Overextended	BIN	B	5	E-15
Branch if Tape Punch Ready	BIN	B	2	E-14
Branch if Supply Reel Low or Chad Box Full	BIN	B	6	E-16

IBM 1026 Transmission Control Unit

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Buffer-Busy Indicator On	BIN	B	¤	G-47
Branch if Inquiry-Clear Indicator On	BIN	B	*	G-48
Branch if Request-Service Indicator On	BIN	B	Q	G-47
Branch if Time-Emitter Indicator On	BIN	B	]	G-48
Disable Interrupt and Branch	SSB	K	<	G-46
Enable Interrupt and Branch	SSB	K	>	G-45

IBM 1231 Optical Mark Page Reader

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Auto-Select	BIN	B	1	G-29
Branch if Buffer Full	BIN	B	2	G-29
Branch if Hopper Empty	BIN	B	4	G-30
Branch if Read Error or Overrun Detection	BIN	B	5	G-30
Branch if Timing Mark Check	BIN	B	6	G-30
Branch if 1231 Ready to Read	BIN	B	3	G-29
Select Stacker and Branch	SSB	K	A	G-31

IBM 1285 Optical Reader

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if End of File	BIN	B	8	G-40
Branch if End of Line	BIN	B	2	G-40
Branch if Error	BIN	B	1	G-40
Branch if Header Information	BIN	B	5	G-40
Branch if Indicator On	BIN	B	d	G-39
Branch if Marked Line	BIN	B	4	G-40
Branch if Reader Ready	BIN	B	7	G-40
Branch if Reader Ready to Read a Line	BIN	B	6	G-40
Branch if Reader Transporting	BIN	B	3	G-40
Go to Next Line and Branch	SSB	K	G	G-39
Mark a Line and Branch	SSB	K	M	G-39

IBM 1301 Disk Storage

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Access Busy	BIN	B	\	F-11
Branch if Access Inoperable	BIN	B	N	F-11
Branch if Any Disk Condition	BIN	B	Y	F-11
Branch if Disk Error	BIN	B	V	F-11
Branch if Indicator On	BIN	B	d	F-11
Branch if Unequal Address Compare	BIN	B	X	F-11
Branch if Wrong Length Record	BIN	B	W	F-11

IBM 1311 Disk Storage Drive

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Access Busy	BIN	B	\	F-29
Branch if Access Inoperable	BIN	B	N	F-29
Branch if Any Disk Condition	BIN	B	Y	F-29
Branch if Disk Error	BIN	B	V	F-29
Branch if Indicator On	BIN	B	d	F-29
Branch if Unequal Address Compare	BIN	B	X	F-29
Branch if Wrong Length Record	BIN	B	W	F-29

IBM 1403 Printer

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Channel 12	BCV	B	@	D-6
Branch if Channel 9	BC9	B	9	D-4
Branch if Printer Busy	BPB	B	P	D-4
Branch if Printer Carriage Busy	BPCB	B	R	D-4
Branch if Printer Error	BIN	B	‡	D-4

IBM 1412 Magnetic Ink Character Recognition Reader

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Account-Number-Field Indicator On	BIN	B	6	G-12
Branch if Amount-Field Indicator On	BIN	B	4	G-11
Branch if Document-Spacing-Check Indicator On	BIN	B	8	G-13
Branch if Process-Control-Field Indicator On	BIN	B	5	G-11
Branch if Read-Check Indicator On	BIN	B	3	G-10
Branch if Late-Read Indicator On	BIN	B	1	G-10
Branch if Read-Not-Ready Indicator On	BIN	B	2	G-10
Branch if Transit-Routing-Field Indicator On	BIN	B	7	G-12
Select Stacker and Branch	SSB	K	d	G-9

IBM 1441 Central Processing Unit

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Arithmetic Overflow	BAV	B	Z	B-7
Branch if Bit Equal (Special Feature)	BBE	W	d	H-1
Branch if Character Equal	BCE	B	d	B-8
Branch if Equal Compare (B = A)	BE	B	S	B-7
Branch if High Compare (B > A)	BH	B	U	B-7
Branch if Indicator On	BIN	B	d	B-7
Branch if Low Compare (B < A)	BL	B	T	B-7
Branch if Minus	BM	V	K	B-9
Branch if No Zone	BWZ	V	2	B-9

