System/File No. 1410-01
Form A22-0526
Pages Revised 5/1/63

Systems Reference Library

IBM 1410 Principles of Operation

This publication contains the instruction set for the IBM 1410 Data Processing System and includes the formula for calculating the time needed to execute each instruction. The operation code for every instruction is given in both actual and mnemonic form. Examples detailing the results of individual instructions are included where necessary. Addressing, address chaining, and indexing are described, and console controls are explained. Control instructions and timing considerations are given for the IBM 7330 and IBM 729 II, IV, V, and VI Magnetic Tape Units. Control instructions and timing considerations, plus functions and operations, are given for the IBM 1415 Console-I/O Printer, IBM 1402 Model 2 Card Read Punch, and IBM 1403 Printer. The 1410 Accelerator, Processing Overlap, and Data Channel 2 special features are also described.

For information on Disk Storage units or Priority Processing, refer to *IBM 1410 Bibliography,* Form A22-0523.

MAJOR REVISION (October, 1962)

This publication, Form A22-0526, obsoletes *IBM 1410 Data Processing System Reference Manual*, Forms A24-1407-1 and A24-1407-2, and the following Technical Newsletters pertaining to the IBM 1410 Data Processing System:

N22-0030 *1401/1410 Compatibility*
N22-0032 *Processing Overlap Feature*
N22-0045 *1410 Accelerator Feature*
N22-0054 *Changes in IBM 1410 Data Processing System Reference Manual*
N22-0059 *Skip and Blank Tape Operations*

Technical Newsletter N22-0018, *IBM 1414 Model 5 Input-Output Synchronizer*, is also obsolete; its information is in *IBM Telecommunications Equipment with IBM 1410 System*, Form A22-0525.

POST-PUBLICATION CHANGE PAGE HISTORY

Technical Newsletter No. N22-0070 has been incorporated in this manual. Text changes are indicated by a line at the left of the text; figure changes by a bullet at the left of the figure title. The following pages were affected:

OLD PAGE	NEW PAGE
1 (cover)	1 (Revised 1/15/63)
2 (this page)	2 (Revised 1/15/63)
14	14 (Revised 1/15/63)
64	64 (Revised 1/15/63)
86	86 (Revised 1/15/63)
96	96 (Revised 1/15/63)
97	97 (Revised 1/15/63)

Technical Newsletter No. N22-0083 has been incorporated in this manual. Text changes are indicated by a line at the left of the text; figure changes by a bullet at the left of the figure title. The following pages were affected:

OLD PAGE	NEW PAGE
1 (Revised 1/15/63) (cover)	1 (Revised 5/1/63)
2 (Revised 1/15/63) (this page)	2 (Revised 5/1/63)
7	7 (Revised 5/1/63)
8	8 (Revised 5/1/63)
9	9 and 9.1 (Revised 5/1/63)
12	12 and 12.1 (Revised 5/1/63)
19	19 (Revised 5/1/63)
21	21 (Revised 5/1/63)
22	22 (Revised 5/1/63)
23	23 (Revised 5/1/63)
24	24 (Revised 5/1/63)
30	30 (Revised 5/1/63)
33	33 and 33.1 (Revised 5/1/63)
40	40 (Revised 5/1/63)
46	46 (Revised 5/1/63)
61	61 (Revised 5/1/63)
62	62 (Revised 5/1/63)
63	63 (Revised 5/1/63)
78	78 (Revised 5/1/63)
79	79 (Revised 5/1/63)
83	83 (Revised 5/1/63)
84	84 (Revised 5/1/63)
85	85 (Revised 5/1/63)
91	91 (Revised 5/1/63)
92	92 (Revised 5/1/63)
93	93 (Revised 5/1/63)
98 (blank page)	98 (Revised 5/1/63)
99	99 (Revised 5/1/63)
100	100 (Revised 5/1/63)
101	101 (Revised 5/1/63)
102	102 (Revised 5/1/63)
103	103 (Revised 5/1/63)
(Comment sheet added at end of index)	

Copies of this and other IBM publications can be obtained through IBM branch offices.
Address comments concerning the contents of this manual to:
IBM Customer Manuals, Dept. B98, P.O. Box 390, Poughkeepsie, New York

Contents

IBM 1410 Data Processing System

All data entering or leaving the IBM 1410 Data Processing System must pass through the IBM 1411 Processing Unit (Figure 1), where the operations that produce processed results at the 1411 output take place. Processing consists of logic, arithmetic, table lookup, comparing, and print editing operations. Numerous check points in the 1411 assure accuracy and reliability of the input and results. Other data manipulation features of the 1411 are:

1. Alphameric data representation
2. Variable word length
3. Sixty-four ways of moving data
4. Core storage cycle of 4.5 microseconds per character (see "1410 Accelerator")
5. Double-address type instructions
6. Instruction chaining
7. Fifteen index registers

Processing needs determine the core storage capacity of the 1411. Five 1411 models are available:

MODEL	STORAGE POSITIONS
1 or 1A	10,000
2 or 2A	20,000
3 or 3A	40,000
4 or 4A	60,000
5 or 5A	80,000

Models with suffix A have a fifth (E) frame. Basic concepts about the 1410 system and brief descriptions of all system units, including the special units requiring the E frame, are in the *IBM 1410 Systems Summary*, Form A22-0524.

Character Coding

The IBM 1410 Data Processing System stores alphameric characters internally in binary-coded decimal form. The bit configuration of a character is determined by the presence of BA8421 bits and a check bit. When the character used is an operation code or is the first character in a field, another bit, called a word-mark bit, is included. Each character must contain an odd number of bits, called parity, for that character. If the combination of BA8421 bits results in an even-bit configuration, a check bit is added to give the character odd-bit parity. If the character has a word mark, this word mark (WM) is counted in the character configuration before the check-bit status is determined. Each character is checked at various locations in the system to be sure that it has an odd number of bits including the check bit and the word mark.

EXAMPLES

CHARACTER	BA8421 BITS	WORD MARK?	ADD CHECK BIT?	ODD PARITY
A	BA 1	No	No	BA1
Ⓐ	BA 1	Yes	Yes	WMCBA1
C	BA 21	No	Yes	CBA21
Ⓒ	BA 21	Yes	No	WMBA21
X	A 421	No	Yes	CA421
Ⓧ	A 421	Yes	No	WMA421

Figure 2 shows the 64 code points (bit combinations) and their character equivalents that are valid in the IBM 1410. The chart is arranged in ascending collating

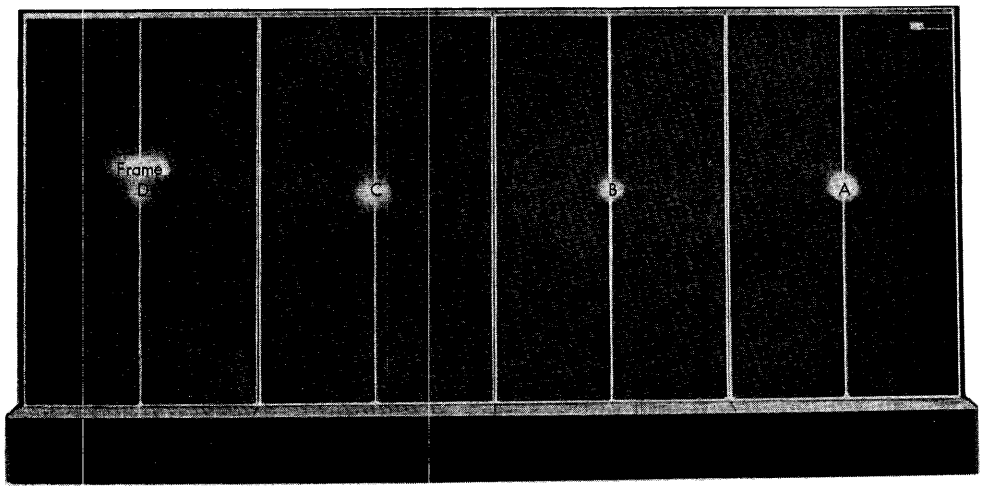


Figure 1. IBM 1411 Processing Unit

	CHARACTER		CARD CODE	BCD CODE (Core Storage)						
	Commerce (Report)	Science (Program)								
(1)	b		No Punches	C						
	•		12-3-8		B	A	8		2	1
	□	)	12-4-8	C	B	A	8	4		
(1)	[		12-5-8		B	A	8	4		1
	<		12-6-8		B	A	8	4	2	
	ǂ		12-7-8	C	B	A	8	4	2	1
	&	+	12	C	B	A				
	$		11-3-8	C	B		8		2	1
	*		11-4-8		B		8	4		
(1)	]		11-5-8	C	B		8	4		1
	;		11-6-8	C	B		8	4	2	
	Δ		11-7-8		B		8	4	2	1
	—		11		B					
	/		0-1	C		A				1
	,		0-3-8	C		A	8		2	1
	%	(	0-4-8			A	8	4		
(1)	~		0-5-8	C		A	8	4		1
	\		0-6-8	C		A	8	4	2	
	⊞		0-7-8			A	8	4	2	1
(2)	ъ		2-8			A				
	#	=	3-8				8		2	1
	@	'	4-8	C			8	4		
(1)	:		5-8				8	4		1
	>		6-8				8	4	2	
	√		7-8	C			8	4	2	1
(3)	?		12-0	C	B	A	8		2	
	A		12-1		B	A				1
	B		12-2		B	A			2	
	C		12-3	C	B	A			2	1
	D		12-4		B	A		4		
	E		12-5	C	B	A		4		1
	F		12-6	C	B	A		4	2	
	G		12-7		B	A		4	2	1
	H		12-8		B	A	8			
	I		12-9	C	B	A	8			1
(4)	!		11-0		B		8		2	
	J		11-1	C	B					1
	K		11-2	C	B				2	
	L		11-3		B				2	1
	M		11-4	C	B			4		
	N		11-5		B			4		1
	O		11-6		B			4	2	
	P		11-7	C	B			4	2	1
	Q		11-8	C	B		8			
	R		11-9		B		8			1
	ǂ		0-2-8			A	8		2	
	S		0-2	C		A			2	
	T		0-3			A			2	1
	U		0-4	C		A		4		
	V		0-5			A		4		1
	W		0-6			A		4	2	
	X		0-7	C		A		4	2	1
	Y		0-8	C		A	8			
	Z		0-9			A	8			1
	Ø		0	C			8		2	
	1		1							1
	2		2						2	
	3		3	C					2	1
	4		4					4		
	5		5	C				4		1
	6		6	C				4	2	
	7		7					4	2	1
	8		8				8			
	9		9	C			8			1

(1) Print Blank
(2) Print ǂ
(3) Print &
(4) Print –

} On IBM 1403 Printer having typical printing chain installed

Figure 2. Standard BCD Interchange Code

sequence, with a blank having the lowest collating number of 00 and a nine having the highest of 63. The machine can read, punch, or type out on the IBM 1415 Console-I/O Printer, any of the characters symbolizing these 64 code points. Forty-eight of the characters will print on an IBM 1403 Printer having an alphameric chain.

Five of the BCD code points print out as two different characters, depending on the console-I/O printer type head in use and the 1403 printing chain installed. The choice of characters depends on the type of data being processed; for example, 1403 print arrangement A2 is available for report writing and most commercial uses, and print arrangement H2 is for program languages such as Cobol and Fortran and meets general scientific requirements for a more mathematical symbolism. Several other 1403 print arrangements are available; this manual assumes that print arrangement A2 (com-

merce) is being used. The console-I/O printer type head in use corresponds with the print arrangement installed in the 1403 Printer.

