



Systems Reference Library

IBM 1440 System Component Description Special Features

This manual contains a detailed description of the special features that are available for the 1440 Data Processing System. A list of instruction times for the instructions pertinent to the special features and a character coding chart are also included.

Address comments regarding this publication to:
IBM, Product Publications Dept., San Jose, Calif.

CONTENTS

SPECIAL FEATURES	5	Card Image Feature	15
Multiply-Divide	5	Branch If Bit Equal	15
Multiply	5	Selective Stacker — Model 1 Card Read Punch	16
Divide	6	Selective Character Set Feature	16
Expanded Print Edit Feature	8	Additional Printing Positions	16
Asterisk Protection	8	Print Storage	16
Floating Dollar Sign	8	Console Printer, 1447 Model 2	17
Sign Control Left	9	Scan Disk	17
Decimal Control	9	Direct Seek	18
Sense Switch Feature	10	Track Record	18
Branch If Indicator On	10	Serial Input/Output Adapter	19
Increased Core Storage	11	IBM 1412 Magnetic Character Reader	19
Addressing	11	APPENDIX A	20
Address Arithmetic	12	APPENDIX B	21
Modify Address	12		
Modify Address (Single Address)	12		
Indexing and Store Address Register Feature	13		
Indexing	13		
Store Address Register	13		
Store A-Address Register	13		
Store B-Address Register	14		



IBM 1440 Data Processing System

The special features that can be added to the IBM 1440 Data Processing System when application coverage or processing conditions require greater program flexibility and faster program execution are described in this section. The Instruction times for these features is given in Appendix A.

Each special feature contributes to the efficiency of the total system by providing one or more of these advantages:

1. Increased speed
2. Simplified programming
3. Additional system capacity
4. Simplified system organization
5. System compatibility of the IBM 1440 with other IBM data processing systems

MULTIPLY-DIVIDE

Although the IBM 1440 system can be programmed to perform multiplication and division, it is advantageous in applications requiring a significant amount of these types of calculations to use the multiply-divide feature because the use of this feature speeds processing and conserves storage space. This feature makes it possible to perform direct multiplication and division with the IBM 1440 system.

MULTIPLY

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>	<u>B address</u>
M	@	xxx	xxx

The multiplicand (data located in the A field) is repetitively added to itself in the B field, where the multiplier is stored in the leftmost positions and where additional positions (rightmost) are allowed for the development of the product. At the end of the multiply operation, the units position of the product is at the B address, and the multiplier in the B field is destroyed as the product is developed. Therefore, if the multiplier is needed for subsequent operations, it must be retained in another storage area.

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
<u>x 543</u>	3-digit multiplier
	<u>+1</u>
	8 positions must be allowed in the B field.

2. A word mark must be associated with the leftmost positions of both the multiplier and multiplicand fields.
3. A and B bits need not be present in the rightmost 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 will have zone bits in the rightmost position of only the product. The multiply operation uses algebraic sign control (see Figure 1).
4. Zone bits that appear in the multiplicand field are undisturbed by the multiply operation. Zone bits in the rightmost position of the multiplicand are interpreted for sign control.

NOTE: 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 rightmost position of the multiplier.

The A-address and the B-address registers indicate the positions within the A and B fields on which operations are currently being performed.

Multiplier Sign	+	+	-	-
Multiplicand Sign	+	-	+	-
Sign of Product	+	-	-	+

Figure 1. Algebraic Sign Control for Multiplication

DIVIDE

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>	<u>B address</u>
D	%	xxx	xxx

This instruction divides a dividend located in the rightmost positions of the B field, by a divisor located in the A field, and develops the quotient in the leftmost positions of the B field. The remainder will be in the low-order positions of the B field.

A word mark must define the leftmost position of the divisor.

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 the number of digits in the 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 located in the leftmost position of the A field (divisor).
3. In all cases, A and B bits (plus sign) or B bit (minus sign) must appear in the rightmost position of the dividend field. Zone bits must not appear in other than the rightmost position of the dividend field. The divisor may be either signed or unsigned. If there are no

Divisor Sign	+	+	-	-
Dividend Sign	+	-	+	-
Quotient Sign	+	-	-	+
Remainder Sign	+	-	+	-

Figure 2. Algebraic Sign Control for Division

bits in the rightmost position of the divisor, the machine assumes the divisor factor is positive. The divide operation uses algebraic sign control (Figure 2).

4. The dividend is loaded in the rightmost positions of the B field (Figure 3) by a ZERO AND ADD instruction to ensure that zeros are present in the leftmost positions of the B field and to place the standard sign notation in the units position.
5. The B address in the DIVIDE instruction specifies the leftmost position of the dividend, within the B field.