Instruction	Mnemonics	Op Code	d-Char.	Page
Branch if Process Check	BIN	B	%	B-7
Branch if Sense Switch A	BSS	B	A	B-7
Branch if Sense Switch B	BSS	B	B	B-7
Branch if Sense Switch C	BSS	B	C	B-7
Branch if Sense Switch D	BSS	B	D	B-7
Branch if Sense Switch E	BSS	B	E	B-7
Branch if Sense Switch F	BSS	B	F	B-7
Branch if Sense Switch G	BSS	B	G	B-7
Branch Unconditionally	B	B	none	B-7
Branch if Unequal Compare	BU	B	/	B-7
Branch if Word Mark	BW	V	1	B-9
Branch if Word Mark or No Zone	BWZ	V	3	B-9
Branch if Word Mark or Zone	BWZ	V	d	B-9
Branch if Word Mark or 0-Zone	BWZ	V	T	B-9
Branch if Word Mark or 11-Zone	BWZ	V	L	B-9
Branch if Word Mark or 12-Zone	BWZ	V	C	B-9
Branch if 0-Zone	BWZ	V	S	B-9
Branch if 11-Zone	BWZ	V	K	B-9
Branch if 12-Zone	BWZ	V	B	B-9

IBM 1442 Card Read-Punch

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Last Card, First Read-Punch	BLC	B	A	D-13
Branch if Last Card, Second Read-Punch	BLC2	B	&	D-13
Branch if Punch Error	BIN	B	!	D-13
Branch if Reader Error	BIN	B	?	D-12

IBM 1443 Printer

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Channel 9	BC9	B	9	D-23
Branch if Channel 12	BCV	B	@	D-24
Branch if Printer Busy	BPB	B	P	D-23
Branch if Printer Error	BIN	B	‡	D-23

IBM 1444 Card Punch

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Punch Error	BIN	B	!	D-28

IBM 1445 Printer

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Channel 9	BC9	B	9	D-32
Branch if Channel 12	BCV	B	@	D-32
Branch if Printer Busy	BPB	B	P	D-32
Branch if Printer Error	BIN	B	‡	D-31

IBM 1447 Console

Instruction	Mnemonic	Op Code	d-Char.	Page
Branch if Buffer Busy	BIN	B	□	H-63
Branch if Inquiry Clear	BIN	B	*	C-3
Branch if Inquiry Request	BIN	B	Q	C-3

IBM 1448 Transmission Control Unit

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if Early Warning	BIN	B	<	G-24
Branch if End of Block	BIN	B	>	G-23
Disable Interrupt and Branch	SSB	K	<	G-23
Enable Interrupt and Branch	SSB	K	>	G-23

IBM 7335 Magnetic Tape Unit

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Branch if End of Reel	BEF	B	K	E-5
Branch if Tape Error	BER	B	L	E-6

IBM 7740 Communication Control System

Instruction	*Mnemonic*	*Op Code*	*d-Char.*	*Page*
Attention Response and Branch	SSB	K	B	G-27
Branch if Attention-Signal Indicator On	BIN	B	4	G-27
Branch if Data Transmission Ended	BIN	B	1	G-27
Branch if Indicator On	BIN	B	d	G-27
Branch if Receive-Request Indicator On	BIN	B	3	G-27
Branch if Successful-Completion Indicator On	BIN	B	2	G-27
Control Request and Branch	SSB	K	F	G-27
End Response and Branch	SSB	K	E	G-27
Read Request and Branch	SSB	K	C	G-27
Sense Request and Branch	SSB	K	G	G-27
Signal Control and Branch	SSB	K	D	G-27
Write Request and Branch	SSB	K	D	G-27