The standard BCD interchange code shown in Figure 2 replaces a previous code; Figure 3 shows the characters affected in the change to the new standard. Systems having the previous code may be converted to the standard BCD interchange code by requesting a no-charge Miscellaneous Engineering Specification (MES). The conversion also provides the 1415 Console-I/O Printer with 12 key-tops and the two type heads corresponding to two common 1403 print arrangements, A2 and H2. Figure 4 indicates the preferred names of special characters that previously had various or uncertain nomenclature.

Word Marks

A special eighth plane bit, called a word mark, is used to define the length of each instruction and data field in core storage. This word mark makes it possible to employ the variable-word-length concept in the 1410. It can be set and cleared, when necessary, by stored-program instructions. If word-mark identification is needed in magnetic tape records or cards, special instructions translate the word marks to word-separator characters during write operations, and translate the word-separator characters to word marks when the data are read back into storage from the tape unit or cards. A special instruction makes it possible to print the digit 1 in the print position, on the 1403 Printer, that corresponds to a word-mark position in the print area. When data are transferred to the console-I/O printer in load mode, each character that contains a word mark in storage has the word mark printed above that character. The letter A with an associated word mark is printed Ǎ.

In this manual, each character that has an associated word mark has the inverted circumflex (v) above it.

The word mark serves several functions:

1. It indicates the first character of an instruction.
2. It defines the size of a data word, or
3. It signals the end of an instruction.

OLD	NEW (COMMERCE)	CARD CODE	BCD CODE						
(	[	12-5-8		B	A	8	4		1
)	]	11-5-8	C	B		8	4		1
=	⌐	0-5-8	C		A	8	4		1
'	\	0-6-8	C		A	8	4	2	
"	≠	0-7-8			A	8	4	2	1
¢	♭	2-8			A				

Figure 3. Differences Between Previous Code and Standard BCD Interchange Code

The rules governing the use of word marks are:

1. Word marks are assigned specific storage locations in the program planning stage. They are set by a program load routine or by specific instructions within the program itself.
2. Word marks remain in their original locations unless cleared or set by a positive action – such as a clear storage, clear word mark, or set word mark instruction – or by a data move operation that specifies that they should be altered.
3. A word mark must be associated with the first character of each instruction (Op code position) and are usually associated with the high-order character in a data field.
4. Every instruction in the program must be followed by a word mark. This word mark stops the reading of the instruction.

Addressing

The instructions and data used by the 1410 are kept in core storage. Each core-storage position is addressable and has its own five-character address. Valid addresses range from 00000 to as high as 79999 for an 80K core storage. During execution of an instruction,

SYMBOL	NAME
ǂ	Group Mark
ǂ	Record Mark
ǂǂ	Segment Mark
⌐	Word Separator
@	At Sign
#	Number Sign
&	Ampersand
+	Plus
*	Asterisk
%	Per cent
/	Slash
\	Backslash
◻	Lozenge
b	Blank
♭	Substitute Blank
(	Left Parenthesis
)	Right Parenthesis
[	Left Bracket
]	Right Bracket
√	Tape Mark
<	Less than
>	Greater than
=	Equal to
;	Semicolon
:	Colon
.	Period or Point
'	Prime or Apostrophe
–	Minus or Hyphen (Dash)
Δ	Delta

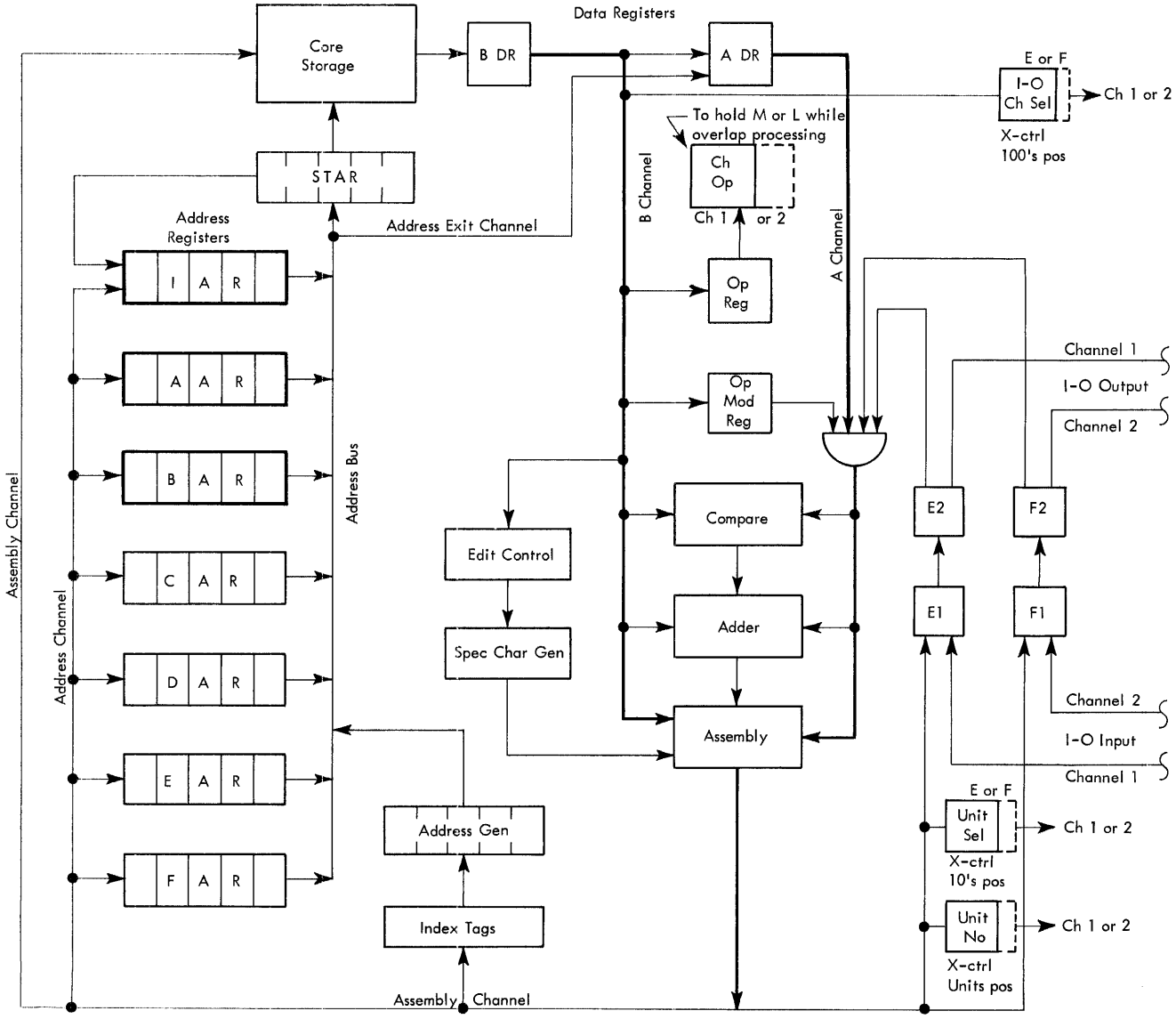
● Figure 4. Symbol Names

no address must be decremented past 00000 or incremented past the highest valid address. This includes operations that act on data in a single position only; such operations will be executed but, after the operation, the associated address register will contain an address beyond one of the limits and the system will stop and signal an error.

An address with zone bits in the units, thousands, or the ten-thousands position is invalid. Some characters are not accepted as valid addresses. These characters are: blank, 8-3, 8-4, 8-5, 8-6, 8-7. If an invalid address is given in an instruction, the system stops and signals an address check.

Address Registers

To read out an address from storage, five storage readout cycles are needed. In addition, a device is needed to accept the address characters and keep them until the whole address has been read out. The devices used to do this are the address registers. See Figure 5.



● Figure 5. Addressing and Data Flow, Simplified

Before being stored in the address registers, all characters are translated to two-out-of-five code. Before translation, the BCD characters are checked; an invalid bit combination causes an assembly channel error (Figure 50). The characters are also checked after translation; more or less than two bits in a character then causes an address channel error. Whenever the content of an address register is placed in core storage, as in a v G ccccc B (store the B-address register), it is first retranslated to BCD.

I-Address Register: The instruction address register (IAR) is a five-character register which is in effect an instruction counter. Its function is to keep track, character by character, of the current address of the stored program. For example, when an instruction is being read out, IAR initially holds an address that specifies the Op-code position of the instruction. The address is increased by one each storage cycle as the instruction is being read out. At the end of instruction read-out, IAR holds the Op-code address of the next sequential instruction. The setting of IAR is changed by one of four methods:

1. An address may be loaded into IAR from the console (for instance, the starting address of the program).
2. During normal sequential operations, IAR advances during each instruction read-out cycle to the address of the next character of the program.
3. After a successful branch instruction and during the execution of the branched-to instruction, the branched-to address — plus one — automatically replaces the current program address in IAR (see example under "Chaining Instructions").
4. A program reset or computer reset automatically loads 00001 into the IAR.

A- and B- Address Registers: The A- and B- address registers (AAR and BAR) are five-character registers; they accept the five-character addresses that specify the core-storage locations of the units positions of any data fields called for in the instruction (see "note" for exceptions). After each position of the data field is acted on, the address is decreased by one (if data were addressed by the low-order position) or increased by one (if data were addressed by the high-order position). The BAR contains the address of the input or output field during non-overlapped I/O operations. The AAR and BAR also contain the addresses needed to accomplish a branch to a subroutine while keeping track of the main routine (see "Chaining Instructions").

NOTE: In most cases, data fields are addressed by specifying the location of the low-order character. This is done so that addition, subtraction, and other operations start with the units position and end with the high-order position. However, data are addressed by specifying the location of the high-order character whenever: (1) Data movements originate or terminate outside the core-storage area; (2) Record moves originate and terminate within the core-storage area.

C-Address Register: The C-address register is a five-character register used in store address register, multiply, divide, table lookup, and other operations; it is not accessible to the programmer. In Store Address Register instructions (for instance a v G ccccc B equaling v G 12345 B), the ccccc signifies only that the C-address register (CAR), and not AAR, is used to store the contents of the BAR at the storage address 12345. The main significance is that the AAR is not changed.

D-Address Register: The D-address register is a five-character register used in multiply, divide, recomplement, and other operations; it is not accessible to the programmer.

E- and F-Address Registers: The E- and F-address registers are described under "Processing Overlap."

Storage Address Register: Each address register addresses core storage indirectly through the storage address register (STAR), a five-character register. STAR is not accessible to the programmer.

Single-Character Registers

The A- and B-data registers, Op and Op-modifier registers, I/O channel select register, unit select register, and unit number register are single-character registers used to store data during the execution of various instructions. Registers E1, E2, F1, F2 are described under "Processing Overlap."

B-Data Register: The B-data register accepts each character as it leaves the 1410 core-storage area. The character is stored in an eight-bit form (BCD code, check bit, and a word-mark bit). This register is reset and filled with a character from core storage during every storage read-out operation. The character can be entered back into storage from the output of the B-data register.

A-Data Register: The A-data register is reset and filled with the eight-bit character output from the B-data register whenever the operation requires it.

Op-Register: The Op (operation) register is reset and filled with a seven-bit character (the WM bit is dropped) output from the B-data register whenever the character is an operation code. The Op register stores the Op code of the instruction in process for the duration of the operation. Processing may be overlapped with certain I/O operations whose instructions have Op codes v M or v L. During overlap time, the M or L is held by a group of latches, freeing the Op-register

for handling the successive Op codes of the processing instructions.

Op-Modifier Register: The Op-modifier register is reset and filled with a seven-bit character output from the B-data register whenever the character is a d-character of an instruction. The Op-modifier register stores the modifier of the instruction in process for the duration of the operation. Processing may be overlapped with certain I/O operations. During overlap time, the A channel keeps track of the read or write condition signified by the d-character of the instruction, freeing the Op-modifier register for handling the successive d-characters of the processing instructions.