At the completion of division:

- a. The quotient is in the leftmost positions of the B field. The location of the rightmost position of the quotient is the address of the rightmost position of the dividend, minus the length of the divisor, minus one.
- b. The remainder is in the rightmost positions of the B field.
- c. The sign of the quotient is over the rightmost position of the quotient field.
- d. Because only one quotient digit can be developed at a time, it is important to address the leftmost position of the dividend (B address of the DIVIDE instruction). This ensures that the first divide operation will result in a single leftmost 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 assigned is not large enough, the overflow that results will not be indicated. This type of overflow is a programming error for which the machine does not check. Division by zero results in an arithmetic overflow condition, and this type of overflow is indicated. Figure 4 shows the result of a divide operation in the B field.

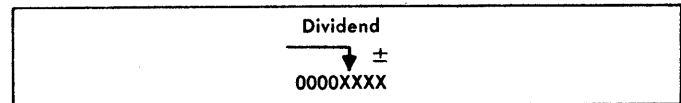


Figure 3. Dividend in B-field

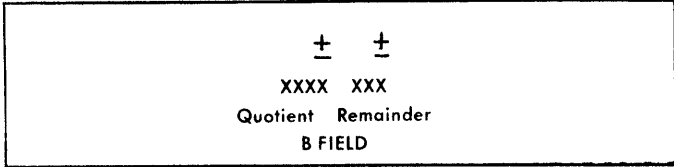


Figure 4. Location of the Results of a Divide Operation

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 5. Note that in this example, the rightmost position of the quotient is not located in the position previously described in AT THE COMPLETION OF DIVISION, paragraph a.

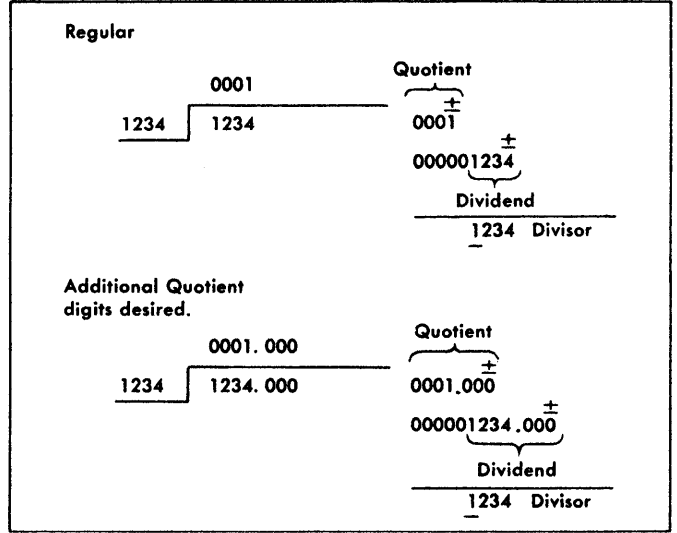


Figure 5. Additional Quotient Digits

EXPANDED PRINT EDIT FEATURE

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.

ASTERISK PROTECTION

The asterisk protection feature (Figure 6) is used when asterisks are to appear to the left of significant digits. 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 rightmost 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 and is then erased, and the operation is stopped.

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 6. Asterisk Protection

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 amount field (Figure 7). The control word is written with the dollar sign 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 dollar sign is sensed.
2. The corresponding digit from the A field replaces the dollar sign that is 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 the dollar sign, and the operation stops.

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 7. Floating Dollar Sign

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 (Figure 8). The control word is written with the CR or minus symbols in the leftmost 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 remain undisturbed in their corresponding positions in the output field 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 9).

Two scans are sufficient to complete this editing operation, unless the field contains no significant digits. In that case, three scans are required.

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 8. Sign Control Left

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 automatically set in this position in the B (output) field.
3. Editing continues normally until the B-field word mark is sensed and removed.

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. If significant digits are present, the edit operation stops.

Second Forward Scan

1. Blanks replace the zeros to the right of the decimal point and the decimal point itself.
2. The operation stops at the decimal column.

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 9. Decimal Control

SENSE SWITCH FEATURE

Seven sense switches on the console make it possible to control the stored program from there (Figure 10). Sense switch A (Last Card operation) is standard and the remaining six are a special feature. The BRANCH IF INDICATOR ON instruction expands to allow the program to test each switch.

For example, a sense switch is turned on if printing only is required for a given job. With the sense switch off, data is both printed and punched.

The program can be written to interrogate the setting of this sense switch to determine if the data is to be printed and punched or only printed. Sense switch settings should not be changed while the system is operating.

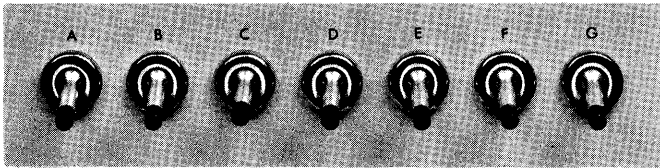


Figure 10. Sense Switches

BRANCH IF INDICATOR ON

<u>Mnemonic</u>	<u>OP code</u>	<u>I address</u>	<u>d character</u>
BSS	<u>B</u>	xxx	x

This instruction tests the position of the toggle switch specified by the d character. If the switch is in the ON position, the next instruction is taken from the I address. If it is off, the program continues with the next sequential instruction.