Index of Instructions

Instruction	Type	Mnemonic	Format	Page
Branch if Last Card (First Reader Punch)	1442	BLC	B III A	D-13
Branch if Last Card (Second Reader Punch)	1442	BLC2	B III &	D-13
Branch if Late-Read Indicator On	1412	BIN	B III 1	G-10
Branch if Low Compare (B < A)		BL	B III T	B-7
Branch if Marked Line	1285	BIN	B III 4	G-40
Branch if Minus		BM	V III BBB 1	B-9
Branch if No Zone		BWZ	V III BBB 2	B-9
Branch if Operational Out	EXPS	BIN	B III 8	H-22
Branch if Paper-Tape Reader Ready	1011	BIN	B III 2	E-11
Branch if Primary-Control Indicator On	EXPS	BIN	B III 7	H-22
Branch if Primary-Read Inndicator On	EXPS	BIN	B III 3	H-20
Branch if Primary-Sense Indicator On	EXPS	BIN	B III 6	H-21
Branch if Primary-Write Indicator On	EXPS	BIN	B III 4	H-21
Branch if Printer Busy	1403	BPB	B III P	D-4
Branch if Printer Busy	1443	BPB	B III P	D-23
Branch if Printer Busy	1445	BPB	B III P	D-32
Branch if Printer-Carriage Busy	1403	BPCB	B III R	D-4
Branch if Printer Error	1403	BIN	B III ‡	D-4
Branch if Printer Error	1443	BIN	B III ‡	D-23
Branch if Printer Error	1445	BIN	B III ‡	D-31
Branch if Process-Control-Field Indicator On	1412	BIN	B III 5	G-11
Branch if Process Check	DDC	BIN	B III 1	H-4
Branch if Process Error		BIN	B III %	B-7
Branch if Punch Error	1442	BIN	B III !	D-13
Branch if Punch Error	1444	BIN	B III !	D-28
Branch if Read-Check Indicator On	1412	BIN	B III 3	G-10
Branch if Read Data	DDC	BIN	B III 7	H-4
Branch if Read Error or Overrun Detection	1231	BIN	B III 5	G-30
Branch if Read-Not-Ready Indicator On	1412	BIN	B III 2	G-10
Branch if Read Request	DDC	BIN	B III 3	H-4
Branch if Reader Error	1442	BIN	B III ?	D-12
Branch if Reader Ready	1285	BIN	B III 7	G-40
Branch if Reader Ready to Read a Line	1285	BIN	B III 6	G-40
Branch if Reader Transporting	1285	BIN	B III 3	G-40
Branch if Receive End of Message or Initial Inquiry	1009	BIN	B III 7	H-55
Branch if Receive (EOF)	1009	BIN	B III 8	H-55
Branch if Receive Error	1009	BIN	B III 5	H-55
Branch if Receive-Request Indicator On	7740	BIN	B III 3	G-27
Branch if Reply Bad	1009	BIN	B III 4	H-55
Branch if Reply Good	1009	BIN	B III 3	H-55
Branch if Request-Service Indicator On	1026	BIN	B III Q	G-47
Branch if Sense Switch A On	SS	BSS	B III A	B-7, H-40
Branch if Sense Switch B On	SS	BSS	B III B	B-7, H-40
Branch if Sense Switch C On	SS	BSS	B III C	B-7, H-40
Branch if Sense Switch D On	SS	BSS	B III D	B-7, H-40
Branch if Sense Switch E On	SS	BSS	B III E	B-7, H-40
Branch if Sense Switch F On	SS	BSS	B III F	B-7, H-40
Branch if Sense Switch G On	SS	BSS	B III G	B-7, H-40
Branch if Successful-Completion Indicator On	7740	BIN	B III 2	G-27
Branch if System A Stopped	DDC	BIN	B III 8	H-4
Branch if Tape Error	7335	BER	B III L	E-6
Branch if Tape-Punch Not Ready to Accept Data	1012	BIN	B III 3	E-14
Branch if Tape-Punch Not Ready to Read	1012	BIN	B III 4	E-15
Branch if Tape-Punch Overextended	1012	BIN	B III 5	E-15
Branch if Tape-Punch Ready	1012	BIN	B III 2	E-14
Branch if Time-Emitter Indicator On	1026	BIN	B III]	G-48
Branch if Timing-Mark Check	1231	BIN	B III 6	G-30
Branch if Transit-Routing Indicator On	1412	BIN	B III 7	G-12
Branch if Transmission-Error Indicator On	EXPS	BIN	B III 1	H-20
Branch if Unequal-Address Compare	1301	BIN	B III X	F-11
Branch if Unequal-Address Compare	1311	BIN	B III X	F-29
Branch if Word Mark		BW	V III BBB 1	B-9
Branch if Word Mark or No Zone		BWZ	V III BBB 3	B-9
Branch if Word Mark or Zero Zone		BWZ	V III BBB T	B-9
Branch if Word Mark or Zone		BWZ	V III BBB d	B-9
Branch if Word Mark or 11 Zone		BWZ	V III BBB L	B-9
Branch if Word Mark or 12 Zone		BWZ	V III BBB C	B-9
Branch if Write Data	DDC	BIN	B III 6	H-4
Branch if Write Request	DDC	BIN	B III 4	H-4
Branch if Wrong-Length Record	1301	BIN	B III W	F-11
Branch if Wrong-Length Record	1311	BIN	B III W	F-29