I/O Channel Select Register: The I/O channel select register accepts the hundreds position of an x-control field. This position specifies the data channel to be used and whether the operation will be performed in the overlap or non-overlap mode. (See "System Features and Considerations" for more detail on processing overlap.)

Unit Select Register (Channels 1 and 2): A unit select register is associated with each data channel. The register accepts the tens position of the x-control field. The tens position specifies the input or output unit being used for the operation.

Unit Number Register (Channels 1 and 2): A unit number register is associated with each data channel. The register accepts the units position of the x-control field. The units position specifies the I/O unit number or the operation shown in Figure 107.

Addressing Example (Figure 6)

Instruction address 02000 contains the operation code for the following instruction, which adds the field specified by the A-address to the field specified by the B-address.

Op Code	A-address	B-address
A	05389	05399

Both fields are addressed by their low-order positions. Addition progresses character by character from right to left, up to and including the first B-field position containing a word mark. For example, an A-field of 4123 added to a B-field of 002 results in a B-field sum of 125.

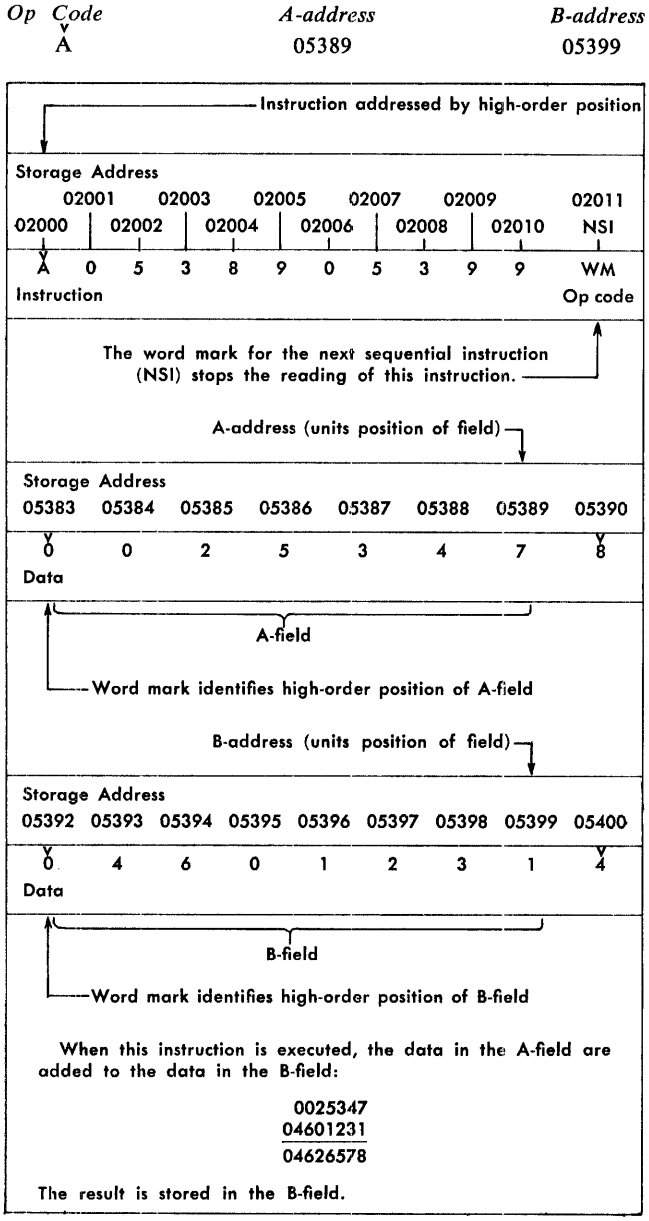


Figure 6. Addressing Example

Instructions

The 1410 uses stored-program instructions to cause all of the input and output devices to operate and to perform all arithmetic, branch, general data, and miscellaneous operations.

The operation performed is indicated by the form and contents of the instruction.

The programmer can write instructions in machine language, or he can use the IBM 1410 Autocoder system. If the programmer writes in machine language, he must write the actual instructions and storage addresses of all instructions and data used in the program. If he chooses the Autocoder system, he uses a set of mnemonic operation codes and assigns labels and symbols for data fields, record areas, and instructions. The Autocoder assembly program translates these symbolic entries into machine-language instructions.

Once the machine-language program has been produced, it can be kept permanently in punched cards, in the disk-storage unit, or on magnetic tape.

Instruction Form

The basic instruction form for the IBM 1410 Data Processing System is divided into four parts – the operation code, the A- or I- address (or x-control field), the B-address, and a d-character that modifies the operation code. Because of the variable length instruction form, the total length of an instruction can vary from one to 12 positions. An address notation of (A), (I), or (B) is used wherever ease of personal reading or vocalizing an instruction is more important than indicating the number of storage positions occupied.

Op Code	A- or I-address or X-control Field	B-address	d-character
x	xxxxx or xxx	bbbbb	x

Op Code is always a single character that specifies the basic machine operation to be performed.

A-Address is always five characters and specifies a core-storage location. An A-address may be symbolized by aaaaa or (A).

I-Address is always five characters and specifies the address of an instruction in storage. An I-address may be symbolized by iiiii or (I).

X-Control Field: If the field between the Op code and the B-address is made up of three characters, it is called the x-control field. The x-control field is used

only for I/O operations. In its hundreds or leftmost position, the x-control field specifies: (1) Which of two possible channels are used, and (2) Whether operation is in the overlap or non-overlap mode. Four characters, %, @, ¤, or *, cover the four possibilities. (See "Processing Overlap.") In its tens or center position, the x-control field specifies the kind of I/O device addressed by a character – such as 1 for reader, 2 for printer, 4 for punch, U for tape unit, or T for console-I/O printer. The low-order or rightmost position further defines the operation. For example, M %48 bbbbb W causes operation on *channel 1 and in the non-overlap* mode (%), and the field called for by the B-address is written out (W) by the *punch* (4). The cards are stacked as specified in *pocket 8.*

B-Address is always five characters and specifies a core-storage location. A B-address may be symbolized by bbbbb or (B).

d-Character is a single alphabetic, numeric, or special character used to specify an operation within the control of the operation code of the instruction.

Instruction Sequence

The IBM 1410 has a sequential method of program execution. Thus, instruction 1 is followed by instruction 2 and so on, unless special circumstances during processing make it necessary to alter this sequence. Branch instructions make it possible to test for special conditions and change the sequence of program execution, repeat an instruction or group of instructions, or transfer to special subroutines. These tests can be made at any time during processing and control the logic of the program.

Instruction Word Marks

Each instruction must have a word mark set over the operation code, and must not contain word marks in any other position. Also, a word mark must be set in the core-storage location immediately to the right of the last character of an instruction. This is normally the word mark associated with the operation code of the next sequential instruction.

Instruction Length Validity

Instruction length checking is incorporated in the system to ensure that each instruction read contains a valid number of characters for the operation code specified.

Valid instruction words vary in length from one to twelve characters depending on the amount of information required for the operation. The general instruction form consists of a single-character operation code followed by one or two 5-character addresses, or a 3-character input-output operation specification (x-control field) and, in some cases, a single-character operation modifier. Valid instruction word lengths are:

O	=	1 position
O d	=	2 positions
O xxx d	=	5 positions
O aaaaa	=	6 positions
O aaaaa d	=	7 positions
O xxx bbbbb d	=	10 positions
O aaaaa bbbbb	=	11 positions
O aaaaa bbbbb d	=	12 positions

The O specifies an operation code. The five a's specify the five-character address of the A-field and five b's specify the five-character address of the B-field. The three x's specify the x-control field, and the d specifies an operation modifier.

Instruction Descriptions

In this manual, instructions are described by using a standard form. Instructions that perform similar functions are incorporated in one general description. In these cases, individual instructions and details are shown in charts.

Descriptions follow this form:

Instruction Form: This contains the mnemonic operation code, the actual operation code, the A-address or I-address or x-control field, the B-address and the d-modification character.

Function: This is the description of the operation performed.

Word Marks: This is the information the programmer must have to determine the effect of word marks on the operation. All instructions must have a word mark associated with the operation code.

Timing: This is the formula used in calculating the time required for executing the instruction (in microseconds). See Figure 7 for symbols used in the formulas. Because there are several valid lengths for some instructions, these lengths, as they apply to the timing formula, are included in the section. Some operations also have extra steps, depending on the data, etc. When this condition is present, the numbers assigned to these steps (D, E, R, etc.) are also given. (All timing formulas change when the 1410 Accelerator feature is installed; the faster timings resulting from the feature are listed together in this manual under "1410 Accelerator.")

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

Address Registers after Operation: The contents of the address registers are represented by the codes described under "Chaining Instructions."

A — is the A-field length.

B — is the B-field length.

C — is 1 if the branch is taken. It is zero otherwise.

D — is the number of characters in the B-field from the start of zero suppression to the place where the dollar sign is inserted. If no dollar sign is inserted, D is zero.

E — is 2 on a single-character multiply or divide operation. It is 1 on a single-character add, subtract, reset add, reset subtract, table search, or a 6-character multiply or divide operation. It is zero otherwise.

I/O is the time used by input/output device to accept or send data and the Synchronizer access time, when applicable.

L — is the instruction length.

M — is the multiplier length.

N — is the number of fields actually compared on a table search. The B-field length on a table search includes only those argument fields actually compared and the intervening function values.

Q — is the quotient length.

R — is 1 if a recomplement is taken on an add or subtract operation. It is zero otherwise.

Z — is the number of characters in the B-field from the start of zero suppression (as indicated in the control field) to the left end of the B-field.

Figure 7. Timing Formula Symbols

Chaining Instructions

If the A- and B-address registers contain addresses of the next fields to be processed during the execution of a program, another complete instruction is not necessary. The operation code, alone, can be given and the contents of the address registers are used to specify the A- and B-fields. Connecting instructions together in this manner is called chaining. This method conserves storage space and saves time, because the address registers do not have to be reloaded during the reading of the instruction. For example, an ADD instruction, A 05985 06985, is executed. Field A has five characters and field B has six characters. The address registers after the operation contain:

A-address Reg	(decreased five positions)	B-address Reg	(decreased six positions)
05980		06979	

The A- and B-addresses of the fields to be used in the next operation are 05980 and 06979 respectively. The next instruction need contain only the operation code, because the A- and B-address registers are already at the desired locations. The instruction S causes the data at location 05980 to be subtracted from the data at 06979. This chaining technique can be used to link several instructions together. The only restriction is that the fields remain in sequence and that the address registers contain valid addresses. However, if a two-address instruction does not require a d-character (11-position instruction), the op-modifier register is blanked; thus, any chained instructions then directly following will be automatically assigned a blank d-character.

The descriptions of the instructions include the contents of the address registers after the operation has been performed. The programmer can use this information to determine which instructions can be chained in particular situations. Figure 8 shows the symbols that represent the contents of the address registers.

Branch instructions have the following effects:

If Branch Occurs:

1. After the instruction has been read and IAR has accordingly stepped, the address in the IAR transfers to the BAR. This NSIB (no longer the next instruction to be read out) can be stored by a G (C) B after the branch operation.

2. The next instruction read is at the address in the AAR. (During the reading of the branched-to instruction, the op-code address is taken from AAR and the address in AAR is placed in STAR; this address is then increased by one and placed in IAR for read-out of the subsequent positions in the branched-to instruction and all the positions of instructions following it.)