<u>d character</u>	<u>Branch to the I address if:</u>
B	Sense switch B is on
C	Sense switch C is on
D	Sense switch D is on
E	Sense switch E is on
F	Sense switch F is on
G	Sense switch G is on.

Word marks are not affected. .

INCREASED CORE STORAGE

The 1441, Model A-3, containing 4,000 positions of core storage is a standard feature for the 1440 Data Processing System. The Model A-3 can be replaced by either the Model A-4 containing 8,000 positions of core storage, or the Model A-5 containing 12,000 positions, or the Model A-6 containing 16,000 positions.

ADDRESSING

The additional core storage locations are addressed by the presence of zone bits located over the right-most position of each storage address over 3999. These zone bits are added to the basic 4,000-character coding system.

<u>Storage Addresses</u>	<u>Zone Bits over Units Positions of Address</u>
4000-7999	A
8000-11999	B
12000-15999	A B

Storage addresses 0000-3999 have zone bits over the hundreds positions, and no zone bits over the units position of the 3-character address.

Storage addresses 4000-7999 have zone bits over the hundreds position, and A bits (0 zone) over the units position of the 3-character address.

Storage addresses 8000-11999 have zone bits over the hundreds position, and B bits (11 zone) over the units position of the 3-character address.

Storage addresses 12000-15999 have zone bits over the hundreds position, and A and B bits (12 zone) over the units position of the 3-character address. Figure 11 is a chart of the addressing system.

The IBM 1440 addresses core-storage locations by assigning a digit value to each bit that appears over the hundreds and units positions of the 3-character address.

<u>Bit and Location</u>	<u>Digit Value</u>
A bit over <u>hundreds</u> position	1
B bit over <u>hundreds</u> position	2
A bit over <u>units</u> position	4
B bit over <u>units</u> position	8

The machine adds the assigned digit values of the A and B bits located over the hundreds and units positions in order to determine the thousands positions of the address. For example, to address core stor-

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

Figure 11. Addressing System

age locations, the IBM 1440 assigns digit values to the bits, as follows:

A 1
 ?99 = B = 2 1 + 2 = 3
 099 099

Address: 3099

R9Z=B A = 2 4 2 + 4 = 6
 999 999

Address: 6999

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

<u>Location</u>	<u>Contents</u>	<u>3-Character Address</u>	<u>Actual Address</u>
A address	A field	100	100
B address	B field	D2U	7424
Result in: B field		E2U	7524

and:

A address	A field	100	100
B address	B field	L2F	14326
Result in: B field		M2F	14426

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.

MODIFY ADDRESS (SINGLE ADDRESS)

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>
MA	#	xxx

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

Address Validity

The IBM 1440 checks each address to ensure that it is valid for the storage capacity installed. The system stops on an address validity error if an invalid address is encountered.

<u>Core Storage Capacity</u>	<u>Valid Addresses</u>	<u>Invalid Addresses</u>
4,000	0000-3999	4000-15999
8,000	0000-7999	8000-15999
12,000	0000-11999	12000-15999
16,000	0000-15999	NONE

ADDRESS ARITHMETIC

To facilitate address arithmetic, an additional operation code (MODIFY ADDRESS) is added to IBM 1440 systems that are equipped with more than 4,000 characters in core storage.

MODIFY ADDRESS

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>	<u>B address</u>
MA	#	xxx	xxx

INDEXING AND STORE ADDRESS REGISTER FEATURE

INDEXING

Many IBM 1440 programs require that the same operations be performed repetitively, with only a change in the A or B address. Modifying these core storage addresses each time a repetitive operation is performed requires several program steps, and the setting aside of additional storage locations for this purpose.

When the indexing feature is installed, three index locations are provided in the IBM 1441 Processing Unit for automatic modification of addresses. With this feature, fewer instructions are needed and storage space is conserved, resulting in faster program execution and over-all simplification of the programming effort.

Index Location Addresses

Addresses assigned to the index locations are:

<u>Index location</u>	<u>Storage</u>	<u>Addresses</u>
1	087-089	089
2	092-094	094
3	097-099	099

Tagging

Each address must be tagged by an indicator so the 1440 knows which of the three index locations is to be used to modify the instruction address.

A combination of A and B bits in the tens position of the A or B address of an instruction (AAA) (BBB), selects the index location.

<u>Index location</u>	<u>Tens positions</u>	<u>Zone punch</u>
1	A bit, No B bit	Zero
2	B bit, No A bit	Eleven
3	A bit, B bit	Twelve

Address Modification

The primary use of index locations is to automatically modify addresses by adding (absolute) the contents of an index location to an address. Both the A address and the B address of an instruction, can be modified by the contents of any index location, but only core storage addresses can be modified.