Instruction	Type	Mnemonic	Format	Page
Branch if Zero Zone		BWZ	V III BBB S	B-9
Branch if 1009 Ready	1009	BIN	B III 1	H-55
Branch if 11 Zone		BWZ	V III BBB K	B-9
Branch if 12 Zone		BWZ	V III BBB B	B-9
Branch if 1231 Ready to Read	1231	BIN	B III 3	G-29
Branch if Supply-Reel Low or Chad-Box Full	1012	BIN	B III 6	E-16
Branch Unconditional		B	B III	B-7
Clear Storage		CS	/ AAA	B-16
Clear Storage and Branch		CS	/ III BBB	B-16
Clear Word Mark (One Address)		CW	□ AAA	B-16
Clear Word Mark (Two Addresses)		CW	□ AAA BBB	B-15
Coded Halt		H	• CCC CCC	B-18
Coded Halt		H	• Cbb Cbb	B-18
Compare		C	C AAA BBB	B-9
Control Carriage	1403	CC	F d	D-6
Control Carriage	1443	CC	F d	D-24
Control Carriage	1445	CC	F d	D-33
Control Carriage Selective Tape Listing Feature	1403	CC	F d	H-58
Control Request	7740	SS	K F	G-27
Control Request and Branch	7740	SSB	K III F	G-27
Diagnostic Read	7335	CU	U %Un A	E-4
Disable Interrupt	EXPS	SS	K <	H-19
Disable Interrupt	1026	SS	K <	G-46
Disable Interrupt	1448	SS	K <	G-22
Disable Interrupt and Branch	EXPS	SSB	K III <	H-19
Disable Interrupt and Branch	1026	SSB	K III <	G-46
Disable Interrupt and Branch	1448	SSB	K III <	G-23
Disengage MICR Reader	1412	CU	U %S1 D	G-7
Divide	MD	D	% AAA BBB	H-33
Edit		MCE	E AAA BBB	B-19
Enable Interrupt	EXPS	SS	K >	H-18
Enable Interrupt	1026	SS	K >	G-45
Enable Interrupt	1448	SS	K >	G-22
Enable Interrupt and Branch	EXPS	SSB	K III >	H-19
Enable Interrupt and Branch	1026	SSB	K III >	G-45
Enable Interrupt and Branch	1448	SSB	K III >	G-23
End Response	7740	SS	K E	G-27
End Response and Branch	7740	SSB	K III E	G-27
Engage MICR Reader	1412	CU	U %S1 E	G-7
Go to Next Line	1285	SS	K G	G-39
Go to Next Line and Branch	1285	SSB	K III G	G-39
Halt		H	•	B-17
Halt and Branch		H	• III	B-18
Initialize a Message Transmission (XMIT)	1009	CU	U %D1 E	H-53
Initialize a Reply from Receiver (RCV)	1009	CU	U %D1 D	H-53
Issue IBM 1026 Line Bid	1026	None	U 5	G-44
Load Character from the Receiving 1009	1009	LU	L %D1 BBB R	H-54
Load Character to the Transmitting 1009	1009	LU	L %D1 BBB W	G-4
Load Character to the Transmitting 1009	1009	LU	L %D1 BBB W	H-54
Load Characters from the Receiving 1009	1009	LU	L %D1 BBB R	G-4
Load Characters to A-Field Word Mark (One Field)		MLCWA	L AAA	B-14
Load Characters to A-Field Word Mark (Two Fields)		MLCWA	L AAA BBB	B-13
Load Characters to Record Mark or GMWM	TRAN	MRCWG	P AAA BBB >	H-51
Load from MICR Reader	1412	LU	L %S1 BBB R	G-8
Load Record	TRAN	MRCWG	P AAA BBB >	H-51
Mark a Line	1285	SS	K M	G-40
Mark a Line and Branch	1285	SSB	K III M	G-40
Modify Address (One Address)		MA	# AAA	A-15
Modify Address (Two Addresses)		MA	# AAA BBB	A-15
Move	1231	MU	M %S3 BBB R	G-30
Move Character from the Receiving 1009	1009	MU	M %D1 BBB R	H-54