If No Branch Occurs (Two-Address Branch Instructions): After the operation, AAR contains the BI (branch-to) address, the IAR contains the NSI, and BAR contains the B-address minus one. This permits the next position to be given the same test, without specifying a B-address, until either the BAR has decremented to a position meeting the test and causing the branch, or until there are no more such conditional branch instructions in the series.

If No Branch Occurs (One-Address Branch Instructions): After the operation, both AAR and BAR contain the BI (branch-to) address. The IAR contains the NSI.

Example: Chained Two-Address Branch Instruction

			IAR	AAR	BAR
No Branch (no 3 in 580)	B 00600 00580 3	before	01000	02000	03000
		after	01012	00600	00579
No Branch (no 3 in 579)	B	before	01012	00600	00579
		after	01013	00600	00578
Branch (3 in 578)	B	before	01013	00600	00578
		after	01014	00600	01014
Optional Branch-To Instruction	G 04000 B	before	01014	00600	01014
		after	00607	00600	01014

CAR is used to store content of BAR in 4000

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 is taken
NSIB	Address of the next instruction in storage following the branch (not executed in normal sequence because branch was taken)
LA	The number of characters in the A-field
LB	The number of characters in the B-field
LW	The number of characters in the A- or B-field, whichever is shorter
Ap	The previous contents of the A-address register
Bp	The previous contents of the B-address register

● Figure 8. Address-Register Symbols

The add, subtract, zero and add, zero and subtract, multiply, and divide operation codes are used to perform the system's arithmetic operations. The use of add-to-storage logic in the IBM 1410 Data Processing System eliminates the need for special-purpose accumulators or counters in the system. Because any group of storage positions can be used as an accumulating field, the capacity for arithmetic functions is not limited by a predetermined number of counter positions.

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.

Figure 9 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 both B and A bits (12 zone) or no B or A bits (no zone), and to indicate a negative factor in only one way: with a B bit (11 zone).

The arithmetic operations in the IBM 1410 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 10). In an algebraic subtraction, recall that the sign of the subtrahend (A-field, in 1410) is changed and the subtrahend, with its changed sign, is then *added* to the minuend (B-field). The sign of the result is the "sign of the greater value" only *after* the A-field sign is considered to have been changed.

The BCD characters to be added are checked for valid BCD coding, and an invalid bit combination causes an A or B channel error (Figure 50). The characters are then translated to qui-binary code and fed to the adder. The adder's output is retranslated to BCD; an invalid combination of bits at this point causes an assembly channel error.

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

Figure 9. Bit Equivalents for Signs

Digit Coding

In all arithmetic operations, the presence of characters represented by the card codes of blank, 8-3, 8-4, 8-5, 8-6, and 8-7 in the numeric portion of a field are treated as zero, 3, 4, 5, 6, and 7 respectively. For example, an A-field of #6AJ subtracted from a B-field of :A1B results in a B-field remainder of 8G2C. See Figure 2 and "Subtract (Two Fields)."

Overflow

If the result exceeds the limit of the B-field (determined by the B-field word mark), the carry is lost and the arithmetic overflow indicator turns on. One of the test and branch instructions, J(I)Z, tests and turns off this indicator.

Zero Balance

If the result of any add, subtract, multiply, zero and add, or zero and subtract operation is a zero balance, the zero balance indicator is turned on. This indicator

TYPE OF OPER.	A-FLD. SIGN	B-FLD. SIGN	TYPE OF ADD CYCLE	SIGN OF RESULT
ADD	+	+	True-Add	+
		−	Compl-Add	Sign of Greater Value
	−	+	Compl-Add	
		−	True-Add	−
SUBTRACT	+	−	True-Add	−
		+	Compl-Add	Sign of Greater Value (after A-field sign is changed as a result of the subtract instruction)
	−	−	Compl-Add	
		+	True-Add	+

Figure 10. Types of Add Cycles and Sign of Result for Add and Subtract Operations

can be tested by one of the test and branch instructions—J(I)V. The indicator is turned off by the next add, subtract, multiply, zero and add, or zero and subtract instruction that does not result in a zero balance.

Arithmetic Operation Codes

Add (Two Fields)

Instruction Form:

Mnemonic	Op Code	A-address	B-address
A	Ä	aaaaa	bbbbb

Function: This instruction causes the numeric data in the A-field to be added algebraically to the numeric data in the B-field. The result is stored in the B-field. Zone bits remain undisturbed in the B-field (except for the sign position, which may be required to change. For example, an A-field of 12C added to a B-field of 00K results in a B-field sum of 12A, changing the B-field sign from − to +). A-field zone bits (except for the sign position) are ignored. The resultant B-field is in true form at the end of the operation.

Word Marks: The B-field word mark stops the operation and must be set to define the high-order position. If the A-field is shorter than the B-field, it must also have a defining word mark to stop transmission of data from the A-field to the B-field. In this case, the system automatically adds zeros to the extra high-order positions of the B-field until it detects the B-field word mark. 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.

Timing: T = 4.5 (L + 1 + E + A + 1.5B + 1.5RB).
When L = 1, E = 1; when L = 11, E = 0.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LW	B-LB

Add (One Field)

Instruction Form:

Mnemonic	Operation Code	A-address
A	Ä	aaaaa

Function: This form of the add instruction is used to double the A-field. The A-field is added to itself and the result is stored in the A-field. This operation is always performed with a true-add cycle and the sign-bit configuration of the result is always the same as the original sign of the A-field.

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

Timing: T = 4.5 (L + 1 + A + 1.5A).
L = 1 or 6.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	A-LA

Subtract (Two Fields)

Instruction Form:

Mnemonic	Op Code	A-address	B-address
S	Š	aaaaa	bbbbb

Function: The numeric data in the A-field are subtracted algebraically from the data in the B-field. The result is stored in the B-field. Zone bits remain undisturbed in the B-field except for the sign position, which may be changed. A-field zone bits are ignored in all positions except the sign position. The B-field result is in true form.

Word Marks: The B-field word mark stops the operation and must be set over the high-order position of that field. If the A-field is shorter than the B-field, it, too, must have a defining word mark to stop transmission of data from A to B. When the A-field is shorter than the B-field, the machine subtracts zeros from the extra high-order positions of the B-field up to and including the word-mark position. 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.

Timing: T = 4.5 (L + 1 + E + A + 1.5B + 1.5RB).
When L = 1, E = 1; when L = 11, E = 0.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LW	B-LB

Subtract (One Field)

Instruction Form:

Mnemonic	Op Code	A-address
S	Š	aaaaa

Function: The A-field is subtracted from itself, and the result is stored in the A-field. The numeric portion of the A-field is always zero after the operation, but zones in the A-field are unchanged and the A-field sign bit configuration is the same as it was before the operation. For example, an A-field of ÄBQ, subtracted from itself, leaves ??! as the remainder.

Word Marks: A word mark must be associated with the high-order position of the A-field.

Timing: T = 4.5 (L + 1 + A + 1.5A).
L = 1 or 6.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	A-LA

Zero and Add (Two Fields)

Instruction Form:

Mnemonic	Op Code	A-address	B-address
ZA	P	aaaaa	bbbbb

Function: This instruction causes the numeric data in the A-field to be stored in the B-field. The sign of the result field (B-field) is the same as the sign of the

A-field. If the A-field has no sign and is thus understood to be positive, the system generates an actual positive sign for the B-field by placing B and A bits over the units position. All other zone positions in the B-field are set to no zone (no B nor A bits).

Word Marks: The B-field must have a defining word mark to stop the operation. The A-field requires a word mark only if it is shorter than the B-field. If the A-field is shorter than the B-field, extra high-order B-field positions are set to zero. 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.

Timing: T = 4.5 (L + 1 + E + A + 1.5B).
When L = 1, E = 1; when L = 11, E = 0.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LW	B-LB

Zero and Add (One Field)

Instruction Form:

Mnemonic	Op Code	A-address
ZA	?	aaaaa

Function: This form of the zero and add instruction causes no change to the numeric data of the A-field. This instruction is used to strip the A-field of all zones, except in the units (sign) position, and to change non-numeric codes (blanks and 8-3, 8-4, 8-5, 8-6, and 8-7 codes) to their numeric equivalents (zero, 3, 4, 5, 6, and 7 respectively). The sign of the A-field is retained, but the bit configuration of the plus sign may change. If the A-field plus sign bit configuration is not a B and A bit, it is changed to the B and A bit configuration.

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

Timing: T = 4.5 (L + 1 + A + 1.5A).
L = 1 or 6.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	A-LA

Zero and Subtract (Two Fields)

Instruction Form:

Mnemonic	Op Code	A-address	B-address
ZS	!	aaaaa	bbbbb

Function: This operation causes the numeric data in the A-field to be moved and stored in the B-field with the opposite sign as shown in Figure 11. All other zone positions in the B-field are set to no zone (no B and no A bits). If the A-field is shorter than the B-field, extra high-order B-field positions are set to zero.

Word Marks: The B-field must have a defining word mark to stop the operation. The A-field requires a word mark only if it is shorter than the B-field. In this instance, the system inserts zeros in the extra high-order positions of the B-field up to, and including, the word-mark position.

If the A-field is longer than the B-field, high-order positions of the A-field that exceed the limits imposed by the B-field word mark are not processed.

Timing: T = 4.5 (L + 1 + E + A + 1.5B).
When L = 1, E = 1; when L = 11, E = 0.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LW	B-LB

Zero and Subtract (One Field)

Instruction Form:

Mnemonic	Op Code	A-address
ZS	!	aaaaa

Function: This form of the zero and subtract instruction causes no change to the numeric data in the A-field. This instruction is used to strip the A-field of all zones, except in the units (sign) position, and to change the A-field sign. If the A-field was positive before the operation, it is negative (B bit) after the operation. If the A-field was negative before the operation, it is positive (B and A bits) after the operation. Non-numeric codes (blanks and 8-3, 8-4, 8-5, 8-6, and 8-7 codes) are replaced by their corresponding numeric equivalents (zero, 3, 4, 5, 6, and 7).

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

Timing: T = 4.5 (L + 1 + A + 1.5A).
L = 1 or 6.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	A-LA

Multiply

Instruction Form:

Mnemonic	Op Code	A-address	B-address
M	@	aaaaa	bbbbb

Function: The multiply instruction causes the numeric data in the A-field (multiplicand) to be repetitively added and stored in the B-field (product), starting with the low-order positions. The multiplier is

A-FIELD SIGN	B-FIELD SIGN AT END OF OPERATION
No B and No A bits (plus)	B bit (minus)
B bit (minus)	B and A bits (plus)
B and A bits (plus)	B bit (minus)
A bit (plus)	B bit (minus)

Figure 11. Sign Changes for Zero and Subtract (Two Fields)

initially located in the B-field, but in the high-order positions.

B-field Length: Because the product is developed in the B-field, the field must be big enough to accommodate the repetitive additions of the A-field and still not interfere with the multiplier position. Therefore, the length of the B-field is determined by adding 1 to the sum of the number of digits in the multiplicand and the multiplier.

Concept:

Multiplier = $\overset{v}{X}XX^{+}$ A-address

Multiplicand = $\overset{v}{Y}YYY\overline{Y}$

Product Field Before = ZZZZZZZZZ B-address

Product Field After = $\overset{v}{0}$PPPPPPPP$\overline{P}$

Because the multiplier has three positions and the multiplicand has five positions, the product field (B-field) has nine positions. Before the multiply instruction is given, an image of the multiplier must be moved or otherwise placed in the high-order positions of the product field, which then reads $\overset{v}{X}XX^{+}$ZZZZZZ. During the multiply, the product field (B-addressed by its units position) is multiplied by the multiplicand, which is A-addressed by its units position. The first scan automatically replaces any data in the positions to the right of the multiplier with zeros, giving $\overset{v}{X}XX^{+}$000000, and the first add changes that to $\overset{v}{X}$XXx0YYYYY as it removes the sign, decrements the multiplier image by 1, and adds the multiplicand to the low-order positions of the product field. Repetitive addition of the multiplicand continues as long as any part of the multiplier image remains, with the final product algebraically signed and the multiplier image destroyed, giving a product field of $\overset{v}{0}$PPPPPPPP$\overline{P}$.