The modification of the A and B addresses 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.

The index locations can be used as normal storage positions when not being used as index locations.

The index factor can be placed in the index location by normal programming, such as with an ADD or MOVE operation. The index factor can be changed by normal arithmetic ADD and SUBTRACT instructions, following the word mark rules for those instructions. In such cases, a word mark should be set in the highorder position of the index location by an initialization routine.

STORE ADDRESS REGISTER

This special feature makes it possible to store the contents of the A and B address registers. It 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.

A subroutine is a set of program instructions that is 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 the machine upon leaving the sequence of the main program can execute the steps of the subroutine and then return to the main program at the point where the main sequence was interrupted.

STORE A-ADDRESS REGISTER

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>
SAR	<u>Q</u>	xxx

This instruction stores the contents of the A-address register from the previous operation in the three-position field that has its units position defined by the A address of the STORE A-ADDRESS REGISTER instruction. Word marks are not affected.

STORE B-ADDRESS REGISTER

<u>Mnemonic</u>	<u>OP code</u>	<u>A address</u>
SBR	<u>H</u>	xxx

This instruction stores the contents of the B-address register resulting from the previous operation in the three-position field that has its units position defined by the A address of the STORE B-ADDRESS REGISTER instruction. Word marks are not affected.

NOTE: If the Store Address Feature is installed in the IBM 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 automatically inserted in the B-address register. See Note under Logic Operations.

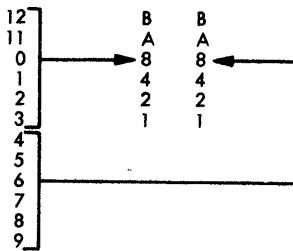
Although the subroutine may be entered from many distinct points in the main program, this use of the SBR operation makes the subroutine linkage complete.

CARD IMAGE FEATURE

Mnemonic OP code A address B address d modifier

RCB	<u>M</u>	%Gn	xxx	R
PCB	<u>M</u>	%Gn	xxx	P

This feature provides the facility to read punched data into the Processing Unit without the normal translation from Hollerith codes 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.



A Card Image operation is indicated by the presence of a 9 or 0 (zero) in the n position of the A field in a READ or PUNCH instruction. Since this position normally selects the Read Punch, the #1 unit is selected by the number 9 in lieu of a 1; the #2 unit is selected by the number 0.

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, the Card Image read operation begins in the normal reading mode. Each card column read is translated and stored in single core locations until a word mark is detected in core storage. This signals the mode of reading to change. Translation from Hollerith is suspended. Suspension causes the card data beginning at the word mark to be stored in two adjacent core positions for each column read. Detection of another word mark causes

the reading to revert to the normal reading mode. The operation is terminated when a group mark-word mark is sensed.

For the Card Image PUNCH instruction, data is punched from two core positions into each card column without the normal translation from BCD code to Hollerith code. Punching starts with the position specified by the B address and it is terminated by the group mark-word mark in the rightmost position of the field. Consecutive Punch instructions (Punch Card and/or Card Image Punch) can be given to punch a card.

Examples:

- a. Read 80 columns in Card Image

Instruction: M % G 9 XXX R
 Word Mark: at XXX
 GM-WM: XXX + 160

- b. Punch 5 columns Hollerith, 20 columns Card Image Mode

Instruction: M % G 1 XXX P
 GM-WM: XXX + 5
 Instruction: M % G 9 XXX P
 GM-WM: XXX + 40

BRANCH IF BIT EQUAL

Mnemonic OP code I address B address d character

BBE	<u>W</u>	xxx	xxx	x
-----	----------	-----	-----	---

The d character can contain any character or any combination of bits that can exist in a single position of the 1440 core storage. If the character at the B address contains any data bit that matches any data bit in the d character (regardless of the number and kind of other bits present), the program branches to the I address. Otherwise, the program continues normally. Word marks are not affected.

**SELECTIVE STACKER FEATURE-MODEL 1 CARD
READ PUNCH**

A second stacker is available for the 1442, Model 1. A SELECT STACKER instruction is provided to enable cards to be directed to the second stacker. Cards not selected will be directed to the standard stacker.

SELECTIVE CHARACTER SET FEATURE

This feature provides the controls necessary to allow the printer to use other than the standard 52-character set. It is a prerequisite for the use of any of the three optional sets available. A four-position switch is provided which allows the operator to designate which of the character sets he is using: 13, 39, 52, or 63. Appendix B is the Character Coding Chart which shows the contents of each set. The printing speed of each character set is shown in Figure 12.

ADDITIONAL PRINTING POSITIONS

Twenty-four additional print positions can be installed on the 1443 Printer to increase the line width to a total of 144 positions. These additional positions are installed, 13 to the left and 11 to the right of the standard 120 positions.