Instruction	Type	Mnemonic	Format	Page
Move Character to the Transmitting 1009	1009	MU	M %D1 BBB W	G-3
Move Character to the Transmitting 1009	1009	MU	M %D1 BBB W	H-53
Move Characters and Edit		MCE	E AAA BBB	B-19
Move Characters and Suppress Zeros		MCS	Z AAA BBB	B-11
Move Characters from the Receiving 1009	1009	MU	M %D1 BBB R	G-3
Move Characters to A- or B-Field Word Mark (One Field)		MLC	M AAA	B-11
Move Characters to A- or B-Field Word Mark (Two Fields)		MLC	M AAA BBB	B-11
Move Characters to Record Mark or GMWM		MRCM	P AAA BBB	B-12
Move from MICR Reader	1412	MU	M %S1 BBB R	G-8
Move Numeric		MLNS	D AAA BBB	B-13
Move Record		MRCM	P AAA BBB	B-12
Move Zone		MLZS	Y AAA BBB	B-13
Multiply	MD	M	@ AAA BBB	H-32
No Operation		NOP	N	B-17
Operate in Load Mode	1009	SS	K L	H-56
Operate in Load Mode and Branch	1009	SSB	K III L	H-56
Punch and Feed	1442	P	M %Gn BBB G	D-11
Punch and Stop	1442	PS	M %Gn BBB P	D-11
Punch Card	1444	P	M %G3 B01 G	D-27
Punch Card Image and Feed	1442	PCB	M %Gn BBB G	H-60
Punch Card Image and Stop	1442	None	M %Gn BBB P	H-60
Punch-Column Skip	1442	PSK	M %Gn nnn C	H-61
Read	7740	MU	M %A1 BBB R	G-28
Read Binary Tape	7335	RTB	M %Bn BBB R	E-6
Read Card	1442	R	M %Gn BBB R	D-10
Read Card Image	1442	RCB	M %Gn BBB R	H-59
Read Data	DDC	MU	M %H1 BBB R	H-5
Read Data with Word Marks	DDC	LU	L %H1 BBB R	H-5
Read Disk Sectors	1301	RD	M %F1 BBB R	F-4
Read Disk Sectors	1311	RD	M %F1 BBB R	F-21
Read Disk Sectors with Word Marks	1301	RDW	L %F1 BBB R	F-5
Read Disk Sectors with Word Marks	1311	RDW	L %F1 BBB R	F-23
Read Disk-Track Record	1301	RDTR	M %F2 BBB R	H-40
Read Disk-Track Record	1311	RDTR	M %F2 BBB R	H-44
Read Disk-Track Record with Address	1301	RDTA	M %F@ BBB R	H-41
Read Disk-Track Record with Address	1311	RDTA	M %F@ BBB R	H-44
Read Disk-Track Record with Address with Word Marks	1301	RDTAW	L %F@ BBB R	H-41
Read Disk-Track Record with Address with Word Marks	1311	RDTAW	L %F@ BBB R	H-44
Read Disk-Track Record with Word Marks	1301	RDTRW	L %F2 BBB R	H-40
Read Disk-Track Record with Word Marks	1311	RDTRW	L %F2 BBB R	H-44
Read Disk-Track Sectors with Addresses	1301	RDT	M %F6 BBB R	F-9
Read Disk-Track Sectors with Addresses	1311	RDT	M %F6 BBB R	F-27
Read Disk-Track Sectors with Addresses and Word Marks	1301	RDTW	L %F6 BBB R	F-9
Read Disk-Track Sectors with Addresses and Word Marks	1311	RDTW	L %F6 BBB R	F-27
Read Disk with Sector-Count Overlay	1301	RDCO	M %F5 BBB R	F-5
Read Disk with Sector-Count Overlay	1311	RDCO	M %F5 BBB R	F-23
Read Disk with Sector-Count Overlay with Word Marks	1301	RDCOW	L %F5 BBB R	F-5
Read Disk with Sector-Count Overlay with Word Marks	1311	RDCOW	L %F5 BBB R	F-23
Read from IBM 1026	1026	RCP	M %T0 BBB R	G-45
Read from IBM 1026 with Word Marks	1026	RCPW	L %T0 BBB R	G-45
Read from Paper Tape	1011	MU	M %P1 BBB R	E-9
Read from Paper Tape with Word Marks	1011	LU	L %P1 BBB R	E-10
Read from Primary	EXPS	MU	M %O1 BBB R	H-23
Read from Primary with Word Marks	EXPS	LU	M %O1 BBB R	H-24
Read from 1447 Console	1447	RCP	M %T0 BBB R	C-1
Read from 1447 Console with Word Marks	1447	RCPW	L %T0 BBB R	C-1
Read in Load Mode	1285	LU	L%V1 BBB R	G-39
Read in Move Mode	1285	MU	M %V1 BBB R	G-38
Read Request	DDC	SS	K C	H-2
Read Request	7740	SS	K C	G-27
Read Request and Branch	DDC	SSB	K III C	H-2
Read Request and Branch	7740	SSB	K III C	G-27
Read Tape	7335	RT	M %Un BBB R	E-1
Read Tape with Word Marks	7335	RTW	L %Un BBB R	E-2
Read with Word Marks	7740	LU	L %A1 BBB R	G-28
Reset	DDC	SS	K E	H-3
Reset and Branch	DDC	SSB	K III E	H-3
Rewind Tape	7335	RWD	U %Un R	E-5
Rewind Tape and Unload	7335	RWU	U %Un U	E-5