Zone Bits and Sign: Zone bits that appear in any position of the multiplicand (A-field) are undisturbed by the multiply operation. Zone bits that appear in the assigned product area are eliminated before product development starts. Zone bits that appear in the multiplier (high-order position of the B-field) are eliminated during product development. Before zone-bit elimination starts, both the units position of the multiplicand and the multiplier are checked for zone bits (the sign of the factor). The presence or absence of zone bits is used to determine the sign of the product. Like signs in the units position of the multiplicand and multiplier result in a plus sign. Unlike signs result in a minus sign. At the end of the operation, the sign of the product is placed in the units (sign) position of the B-field.

Multiplier Factor: As the product is developed, the multiplier is eliminated, digit by digit. If it is required for later use in the program, it must be retained in another storage area.

Word Marks: Word marks must be set to identify the high-order position of the multiplicand, and also of the multiplier as represented in the product field.

Example: Multiplication in the example shown (Figure 12) is a series of repetitive true or complement additions. The number and type of additions are determined by the multiplier digit. A multiplier digit from one to four causes the A-field (multiplicand) to be true-added that number of times. The multiplier digit is reduced by one each true-add cycle until zero is reached. The recognition of the zero ends the true-add cycles, and shifts the portion of the product field, being developed, one position to the left. A multiplier digit of five or more causes the A-field to be repetitively complement-added in the low-order positions of the product field. The number of complement-additions depends on the multiplier digit. Thus, if the multiplier digit is 8, two complement-additions occur (the tens complement of 8 is 2). After the required number of complement-additions have been made, the product field is left-shifted and the A-field is true-added to the product field, starting in the *tens* position of the product. This method of processing multiplier digits five

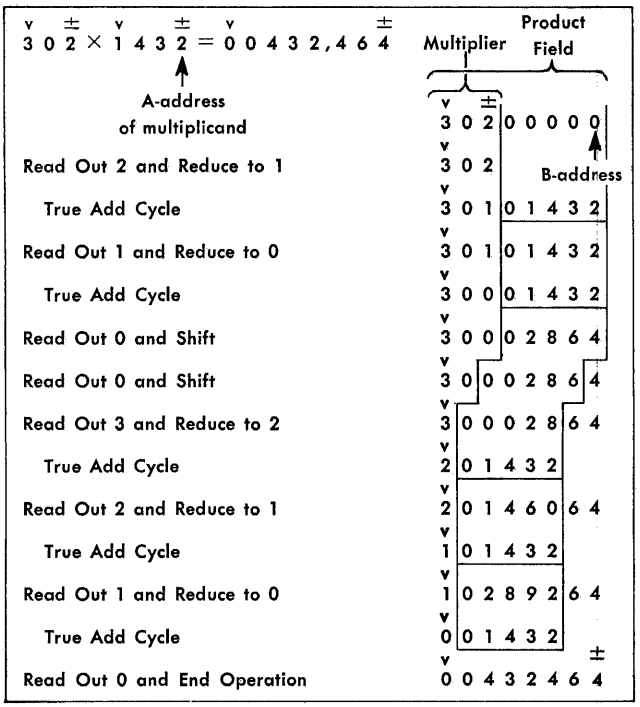


Figure 12. Multiply Example

through nine saves processing cycles and time. For example, a multiplier digit of 8 causes, instead of eight true-added cycles, only three total cycles.

```
Example                          Product Field
 3456*                           800000
 × 8                  CA         96544    one
27648                            996544    result of 1st CA
                      CA         96544    two
*Tens complement of              093088    result of 2nd CA
 3456 multiplicand    TA         3456     three
 = 96544                         027648    end result
```

Timing: T ≈ 4.5 [L + 1 + E + 2.5M + (2.5M + 1.5) (2.5A + 3)]

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	B-LB

Divide

Instruction Form:

Mnemonic	Op Code	A-address	B-address
D	%	aaaaa	bbbbb

Function: During a divide operation, the dividend located in the B-field is divided by the divisor located in the A-field. The result (quotient) is stored in the high-order positions of the B-field. The dividend is destroyed during the operation, except for any remainder, which remains in the low-order positions of the field. Conditions that must be considered before and during a divide operation are:

1. Addressing of Factors: The A-address specifies the units position of the divisor. The B-address specifies the high-order position of the dividend. The dividend itself is located in the low-order positions of the quotient-dividend (B) field.