Number of Characters	Print Rate (LPM)	Character Set (Type Arrangement A)
13	430	10 numerical characters 3 Special Characters: . - *
39	190	26 alphabetic, 10 numerical 3 Special Characters: . \$,
52*	150	26 alphabetic, 10 numerical 16 Special Characters: & . - \$ * , % / # @ ? ! # : ;
63	120	26 alphabetic, 10 numerical 27 Special Characters (16 shown above plus) [< #] ; Δ √ \ * > √

* Standard

Figure 12. Additional Character Set Specifications

PRINT STORAGE

This feature provides 120 or 144 nonaddressable extra positions of core storage for temporary storage of data to be printed. Without this feature, the processing unit must delay operation on data while other data is being printed. This arrangement is known as interlocking the processing unit.

With this feature installed, nearly all normal interlock time is eliminated, the processing unit being interlocked only when data is automatically transferred from regular storage to special print storage positions. Once the 1441 has completed the automatic transfer of data to these extra positions of storage, it is free to perform other operations because actual printing involves only the print storage area and the 1443 Printer. Thus, available processing time is greatly increased for applications characterized by high printing requirements.

The instruction to control printing with the Print Storage Feature is: M%Y1 BBB W; and the assignment of the B Area is no longer constrained to zero one bands of storage. A GM-WM must be positioned to designate the end of the print line.

If the Printer receives another Print instruction while it is printing from print storage, it will interlock with the print storage area. Processing, however, may continue. A Printer Busy indicator is included with the Print Storage Feature to enable the programmer to determine before a Print instruction is given if the Printer is busy. If the indicator is off (Printer not busy), the Print instruction may be taken. If the indicator is on, (Printer is busy with a previous print instruction) other processing can be done, and the program can then return to test the indicator and continue with the Print instruction. If this programming flexibility is not required, Print instructions are given without first testing the Printer Busy indicator. In this case, if the Printer is busy when another Print instruction is given, the program halts temporarily until the Printer completes the previous print operation. When the previous print operation is completed, the program continues with the next print instruction.

CONSOLE PRINTER, 1447 MODEL 2

The Console Printer is available to provide communication between the operator and the processing unit and between the operator and the disk storage unit. The Console Printer can be used for direct input and output to the 1440 or as a secondary printer. When the high-speed output of a 1443 Printer is not required, the console printer can be used as the only printer.

The Console Printer can read from, and write into, storage. It permits the 1440 system operator to either examine or alter the status of a particular account, record, or instruction stored in the system.

This feature is especially useful in obtaining information in the disk file, such as customers' accounts, stock status data, and payroll details. Through the Console Printer, the operator can request specific data from any disk record and have the information typed out immediately. This typed copy also serves as a log of information entered into or received from the 1440 system.

The Console Printer also operates in conjunction with the Alter and Character Display modes of the Console Mode switch.

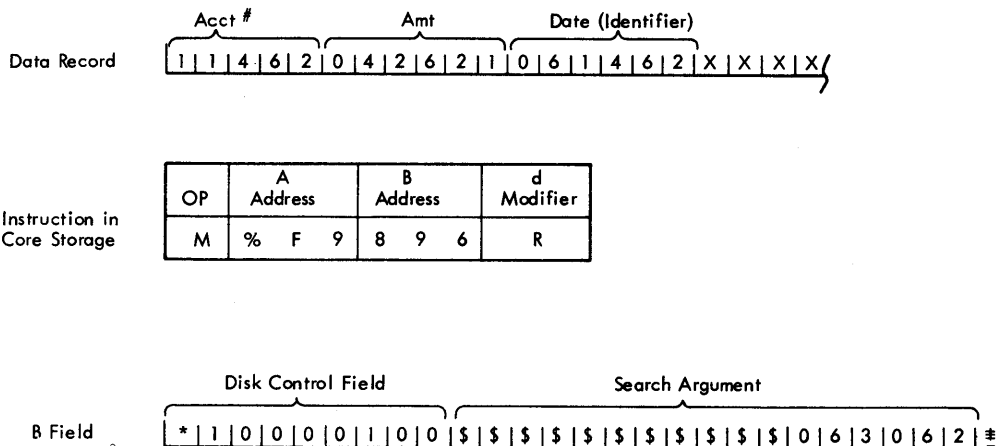
The Console Printer has character spacing of 10 characters per inch. This spacing provides 85 characters per line.

SCAN DISK

Scan Disk provides an automatic rapid search of file information for a specific identifier or condition stored within the data itself. The Disk Control Field initially is established to reflect the sector address where the scan is to begin and the maximum number of sectors to be scanned. The search argument area adjacent to the Disk Control Field is established in the format of the sectors being searched. The search argument located in this area must correspond to the positions in which the identifier is located in the disk record. Fields in the search argument area corresponding to those in the disk record that are not to be considered in the scan, must be filled with Scan Skip signals (\$). A GM-WM must be placed after the last position of the argument. The last character of a sector cannot be included in the scan argument. Word marks are not compared in either 7 bit or 8 bit mode. The scan operation can only be performed in the sector mode of operation.