Instruction	Type	Mnemonic	Format	Page
Scan	1448	None	O III	G-20
Scan Disk Equal	SCAN	SDE	M %F8 BBB W	H-39
Scan Disk Equal with Word Marks	SCAN	SDEW	L %F8 BBB W	H-39
Scan Disk High	SCAN	SDH	M %F9 BBB W	H-39
Scan Disk High with Word Marks	SCAN	SDHW	L %F9 BBB W	H-39
Scan Disk Low	SCAN	SDL	M %F7 BBB W	H-39
Scan Disk Low with Word Marks	SCAN	SDLW	L %F7 BBB W	H-39
Seek Disk	1301	SD	M %F0 BBB R	F-4
Seek Disk	1311	SD	M %F0 BBB R	F-19
Seek Disk with Word Marks	1301	LU	L %F0 BBB R	F-4
Select Digital Time Unit	1026	None	U 0	G-44
Select IBM 1026 on Line 1	1026	None	U 6	G-44
Select IBM 1026 on Line 2	1026	None	U 7	G-44
Select IBM 1026 on Line 3	1026	None	U 8	G-44
Select IBM 1026 on Line 4	1026	None	U 9	G-44
Select Request-Service Indicator	1026	None	U #	G-44
Select Stacker	1231	SS	K A	G-30
Select Stacker	1442	SS	K 2	D-12
Select Stacker	1442	SS	K 0	D-12
Select Stacker	1444	SS	K #	D-27
Select Stacker and Branch	1231	SSB	K III A	G-30
Select Stacker and Branch	1412	SSB	K III d	G-9
Select Stacker MICR Reader	1412	SS	K d	G-9
Select Stacker (Selective-Stacker Feature)	1442	SS	K 2	H-61
Send Attention Signal	EXPS	SS	K A	H-16
Send Attention Signal and Branch	EXPS	SSB	K III A	H-17
Send End of File (EOF)	1009	SS	K F	H-56
Send End of File (EOF) and Branch	1009	SSB	K III F	H-56
Send End Signal	EXPS	SS	K E	H-17
Send End Signal and Branch	EXPS	SSB	K III E	H-17
Send Unusual-End Signal	EXPS	SS	K F	H-18
Send Unusual-End Signal and Branch	EXPS	SSB	K III F	H-18
Sense Request	7740	SS	K G	G-27
Sense Request and Branch	7740	SSB	K III G	G-27
Set Correction Mode	1285	SSB	K III C	G-39
Set Direction to Receive	1009	SS	K D	H-56
Set Direction to Receive and Branch	1009	SSB	K III D	H-56
Set Direction to Transmit	1009	SS	K E	H-56
Set Direction to Transmit and Branch	1009	SSB	K III E	H-56
Set Ready to Receive	1009	CU	U %D1 D	G-3
Set Word Mark (One Address)		SW	, AAA	B-15
Set Word Mark (Two Addresses)		SW	, AAA BBB	B-15
Signal Control	7740	SS	K d	G-27
Signal Control and Branch	7740	SSB	K III d	G-27
Skip and Blank Tape	7335	SKP	U %Un E	E-4
Start Transmission	1009	CU	U %D1 E	G-3
Store A-Address Register	INDX	SAR	Q AAA	H-29
Store B-Address Register (One Address)	INDX	SBR	H AAA	H-30
Store B-Address Register (Two Addresses)	INDX	SBR	H AAA BBB	H-31
Subtract (One Field)		S	S AAA	B-4
Subtract (Two Fields)		S	S AAA BBB	B-4
Suppress 3-Second Alarm	1009	SS	K A	G-5, H-55
Suppress 3-Second Alarm and Branch	1009	SSB	K III A	G-6, H-55
Tape-Punch Read-Back Check	1012	MU	M %P1 BBB R	E-12
Translate	TRAN	TR	T AAA B00	H-48
Translate with Word Marks	TRAN	TRW	T AAA B00 >	H-48
Write	7740	MU	M %A1 BBB W	G-28
Write Binary Tape	7335	WTB	M %Bn BBB W	E-7
Write Data	DDC	MU	M %H1 BBB W	H-6
Write Data with Word Marks	DDC	LU	L %H1 BBB W	H-6
Write Disk Check	1301	WDC	M %F3 BBB W	F-9
Write Disk Check	1311	WDC	M %F3 BBB W	F-27
Write Disk Check with Word Marks	1301	WDCW	L %F3 BBB W	F-9
Write Disk Check with Word Marks	1311	WDCW	L %F3 BBB W	F-27
Write Disk Sectors	1301	WD	M %F1 BBB W	F-7
Write Disk Sectors	1311	WD	M %F1 BBB W	F-25
Write Disk Sectors with Word Marks	1301	WDW	L %F1 BBB W	F-8
Write Disk Sectors with Word Marks	1311	WDW	L %F1 BBB W	F-25
Write Disk-Track Record	1301	WDTR	M %F2 BBB W	H-41