2. B-field Length: Because the quotient is developed in the B-field, the field must be big enough to accommodate the repetitive complement-additions of the A-field and still not interfere with the quotient position being developed. Therefore, 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
          v
   12 ⌈ 147   3-digit dividend
      ↖----   2-digit divisor
        +1
          6 positions must be allowed
            in the B-field – 000DDD±
                             v
```

3. Sign and Zeros: The divisor (located in the A-field) can be signed or unsigned. If no bits are in the units position of the divisor, the system assumes that the divisor is positive. There must be a sign in the units position of the dividend (located in the B-field). The sign will stop the division. The sign must consist of B and A bits for plus, or a B bit for minus. The quotient field should contain zeros when the divide operation starts. Moving the dividend into the B-field by means of a zero and add instruction ensures both the presence of zeros in the high-order (quotient) positions of the B-field and the proper signing of the B-field.

Word Marks: A word mark must be set to define the high-order position of the divisor (A-field). If a zero and add instruction was used to move the dividend into the B-field, the high-order position of the field contains a word mark. This B-field word mark is not needed during the divide operation; it is ignored, but retained.

Example: Division in the example shown (Figure 13) is a series of complement-additions, and, in the case of an overdraw, a true-addition. Complement-add cycles take place until signalled by a no-carry condition. The divisor factor is true-added, and then the shift to the next position takes place. The carry resulting from a successful complement-add cycle is used to develop the quotient. This type of operation continues until the true-add cycle that corrects the units position of the dividend occurs. During this cycle, the signs of the dividend and the divisor are analyzed. The resultant sign is then applied to the units position of

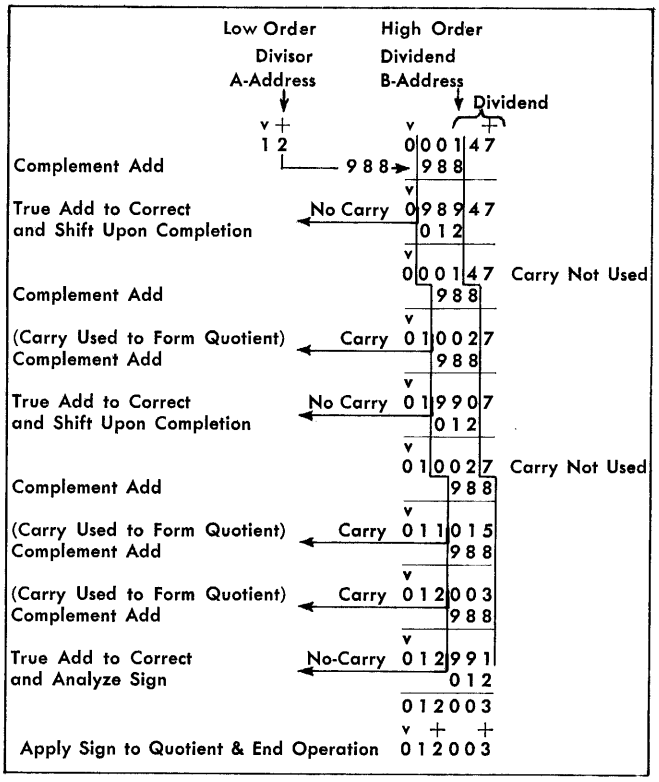


Figure 13. Division Example

the quotient and the operation is ended. At the completion of the operation:

1. 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.

2. The remainder is located in the low-order positions of the B-field.

3. The sign of the quotient follows algebraic sign rules (Figure 14), and appears over the units position of the quotient at the end of the operation.

4. The sign of the remainder is the sign of the original dividend (Figure 14).

Divisor Sign	+	+	−	−
Dividend Sign	+	−	+	−
Remainder Sign	+	−	+	−
Quotient Sign	+	−	−	+

Figure 14. Divide Sign Control

Notes:

1. 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. An improperly addressed dividend can cause a divide overflow condition if the result of the first divide operation is greater than 9.

2. If the quotient field is not large enough, a resulting overflow may or may not be indicated. If the field is one position too small, there will not be an overflow indication, even though the units position of the adjacent field is changed. If the field is two or more positions short, the divide operation usually results in a divide overflow. Too small a quotient field is a programming error, and is not checked by the system. The divide overflow condition can be tested, however, by one of the test and branch instructions – $\overset{v}{J}$(I)W.

3. Division by zero always results in a divide overflow indication.

4. If a larger quotient is required, extra zeros can be added to the dividend before the divide operation starts. For each additional quotient digit desired, insert one zero to the right of the dividend as shown in Figure 15.

Timing: T $\approx$ 4.5 {L + 1 + E + 6.5Q [A + 1.5 (A + 2)]}.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	Tens position of the quotient field.

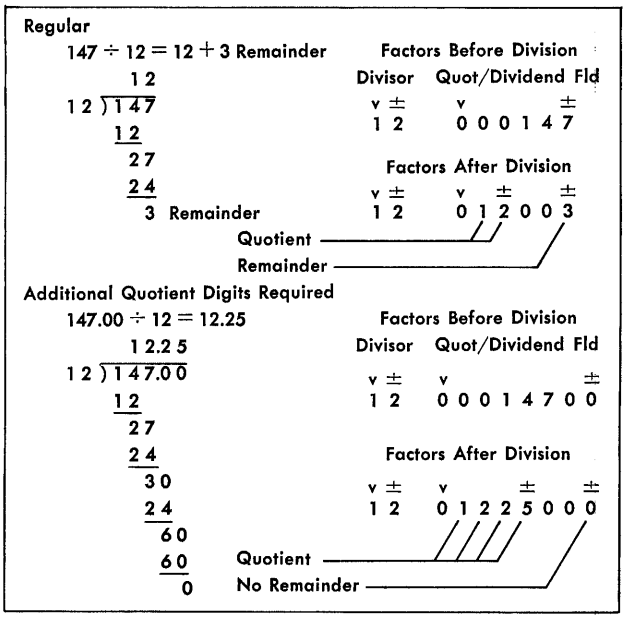


Figure 15. Additional Quotient Digits

Indexing

The IBM 1410 Data Processing System is equipped with fifteen index registers that can be used to modify the A-, I-, or B-address of most instructions. Each index register is assigned five storage locations (Figure 16).

Addresses that cannot be indexed:

1. X-control field of M, L, or U instruction.
2. G instruction (Store address register).

To modify addresses, the index register containing the index factor must be selected. The index factor (contents of the index register) is not a field address, but an actual number. To select the correct index register, the A-address, the B-address, or both addresses must be tagged. A tag is a zone bit over the hundreds position, the tens position, or both the hundreds and tens positions of the address to be modified (Figure 17). Both the A- and B-addresses may be tagged; if they are, the index register tagged in the A-address is not necessarily the same as the one tagged in the B-address (see Example 5).

The index factor of the index register selected is added algebraically to the address of the instruction after the address has entered the address register from storage and before the instruction is executed. The address is modified in the appropriate address register.

INDEX REGISTER	INDEX FACTOR STORAGE LOCATIONS
1	00025 to 00029
2	00030 to 00034
3	00035 to 00039
4	00040 to 00044
5	00045 to 00049
6	00050 to 00054
7	00055 to 00059
8	00060 to 00064
9	00065 to 00069
10	00070 to 00074
11	00075 to 00079
12	00080 to 00084
13	00085 to 00089
14	00090 to 00094
15	00095 to 00099

Figure 16. Index Register Locations in Storage

Thus, the actual instruction in storage itself is not changed but, because data are read from storage under control of the address registers, the effect of the original instruction is changed. The index factor also remains unchanged as a result of indexing; however, the index factor can be changed, whenever required, by other operations. For example, an index register containing 00123 can be reduced to 00000 by a subtract (one field).

If the sign of the index factor is plus, the factor is added to the tagged address; if negative, the factor is subtracted from the tagged address. The arithmetic overflow latch is not set as a result of overflows incurred during indexing. The result of this modification must be a valid storage address, or the system will stop on an error when the instruction is executed. The validity of addresses must be considered when altering or interchanging programs between 10K, 20K, 40K, 60K, and 80K systems.

Storage positions 00025 to 00099 can be used for general storage if they are not required for indexing purposes. Word marks can be set in this area at any time because they do not control the indexing operation, which ignores word marks. Zone bits are undisturbed in the index registers and have no effect on indexing except when they appear in the sign position of an index factor.

Examples:

1. Modify the A-address of this instruction:

Op Code	A-address	B-address
A	009Z6	00961

The A-address is tagged by an A bit over the tens position (Z = A81). Index register 1 is selected (Figure 17). Because the index register 1 factor is minus it is subtracted from the A-address:

A-address	= 009Z6	=	00996
Index register 1 factor	= 0001J	=	−00011
Effective A-address	=		00985

	Op Code	A-address	B-address
Effective instruction:	A	00985	00961

Valid addresses on 10K and larger systems.

2. Modify the B-address of this instruction:

Op Code	A-address	B-address
A	00459	01MT1

The B-address is tagged by a B bit over the hundreds position and an A bit over the tens position (M = B4

and T = A21). Index register 9 is selected. Because the index register 9 factor is plus, it is added to the B-address:

B-address	= 01MT1	=	01431
Index register 9 factor	= 0010C	=	+00103
Effective B-address	=		01534

	Op Code	A-address	B-address
	v		
Effective instruction:	A	00459	01534

Valid addresses on 10K and larger systems.

3. Modify the A- and B-addresses of this instruction:

Op Code	A-address	B-address
v		
S	00V51	00W50

Both the A- and B-addresses are tagged by an A bit over the hundreds position (V = A41 and W = A42). Index register 4 is selected. Because the index register 4 factor is unsigned, it is added to both addresses.

A-address	= 00V51	= 00551
Index register 4 factor	=	+00100
Effective A-address	=	00651
B-address	= 00W50	= 00650
Index register 4 factor	=	+00100
Effective B-address	=	00750

	Op Code	A-address	B-address
	v		
Effective instruction:	S	00651	00750

Valid addresses on 10K and larger systems.

4. Modify the A-address of this instruction:

Op Code	A-address	B-address
v		
A	126A8	06429

The A-address is tagged by B and A bits over the tens position (A = BA1). Index register 3 is selected (Figure 17). Because the index register 3 factor is plus, it is added to the A-address.

A-address	= 126A8	=	12618
Index register 3 factor	= 0643E	=	+06435
Effective A-address	=		19053

	Op Code	A-address	B-address
	v		
Effective instruction:	A	19053	06429

Valid addresses on 20K and larger systems.

5. Modify the A- and B-addresses of this instruction, by tags selecting two different index registers:

Op Code	A-address	B-address
v		
A	77TM8	23GY5

The A-address is tagged by an A bit over the hundreds position and a B bit over the tens position (T = A21 and M = B4). Index register 6 is selected (Figure 17). Because the factor in index register 6 is positive, it is added to the A-address.

The B-address is tagged by B and A bits over the hundreds position and an A bit over the tens position (G = BA421 and Y = A8). Index register 13 is selected (Figure 17). Because the factor in index register 13 is negative, it is subtracted from the B-address.

A-address	= 77TM8	=	77348
Index register 6 factor	= 0020B	=	+00202
Effective A-address	=		77550
B-address	= 23GY5	=	23785
Index register 13 factor	= 0200N	=	−02005
Effective B-address	=		21780

Op Code	A-address	B-address
v		
A	77550	21780

Valid addresses on 80K system.

Timing for indexing operations:

T = 34.5 for each single address indexed.

Note: For proper operation of indexing on a 10K system, the high-order position of the address being indexed must always contain a zero.

B-BIT OVER HUNDREDS POSITION	A-BIT OVER HUNDREDS POSITION	B-BIT OVER TENS POSITION	A-BIT OVER TENS POSITION	TAG INDEX REGISTER
				NONE
			A	1
		B		2
		B	A	3
	A			4
	A		A	5
	A	B		6
	A	B	A	7
B				8
B			A	9
B		B		10
B		B	A	11
B	A			12
B	A		A	13
B	A	B		14
B	A	B	A	15

Figure 17. Zone Bits Used to Tag Index Registers

The 1410 program can examine conditions that may arise during processing and transfer the program to a predetermined set of instructions or subroutines as a result of specific tests. This is called the logical ability of the IBM 1410 Data Processing System. A well designed program uses this logic by making tests at program points where processing conditions may call for a change in the normal sequence of program-step execution. For example, the existence of a 2 in column 1 of an input card may be used as a signal to combine fields A and B; or the existence of a 3 may call for combining fields A and C. A programmed test is made to determine which digit the card column contains. The next program step taken is then a condition of the kind of digit encountered. A transfer from one instruction to another instruction or set of instructions to alter the sequential execution of program steps is called a program branch. A branch instruction can be one of two types:

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

2. A branch that occurs as a result of a particular condition such as an arithmetic overflow, zero balance, etc., is called a *conditional branch.* If the condition is present at the time a conditional branch instruction is executed, sequential execution of program steps is bypassed. Then, the program branches to the address of the instruction specified by the I-address of the conditional branch instruction. If the condition is not present, the system simply continues with the next sequential instruction.

All branch instructions have a d-character that is used to specify the conditions necessary for a program transfer.

Branch Codes

Branch Unconditionally

Instruction Form:

Mnemonic	Op Code	I-address	d-character
B	J	iiiii	blank

Function: This is an unconditional branch instruction. Whenever it is executed, it causes a program branch to the location specified by the I-address.

Word Marks: No special considerations.

Timing: T = 4.5 (L + 2).

L = 1 or 7

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSIB	BI	NSIB

Test and Branch (Conditional)

Instruction Form:

Mnemonic	Op Code	I-address	d-character
xxx	J	iiiii	x

(See Figure 18)

Function: This is a conditional branch instruction. It allows the program to test the conditions that can arise during the processing. The d-character specifies the condition or internal indicator that is examined for an on or off condition. If the indicator is on, the program branches to the I-address for the next instruction. If the indicator is off, the program continues with the next sequential instruction. The indicators and the d-characters used to test them are shown in Figure 18.