Each sector is examined in sequence until the search argument is satisfied or the Sector Count reaches zero. The Core Sector Address segment in the Disk Control Field contains the address of the sector satisfying the scan condition. The address is then inserted in a separate Disk Read instruction in order to read the data record into core storage.

The drawing below illustrates how the search argument area is formed.



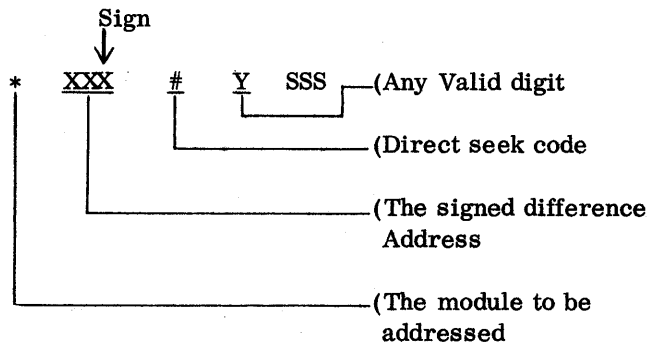
Scan can be accomplished on the basis of Equal, High/Equal, or Low/Equal as the argument relates to disk data. For example, scanning for Low/Equal means that any data records in the portion of the file being searched are selected when the search argument in core storage is higher than or equal to the data identifier in disk storage (see previous example).

These operations are dependent upon the high-low-equal compare function. The result of the scan is determined by testing the compare indicators with a Branch If Indicator On instruction. Scan is confined to the sector mode of operation.

DIRECT SEEK

This feature reduces access time by allowing an access mechanism to be repositioned directly to a new setting without first having returned to the home position. A program generated number that is equal to twice the cylinder difference between the new address and the previous address controls the Direct Seek Feature.

The Direct Seek Feature is specified by placing a number sign in the normal seek address, as illustrated:



A Method of Calculating the Signed Difference Address

Assume the two addresses shown for an example:

New 04372 Old 03291

1. Drop the two low-order positions of both addresses.

New 043 Old 032

2. Be sure that the low-order digits of both addresses are either odd or even.

New 043 Old 032

3. To make both low-order digits either odd or even, add one to the even digit or subtract one from the odd digit.

New 043 Old 032
 $\begin{array}{r} -1 \\ \hline 042 \end{array}$ $\begin{array}{r} +1 \\ \hline 033 \end{array}$

Assume the "add one" approach is used. Therefore,

New remains 043 and old becomes 33.

4. Subtract the old address from the new address

043
 $\begin{array}{r} -033 \\ \hline -010 \end{array}$

If the old address had been larger than the new address, the result would be:

033
 $\begin{array}{r} -043 \\ \hline -010 \end{array}$

The sign of the low-order digit in the result will be considered plus unless a minus sign is indicated. The minus sign is the B bit only.

TRACK RECORD

This feature provides the ability to write and read one record per track under control of the program. The track is comprised of one sector address in 7 bit mode plus 2980 7 bit characters (Move mode) or 2682 8 bit characters (Load mode). Records assembled in Track Record Mode can be intermixed within a pack of records assembled in the Sector mode. On any one disk surface, tracks written in the Track Record mode must be separated by one unused track from tracks written in the Sector mode. Transfer of data that is stored as a Track Record first requires comparison of the core sector address against the disk sector address as with other disk transfer operations. Track Records cannot be scanned.

SERIAL INPUT/OUTPUT ADAPTER

This device is used to attach special-purpose machines to the the 1440 Data Processing System.

IBM 1412 MAGNETIC CHARACTER READER

The Serial Input/Output adapter makes it possible to have direct input from the IBM 1412 Magnetic Character Reader (Figure 13) to the IBM 1440 Data Processing System.

The addition of the 1412 to the 1440 system provides the banking industry with a fast, direct, and accurate method for reading and processing large volumes of daily transactions.

Documents read by the 1412 are selected into stackers specified by the 1441 stored program, while the data from these documents is entered into 1441 core storage for processing. Thus, uninterrupted data flow from source document to final output is effected.