Instruction	Type	Mnemonic	Format	Page
Write Disk-Track Record	1311	WDTR	M %F2 BBB W	H-45
Write Disk-Track Record with Address	1301	WDTA	M %F@ BBB W	H-42
Write Disk-Track Record with Address	1311	WDTA	M %F@ BBB W	H-46
Write Disk-Track Record with Address with Word Marks	1311	WDTAW	L %F@ BBB W	H-46
Write Disk-Track Record with Address with Word Marks	1301	WDTAW	L %F@ BBB W	H-42
Write Disk-Track Record with Word Marks	1301	WDTRW	L %F2 BBB W	H-41
Write Disk-Track Record with Word Marks	1311	WDTRW	L %F2 BBB W	H-45
Write Disk-Track Sectors with Addresses	1301	WDT	M %F6 BBB W	F-10
Write Disk-Track Sectors with Addresses	1311	WDT	M %F6 BBB W	F-28
Write Disk-Track Sectors with Addresses with Word Marks	1311	WDTW	L %F6 BBB W	F-28
Write Disk-Track Sectors with Addresses with Word Marks	1301	WDTW	L %F6 BBB W	F-10
Write Disk with Sector-Count Overlay	1301	WDCO	M %F5 BBB W	F-8
Write Disk with Sector-Count Overlay	1311	WDCO	M %F5 BBB W	F-26
Write Disk with Sector-Count Overlay with Word Marks	1301	WDCOW	L %F5 BBB W	F-8
Write Disk with Sector-Count Overlay with Word Marks	1311	WDCOW	L %F5 BBB W	F-26
Write Line	1403	W	M %Y1 BBB W	D-3
Write Line	1443	W	M %Y1 B01 W	D-22
Write Line	1445	W	M %Y1 B01 W	D-30
Write Line and Suppress Space	1403	WS	M %Y1 BBB S	D-3
Write Line and Suppress Space	1443	WS	M %Y1 B01 S	D-22
Write Line and Suppress Space	1445	WS	M %Y1 B01 S	D-31
Write on Tape Punch	1012	MU	M %P1 BBB W	E-12
Write on 1447 Console	1447	WCP	M %T0 BBB W	C-2
Write on 1447 Console with Word Marks	1447	WCPW	L %T0 BBB W	C-2
Write Request	DDC	SS	K D	H-2
Write Request	7740	SS	K D	G-27
Write Request and Branch	DDC	SSB	K III D	H-3
Write Request and Branch	7740	SSB	K III D	G-27
Write Tape	7335	WT	M %Un BBB W	E-2
Write Tape Mark	7335	WTM	U %Un M	E-4
Write Tape with Word Marks	7335	WTW	L %Un BBB W	E-3
Write to IBM 1026	1026	WCP	M %T0 BBB W	G-45
Write to IBM 1026 with Word Marks	1026	WCPW	L %T0 BBB W	G-45
Write to Primary	EXPS	MU	M %O1 BBB W	H-24
Write to Primary with Word Marks	EXPS	LU	L %O1 BBB W	H-25
Write with Word Marks	7740	LU	L %A1 BBB W	G-28
Zero and Add (One Field)		ZA	? AAA	B-5
Zero and Add (Two Fields)		ZA	? AAA BBB	B-5
Zero and Subtract (One Field)		ZS	! AAA	B-6
Zero and Subtract (Two Fields)		ZS	! AAA BBB	B-5