The carriage channel 9 and carriage channel 12 indicators are turned on whenever the corresponding

DESCRIPTION	MNEMONIC	d-CHARACTER
Branch Unconditionally	B	Blank
Branch if Carriage 9 (Ch 1)	BC9 or BC91	9
Branch if Carriage 9 (Ch 2)	BC92	!
Branch if Carriage Overflow, 12 (Ch 1)	BCV or BCV1	@
Branch if Carriage Overflow, 12 (Ch 2)	BCV2	□
Branch if Compare Unequal	BU	/
Branch if Compare Equal (B = A)	BE	S
Branch if Compare Low (B < A)	BL	T
Branch if Compare High (B > A)	BH	U
Branch if Zero Balance	BZ	V
Branch if Divide Overflow	BDV	W
Branch if Arithmetic Overflow	BAV	Z
Branch if Inquiry Request (Ch 1)	BNQ or BNQ1	Q
Branch if Inquiry Request (Ch 2)	BNQ2	*
Branch if Overlap in Process (Ch 1)	BOL1	1
Branch if Overlap in Process (Ch 2)	BOL2	2
Branch if Carriage Busy (Ch 1)	BPCB or BPCB1	R
Branch if Carriage Busy (Ch 2)	BPCB2	L

● Figure 18. Test and Branch Instructions

holes in the carriage tape are sensed. They are turned off whenever another carriage tape channel is sensed. To avoid losing the result of a compare operation, it is necessary to test the compare indicators before they are turned off by the next compare, table lookup, or branch if character equal instruction and are set to the result of that next operation.

The overflow indicators are turned off by either a test and branch instruction or a computer reset operation. A computer reset operation also turns on the unequal- and low-compare indicators, and turns off the zero result indicator.

Word Marks: No special considerations.

Timing: T = 4.5 (L + 1 + C).

L = 1 or 7

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Branch	NSIB	BI	NSIB
No branch	NSI	BI	BI

Branch if I/O Channel Status Indicator On

Instruction Form:

Mnemonic	Op Code	I-address	d-character
xxx	R (Ch. 1)	iiiii	x
xxx	X (Ch. 2)	iiiii	x

(See Figure 19)

Function: This is a conditional branch instruction. Branching to the specified I-address occurs if I/O channel status indicators for channel 1 or channel 2, when tested, are on. These indicators are set on as a result of conditions that arise during an operation of one of the input-output units serviced by that channel. The bit configuration of the d-character used in the branch if I/O channel status indicator on instruction determines which test or tests are made. This instruction with a ≠ d-character must be given before the execution of another input or output instruction on the same channel to avoid interlocking the system. The system is inter-

DESCRIPTION	MNEMONIC	INDICATOR	d-CHARACTER	OPERATION
Branch if I/O unit Not Ready	BNR 1 or 2	Not Ready	1	The indicator is internally set during instructions involving input/output devices, if these devices or their associated buffers are in a not ready condition, but before any data transfer takes place. If the indicator is set ON, the operation is terminated and no data are transferred.
Branch if I/O unit Busy	BCB 1 or 2	Busy	2	The indicator is internally set during instructions involving input/output devices, if these devices or their associated buffers are in a busy condition, but before any data transfer takes place. If the indicator is set ON, the operation is terminated and no data are transferred.
Branch if I/O Unit Data Check	BER 1 or 2	Data Check	4	The indicator is set ON, after the transfer of data involving input/output devices, their associated buffers or the processing unit, if a parity error was detected during the data transfer.
Branch if I/O Unit Condition	BEF 1 or 2	Condition	8	The indicator is normally set during the move or load instruction, before any data transfer takes place. As an example, the indicator is set ON if an end of file (last card stacked) has occurred in the card reader. If the indicator is set ON, the operation is terminated and no data are transferred.
Branch if I/O Wrong Length Record	BWL 1 or 2	Wrong Length Record	—(B-bit)	The indicator is set ON, if the record written from storage or written in storage is not the correct length.
Branch if I/O Unit No Transfer	BNT 1 or 2	No Transfer	⅋ (A-bit)	No Transfer. The indicator is normally set before any data transfer takes place. If it is set ON, it indicates that no data was available to transfer.
Branch if Any I/O Channel Status Indicator On	BA 1 or 2	All	≠	The group mark has all the bits needed to test all of the above six indicators.
Branch if Any On in Plural Indicator Test	BEX 1 or 2	Not All	≠ (Example)	Example has A8421 bits and tests all indicators except wrong length record.

Note: These indicators are reset at the beginning of the next I/O operation

● Figure 19. Branch If I/O Channel Status Indicator On Instructions

locked if the status test is not satisfied. This status test is satisfied if either:

1. A branch if any I/O channel status indicator on instruction X- or R (I) ≠ is given before the next I/O unit instruction, or;

2. A specific X- or R (I) d instruction (see Figure 19) is given and results in a branch before the next I/O unit instruction. Figure 19 is a chart of the I/O channel status indicators, the d-characters that test them, and a brief description of their operation. For more detailed information concerning these indicators, see "Input and Output Operations."

If an R (I) ≠ (BA8421 bits in the d-character) instruction is given following a channel 1 input-output operation, this instruction tests all channel 1 indicators and, if any of them are on, the program branches to the specified I-address. Then the program can test the indicators (individually or in groups, determined by the bit structure of the d-characters) to determine the exact condition present. This technique saves total program execution time because individual test instructions need be given only if processing conditions call for them.

If the system is equipped with the processing overlap feature, the program should test the overlap-in-process indicator with a test and branch instruction. This is done to ensure that the overlapped I/O function is complete before the branch if I/O channel status indicator on instruction is given. A branch if I/O channel status indicator on instruction, executed while the system is performing an overlap operation, usually causes the system to interlock until the overlapped function is complete. See "Overlap Operational Considerations," under "Processing Overlap."

Word Marks: No special considerations.

Timing: T = 4.5 (L + 1 + C).

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Branch	NSIB	BI	NSIB
No branch	NSI	BI	BI

Branch if Character Equal

Instruction Form:

Mnemonic	Op Code	I-address	B-address	d-character
BCE	B	iiiii	bbbbb	x

Function: This instruction causes the bit configuration (BA8421 bits) of the character at the B-address to be compared to the bit configuration of 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 exactly the same, the program continues with the next sequential instruction. This instruction also results in the setting of the high, low, or equal indicator. The high indicator is set if the B-address character is higher than the d-character (collating sequence).

Word Marks: Word marks do not affect this operation. The nature of the instruction specifies that only one character is to be included in the test.

Timing: T = 4.5 (L + 2.5 + C).
L = 1, 6 or 12.
An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Branch	NSIB	BI	NSIB
No branch	NSI	BI	B-1

Branch if Bit Equal

Instruction Form:

Mnemonic	Op Code	I-address	B-address	d-character
BBE	W	iiiii	bbbbb	x

Function: This instruction causes the character at the B-address to be compared, bit by bit, with the d-character. If any bit in the character at the B-address matches any bit in the configuration of the d-character, the program branches to the I-address (WM and C bits not compared). For example, if position 0 5 8 9 6 (B-address) contains a Z (A81 bits) and the d-character contains a 3 (C21 bits), the program branches.

Word Marks: Word marks cannot be tested with this instruction and have no effect on the operation.

Timing: T = 4.5 (L + 2.5 + C.)
L = 1, 6 or 12.
An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Branch	NSIB	BI	NSIB
No branch	NSI	BI	B-1

Branch if Word Mark Present or Zone Equal

Instruction Form:

Mnemonic	Op Code	I-address	B-address	d-character
xxx	V	iiiii	bbbbb	x

(See Figure 20)

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. A 1 bit in the d-character examines the B-address for a word mark. A 2 bit compares the zone bits of the B-address character against the zone bits in the d-character. A combination of the 1 bit and 2 bit allows either a word mark or a correct zone bit comparison to cause a branch to the specified I-address. The criterion for a correct zone-bit comparison is established by the zone bits present, or absent, in the actual d-character. Both actual and mnemonic

d-characters, and the conditions they test, are shown in Figure 20. If the program does not branch to the I-address, it continues with the next sequential instruction.

Word Marks: A word mark is not required at the B-address to stop transmission because this is always a one-character operation.

Timing: T = 4.5 (L + 2.5 + C).
L = 1, 6, or 12.

An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Branch	NSIB	BI	NSIB
No branch	NSI	BI	B-1

INSTRUCTION	ACTUAL OP CODE, ADDRESSES, AND ACTUAL d-CHARACTER	MNEMONIC OP CODE, ADDRESSES, AND MNEMONIC d-CHARACTER	TRANSLATION
Branch if Word Mark Present	V(I)(B)1	BW(I)(B)	Branch to I-address if B-address has a WM-bit.
Branch if Zone Bits Absent	V(I)(B)2	BZN(I)(B)	Branch to I-address if B-address has no B nor A bits.
Branch if Zone Equal AB	V(I)(B)B	BZN(I)(B)AB or BZN(I)(B)+	Branch to I-address if B-address has both B and A bits (plus test).
Branch if Zone Equal B	V(I)(B)K	BZN(I)(B)B or BZN(I)(B)−	Branch to I-address if B-address has a B bit but no A bit (minus test).
Branch if Zone Equal A	V(I)(B)S	BZN(I)(B)A or BZN(I)(B)&	Branch to I-address if B-address has an A bit but no B bit.
Branch if Word Mark Present, or Zone Bits Absent	V(I)(B)3	BWZ(I)(B)	Branch to I-address if B-address has either a WM-bit or no B nor A bits.
Branch if Word Mark Present, or Zone Equal AB	V(I)(B)C	BWZ(I)(B)AB or BWZ(I)(B)+	Branch to I-address if B-address has either a WM-bit or both B and A bits.
Branch if Word Mark Present, or Zone Equal B	V(I)(B)L	BWZ(I)(B)B or BWZ(I)(B)−	Branch to I-address if B-address has either a WM bit or a B bit but no A bit.
Branch if Word Mark Present, or Zone Equal A	V(I)(B)T	BWZ(I)(B)A or BWZ(I)(B)&	Branch to I-address if B-address has either a WM bit or an A bit but no B bit.

● Figure 20. Branch if Word Mark Present or Zone Equal Instructions

These operations are used to manipulate data within core storage during processing. They include data moving, comparing, table lookup, and editing.

Data Moving

This operation concerns moving data, either left-to-right or right-to-left, from the A-field to the B-field (with or without word marks). Data can be moved by fields or by records. If a data field is moved, the operation can be programmed to stop at:

1. A word mark in the A-field.
2. A word mark in the B-field.
3. A word mark in either field.

If a record is moved, the operation can be programmed to stop at:

1. A record mark in the A-field.
2. A group-mark — word-mark in the A-field.
3. Either a record mark or group-mark — word-mark in the A-field.

The operation code for the move instruction is $\check{D}$. The bit structure of the d-character used with the move instruction determines the type of operation that will be performed. (In Figure 21, all 64 characters composed of the bits shown in the figure are valid; each one accomplishes a special purpose.) These operations are:

1. The transfer of the numeric portion of the data field.
2. The transfer of the zone portion of the data field.
3. The transfer of word marks from the A-field to the B-field.
4. The scanning of the A-field and B-field for word marks, record marks, or group-mark — word-marks (this operation is used when the storage positions containing the stated symbols can vary from one record to another — no data are transferred).

Move Instructions

Instruction Form:

Mnemonic	Op Code	A-address	B-address	d-character
x - - x	$\check{D}$	aaaaa	bbbbb	x

(See Figure 24)

Function: Data are moved from left to right or from right to left, serially by character, from the A-field to the B-field under control of the d-character (Figure 21).*

The portion of the A-field that is transferred replaces only the corresponding portion of the B-field. If data are moved from left to right, the A-address specifies the high-order position of the A-field; the B-address specifies the high-order position of the B-field. If data are moved from right to left, the A-address specifies the low-order position of the A-field; the B-address specifies the low-order position of the B-field. The position that contains the terminating character is moved or replaced the same as the rest of the field.

d–CHARACTER CONTROL BITS		CONTROL