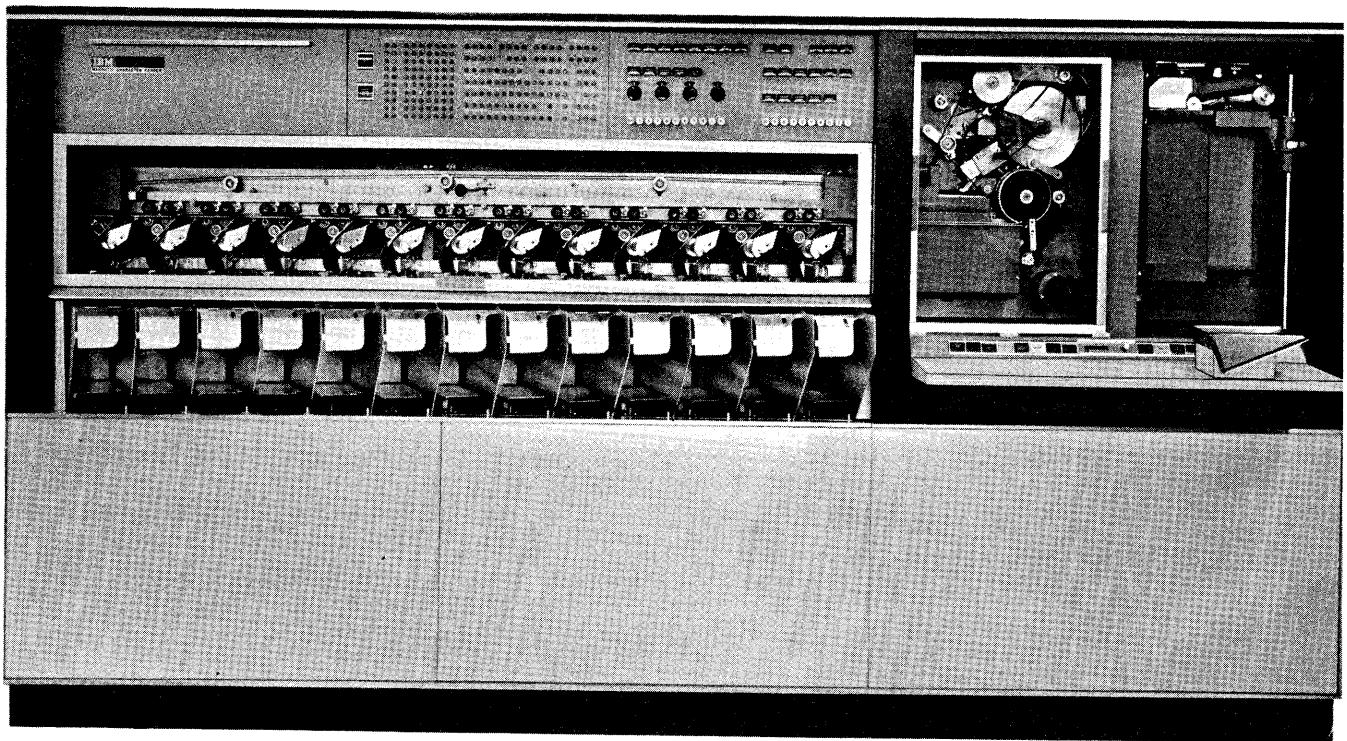


Figure 13. 1412 Magnetic Character Reader

APPENDIX A INSTRUCTION TIMES

Instruction	TIMING (ms) .0111 x (L _I +)	Address Registers after Operation		Notes
		A address	B Address	
Branch If Bit Equal	2)	BI	B-1	
Branch If Indicator On	1) or **2)	BI	BBB **NSI	
Divide	2+7L _R L _Q +8L _Q)	A minus length of divisor	tens position of quotient	
*Modify Address	9)	A-3	B-1 or B-3	
*Modify Address (one address)	9)	A-3	A-1 or A-3	
Move Characters and Edit	1)+L _A +L _B +L _Y	A address minus length of A field	B-L _B	Without zero suppression
Move Characters and Edit	1)+L _A +L _B +L _Y	A address minus length of A field	Location of control zero plus 1	With zero suppression
Multiply	3+2L _C +5L _C L _M +7L _M)	A minus length of multiplicand	B minus length of product field	Average time
Print (Print Storage)	Total Time = 2.4 ms	---	---	
Read Console Printer	1)	%TO	B+L _B +1	
Read Track Record	1)+62ms	B+8	See 1440-07	
Stacker Select	1)	---	---	
Store A Address Register	5)	A-3	A _p	
Store B Address Register	5)	A-3	B _p	
Write Console Printer	1)+68(L _B)+800(CR-1)	%TO	B+L _B +1	CR = Number of print element returns
Write Track Record	1)+62ms	B+8	See 1440-07	

**If the Store Address Register feature is installed

Key to abbreviations used in formulas:	
L _A = Length of the A field	L _Y = Number of characters back to rightmost zero in control field.
L _B = Length of the B field	L _Z = Number of zeros inserted in a field
L _C = Length of Multiplicand field	BI = Next Instruction if Branch occurs
L _I = Length of Instruction	A = A address of instruction
L _M = Length of Multiplier field	B = B address of instruction
L _Q = Length of Quotient field	A _p = Previous setting of A-address register
L _R = Length of Divisor field	B _p = Previous setting of B-address register
L _S = Number of significant digits in Divisor (excludes high-order zeros and blanks).	NSI = Next Sequential Instruction