Index

E056515MOb

IBM

Technical Newsletter

System General 03

Re: Form No. A24-1499-1
A24-1421-2
This Newsletter No. N24-0354

Date: November 5, 1965

Previous Newsletter Nos. N24-0314 (1419)
N24-0316 (1412)

This newsletter provides additional operating information for the IBM 1412 or IBM 1419 when attached to the IBM System/360.

POWER ON – POWER OFF

A system-error condition will result if power to the 1412/1419 is turned on or off while the System/360 is in operation. The following procedure must be used when power is turned on or off on the 1412/1419:

1. Stop the CPU by pressing the stop key on the CPU console.
2. Set the rate switch on the CPU console to SINGLE CYCLE.
3. Press the start key on the CPU console.
4. Turn power on or off on the 1412/1419.
5. Set the rate switch back to the process position.
6. Press the start key to resume operation.

CHANNEL LINE TERMINATION

If the 1412/1419 is physically the last device on the channel (that is, the channel line terminators are located in the 1412/1419), power to the 1412/1419 must be on continuously whenever the System/360 is operating. The 1412/1419 can be operated in any one of the three available modes.

SORT MODE SELECTION

Changing the 1412/1419 from one mode to another can be done with the System/360 in operation, provided the program in process is not using the 1412/1419. Changing the mode of the 1412/1419 while the CPU program is communicating with the 1412/1419 may cause a System/360 error condition.

IBM Corp., Product Publications Dept., Endicott, N. Y. *13764*

N24-0354 (A24-1499-1) (A24-1421-2)

PRINTED IN U.S.A.

IBM **Technical Newsletter**

File No. 1440-01

Re: Form No. A24-3116-3

This Newsletter No. N24-0392

Date: December 30, 1966

Previous Newsletter Nos. None

Replacement pages for System Operation Reference Manual, IBM 1440 Data Processing System, Form A24-3116-3.

To bring your publication up to date, please replace the following pages with the pages attached to this Newsletter. Changed figures are designated by a bullet (●) to the left of the figure title. Changes to text are indicated by a vertical line (|) next to the affected text.

G37, G38
G39, G40
G41, G42
G42.1, G42.2

Please insert this page to indicate that your publication now includes the modified pages issued with this Technical Newsletter.

IBM Corp., Product Publications Dept., Endicott, N. Y. 13760

PRINTED IN U.S.A.