APPENDIX B CHARACTER CODING CHART

CARD		BCD		13	39	52	52	63	CARD		BCD		13	39	52	52	63	
DEFINED CHARACTER	CODE		CODE			A	H		DEFINED CHARACTER	CODE		CODE			A	H	63	
Blank		C		X	X	X	X	X	G	12-7	B	A	4	2	1	X	X	X
. Period	12-3-8	B	A	8	2	1	X	X	H	12-8	B	A	8			X	X	X
Lozenge	12-4-8	C	B	A	8	4			I	12-9	C	B	A	8	1	X	X	X
[Left Bracket	12-5-8	B	A	8	4	1			I (- zero)	11-0	B	8	2			X	X	X
< Less Than	12-6-8	B	A	8	4	2			J	11-1	C	B		1		X	X	X
≡ Group Mark	12-7-8	C	B	A	8	4	2	1	K	11-2	C	B		2		X	X	X
& Ampersand	12	C	B	A					L	11-3	B		2	1		X	X	X
\$ Dollar Sign	11-3-8	C	B	8	2	1		X	M	11-4	C	B	4			X	X	X
* Asterisk	11-4-8	B	8	4			X	X	N	11-5	B	4	1			X	X	X
] Right Bracket	11-5-8	C	B	8	4	1			O	11-6	B	4	2			X	X	X
; Semicolon	11-6-8	C	B	8	4	2			P	11-7	C	B	4	2	1	X	X	X
△ Delta	11-7-8	B	8	4	2	1			Q	11-8	C	B	8			X	X	X
- Hyphen	11	B					X	X	R	11-9	B	8	1			X	X	X
/ Diagonal	0-1	C	A		1			X	≠ Record Mark	0-2-8	A	8	2			X	X	X
, Comma	0-3-8	C	A	8	2	1		X	S	0-2	C	A	2			X	X	X
% Percent Mark	0-4-8		A	8	4			% (X	T	0-3	A	2	1			X	X	X
∨ Word Separator	0-5-8	C	A	8	4	1			U	0-4	C	A	4			X	X	X
\ Left Oblique	0-6-8	C	A	8	4	2			V	0-5	A	4	1			X	X	X
≡ Segment Mark	0-7-8		A	8	4	2	1		W	0-6	A	4	2			X	X	X
␣ Substitute Blank	2-8	A						X	X	0-7	C	A	4	2	1	X	X	X
# Number Sign	3-8		8	2	1			# = X	Y	0-8	C	A	8			X	X	X
@ At Sign	4-8	C	8	4				@ ' X	Z	0-9	A	8	1			X	X	X
: Colon	5-8		8	4	1			X	0 (Zero)	0	C	8	2	X	X	X	X	X
> Greater Than	6-8		8	4	2			X	1	1			1	X	X	X	X	
√ Radical	7-8	C	8	4	2	1			2	2			2	X	X	X	X	
? (Plus Zero)	12-0	C	B	A	8	2		X	3	3	C		2	1	X	X	X	
A	12-1	B	A	8	1			X	4	4			4	X	X	X	X	
B	12-2	B	A	2				X	5	5	C	4	1	X	X	X	X	
C	12-3	C	B	A	2	1		X	6	6	C	4	2	X	X	X	X	
D	12-4	B	A	4				X	7	7		4	2	1	X	X	X	
E	12-5	C	B	A	4	1		X	8	8		8		X	X	X	X	
F	12-6	C	B	A	4	2		X	9	9	C	8	1	X	X	X	X	

INDEX

Additional Printing Positions 16
Address Arithmetic 12
Address Modification 13
Address Validity, Increased Core Storage 12
Addressing, Increased Core Storage 11
Asterisk Protection 8

Branch If Bit Equal 15
Branch If Indicator On 10

Card Image Feature 15
Character Coding Chart 21
Console Printer, 1447 Model 2 17

Decimal Control 9
Direct Seek 18
Divide 6

Expanded Print Edit Feature 8

Floating Dollar Sign 8

IBM 1412 Magnetic Character Reader 19

Increased Core Storage 11
Indexing 13
Indexing and Store Address Register Feature 13
Input/Output Adapter 19
Instruction Times 20

Modify Address 12
Multiply 5
Multiply-Divide Feature 5

Print Storage 16

Scan Disk 17
Selective Character Set Feature 16
Selective Stacker Feature 16
Sense Switch Feature 10
Serial Input/Output Adapter 19
Sign Control Left 9
Store A-address Register 13
Store B-address Register 14
Store Address Register 13

Track Record 18





Printed in U. S. A. A26-5669-0



IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York

THE COMPUTER MUSEUM HISTORY CENTER



1 026 1802 0