1		Transfer of numeric portion of data field
2		Transfer of zone portion of data field
4		Transfer word marks from A–field to B–field
Blank (No 1, 2, or 4 Bit)		Scan for word marks, record marks, or group-mark — word-marks
8–BIT (LEFT TO RIGHT MOVE)	No B and No A Bits	Stop transfer or scan at first word mark sensed in either field
	* A–Bit Only	Stop transfer or scan at A–field record mark
	B–Bit Only	Stop transfer or scan at A–field group-mark — word-mark
	B and A Bits	Stop transfer or scan at A–field record mark or group-mark — word-mark
NO 8–BIT (RIGHT TO LEFT MOVE)	No B and No A Bits	Transfer or scan only one storage position
	* A–Bit Only	Stop transfer or scan at A–field word mark
	B–Bit Only	Stop transfer or scan at B–field word mark
	B and A Bits	Stop transfer or scan at first word mark sensed in either field

*Whenever the A-bit d-character modifier is used in instructions to write programs on tape, the odd parity mode should be used. See Figure 94.

Figure 21. d-Character Control Bits for Move Instructions

This same instruction, with the appropriate d-characters, is also used for scan operations (no data transferred).

Word Marks: See Figure 21.

Timing: $T = 4.5 (L + 1 + A + 1.5B)$.

L = 1, 6 or 12

An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Address Registers after Operation: See Figure 22.

Mnemonics

Because each mnemonic character has a special meaning (Figure 23), it is possible to construct the entire mnemonic for any of the sixty-four move instructions by applying certain rules. These rules are:

DATA TRANSFERRED

1. The first character of the mnemonic is M.
2. The second character of the mnemonic specifies the direction of data movement, either left-to-right or right-to-left (L is right-to-left; R is left-to-right).
3. The third section of the mnemonic specifies the portion of data moved. If only one portion of data is moved, this section contains a single mnemonic character (W, Z, N, or C). If word marks and one other portion of data are moved, this section contains two mnemonic characters (ZW, NW, or CW).
4. The fourth section of the mnemonic specifies the terminating condition. If more than one data character is moved, the terminating mnemonic character is A, B, blank, R, G, or M. If only one data character is moved, the terminating mnemonic character is S.

NO DATA TRANSFERRED (SCAN)

1. The first three characters of the mnemonic are SCN.
2. The fourth character of the mnemonic specifies the direction of scan, either L or R.
3. The fifth character of the mnemonic specifies the terminating condition. The terminating mnemonic character is A, B, blank, R, G, M, or S.

CONTROL	MNEMONIC CHARACTER	MEANING	DESCRIPTION
Direction or Type of Operation	M	Move	Move data serial by character
	SCN	Scan	Affect A- and B-address registers only, do not move data
	L	Left	Right to left operation
	R	Right	Left to right operation
Portion of Data Transferred	N	Numeric	Move only numeric portion of data
	Z	Zone	Move only zone portion of data
	C	Character	Move character(s) (zone and numeric portions of data)
	W	Word Mark	Move word mark(s)
TERMINAL POINT	A (L)	A-Field Word Mark	Stop at A-field word mark
	B (L)	B-Field Word Mark	Stop at B-field word mark
	blank (L or R)	Either A- or B-Field Word Mark	Stop at first word mark sensed in either A- or B-field
	S (L)	One Position	Affect only one position
	R (R)	Record Mark	Stop at A-field record mark
	G (R)	Group Mark	Stop at A-field group-mark—word-mark
	M (R)	Record or Group Mark	Stop at A-field record mark or group-mark — word-mark

Figure 23. Mnemonic Characters for Move Instructions

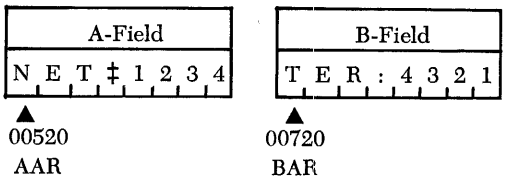
CONTROL	DIRECTION	ADDRESS REGISTERS		
		I–Add. Reg.	A–Add. Reg.	B–Add. Reg.
Stop at first word mark sensed in either field	L to R	NSI	A + LW	B + LW
Stop at A–field record mark	L to R	NSI	A + LA	B + LA
Stop at A-field group-mark—word-mark	L to R	NSI	A + LA	B + LA
Stop at A-field record mark or group-mark—word-mark	L to R	NSI	A + LA	B + LA
Stop after one storage position	R to L	NSI	A − 1	B − 1
Stop at A–field word mark	R to L	NSI	A − LA	B − LA
Stop at B–field word mark	R to L	NSI	A − LB	B − LB
Stop at first word mark sensed in either field	R to L	NSI	A − LW	B − LW

Figure 22. Address Registers after Move Operations

EXAMPLE OF SCAN

Instruction: Ď 00520 00720 Y (mnemonic SCNRR). The most important results shown are the contents of the address registers after the operation. No data are transferred. The B-address must be a part of the instruction, even if, as in the example, the scan is for the first record mark in the A-field exclusively. Because the scan is from left to right, the A- and B-addresses specify the high-order positions of the respective fields.

Before



After

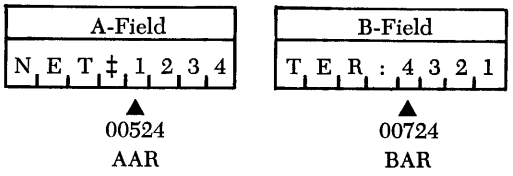


Figure 24 is a complete chart of the data move d-characters and mnemonics.

Move Characters and Suppress Zeros

Instruction Form:

Mnemonic	Op Code	A-address	B-address
MCS	Ž	aaaaa	bbbbb

Function: This instruction causes the data in the A-field to be moved to the B-field. The A-field remains

Direction of Move	Condition Which Ends Operation	No Portion Moved	Move Numeric Portion of A-Field to B-Field	Move Zone Portion of A-Field to B-Field	Move Numeric and Zone from A-Field to B-Field	Move WM in A-Field to B-Field	Move Numeric and WM from A-Field to B-Field	Move Zone and WM from A-Field to B-Field	Move Numeric, Zone, and WM from A-Field to B-Field	d-Ch BCD Coding (BA8 Bits)
RIGHT TO LEFT	Move data one position	blank SCNLS	1 MLNS	2 MLZS	3 MLCS	4 MLWS	5 MLNWS	6 MLZWS	7 MLCWS	NONE
	Move data through 1st WM in A-field	ъ SCNLA	/ MLNA	S MLZA	T MLCA	U MLWA	V MLNWA	W MLZWA	X MLCWA	A
	Move data through 1st WM in B-field	— SCNLB	J MLNB	K MLZB	L MLCB	M MLWB	N MLNWB	O MLZWB	P MLCWB	B
	Move data through 1st WM in either A- or B-field	& SCNL	A MLN	B MLZ	C MLC	D MLW	E MLNW	F MLZW	G MLCW	B, A
LEFT TO RIGHT	Move record through 1st WM in either A- or B-field	8 SCNR	9 MRN	0 MRZ	# MRC	@ MRW	: MRNW	> MRZW	√ MRCW	8
	Move record through 1st RM in A-field	Y SCNRR	Z MRNR	‡ MRZR	, MRCR	% MRWR	ṁ MRNWR	\ MRZWR	⧺ MRCWR	A, 8
	Move record through 1st GM-WM in A-field	Q SCNRG	R MRNG	! MRZG	$ MRCG	* MRWG	] MRNWG	; MRZWG	Δ MRCWG	B, 8
	Move record through 1st RM or GM-WM in A-field	H SCNRM	I MRNM	? MRZM	• MRCM	□ MRWM	[MRNWM	< MRZWM	⧺ MRCWM	B, A, 8
	d-Ch BCD Coding (421 Bits)	NONE	1	2	2, 1	4	4, 1	4, 2	4, 2, 1	

Figure 24. Data Move d-Characters and Mnemonics

unchanged after the operation. High-order zeros and commas in the B-field are replaced by blanks, and zone bits in the units (sign) position of the B-field are removed. Refer to Figure 25 for an example of move characters and suppress zeros.

Example	Op Code	A-address	B-address
Move Char. and Suppress Zeros	v Z	xxxxx	xxxxx
Storage before		A-field (data) v ± 001206	B-field (data) v v ± bbbbbb
Storage after		v ± 001206	bb1206

Figure 25. Move Characters and Suppress Zeros Example

Figure 26 is another example of the move characters and suppress zeros instruction, but one involving a multiple field transfer. In this operation there are effectively two groups of high-order zeros. Alphabetic characters and most special characters (for example, the @ sign) are 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.

Example	Op Code	A-address	B-address
Move Char. and Suppress Zeros	v Z	xxxxx	xxxxx
Storage before		A-field (data) v ± 0010b@00.25	B-field (data) v v v ± bbbbbbbbbbb
Storage after		v ± 0010b@00.25	bb10b@bb.25

Figure 26. Move Characters and Suppress Zeros Example, Multiple Field

Word Marks: The A-field must have a defining word mark. It is this word mark that specifies the length of the data moved to the B-field. B-field word marks within this specified area, including the high-order position, are removed during the operation.

Timing: T = 4.5 (L + 1 + 4A).
L = 1, 6 or 11.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A−LA	B+1

Comparing

The IBM 1410 compares data fields by testing the bit structure of each character in the B-field with the bit structure of each character in the A-field. All BA8421 bits are compared, but not C bits or word marks. The result of the compare operation is determined by the collating sequence of 1410 characters (see Figure 2). B can be equal to, unequal to, higher than, or lower than A.

Compare

Instruction Form:

Mnemonic	Op Code	A-address	B-address
C	v C	aaaaa	bbbbb

Function: The data in the B-field are compared to the data in the A-field. The comparison is never made A to B, but always B to A. The operation does not change either field. The result of the compare sets the high (B > A), equal (B = A), or low (B < A) indicator, depending on whether the B-field data are high, equal, or low with respect to the A-field. These indicators can be tested by a subsequent test and branch instruction.

Word Marks: The compare operation is terminated by either an A-field or a B-field word mark. If the A-field is shorter than the B-field, it must also have a defining word mark. In this case, the high-compare indicator (B > A) is turned on.

Note: The compare indicators must be tested before the next compare, branch if character equal, or table lookup instruction is executed.

Timing: T = 4.5 (L + 1 + A + B).
L = 1, 6 or 11.

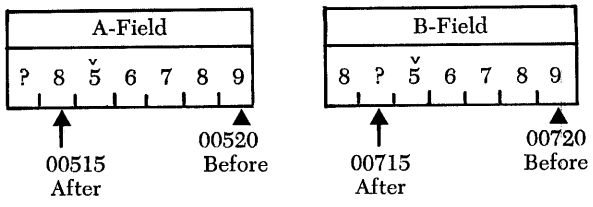
Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A−LW	B−LW

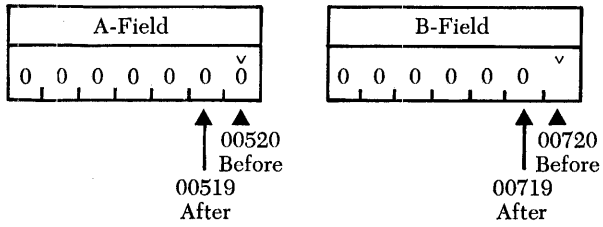
EXAMPLES

Instruction: v C 00520 00720 (used for all examples). The address register contents are shown before and after the compare operation. The result of the compare is above each example.

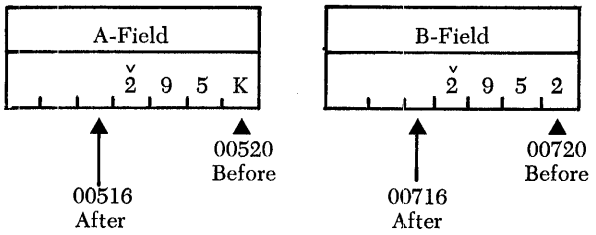
1. Result: B-field is equal to A-field.



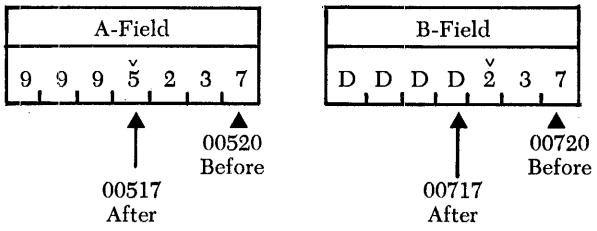
2. Result: B-field is low because of collating sequence.



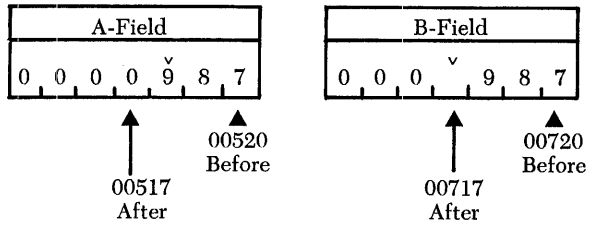
3. Result: B-field is high.



4. Result: B-field is equal to A-field.



5. Result: B-field is high because A-field is shorter.



6. Result: B-field is high, even though negative.

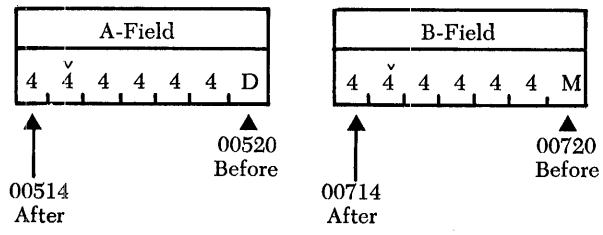


Table Lookup

Many commercial and scientific applications are characterized by the need to search through a table for rates, mathematical factors, or other types of information that vary with the requirements of the input data.

The IBM 1410 Data Processing System has a powerful table lookup instruction that causes the system to search through the table and find the *function* (desired factor or address of desired factor).

To do this, the machine requires two arguments in addition to the function. They are the search argument and the table argument.

Search Argument

The search argument is a data field that has been generated internally or read into the system from a card, tape, disk record, or other input medium. It is used to find the table argument.

Table Argument

The table argument is kept in a table of arguments in core storage. It is exactly the same number of characters as the search argument. If it is shorter, it signifies the end of the table and ends the table search.

Function

The function is kept in core storage with the table argument. If the desired factor is five positions or less, it is often practical to store the factor itself in this place. In this case, the desired factor is the function. If the desired factor is more than five characters, it is usually kept in another area of core storage. In this case, the function is the five-character address of the desired factor. Because the timing of the table lookup operation is determined by the number of characters in the table that are read before a table argument is found, it is desirable to have the least possible number of characters in the function.

Another suggested method for reducing the number of characters in the function is to store the desired factors in a separate table and store the starting address of this table in an accumulator field. If the function contains a factor (less than five characters) that can be added to the starting address to give the actual address of the desired factor, the lookup operation takes less time.

Function values may also be stored a fixed number of core-storage positions from their arguments. Thus, having found the location N of the argument, the function is located at $N + C$, where C is the fixed separation of the functions from the arguments.

Finding the Function

The operation can be programmed to stop when a table argument is found that is equal to the search argument. The program can then move the desired factor to a working area for processing. If the function is the desired factor, it can be moved directly to the working area. If not, it is necessary to bring out the address of the desired factor and then move the factor to the working area.

A table lookup operation can also be stopped if a table argument is found that is higher than, or lower than, the search argument, or when the B-field (table argument) is shorter than the A-field (search argument). The latter condition results in setting the high compare indicator on.

Table Lookup

Instruction Form:

Mnemonic	Op Code	A-address	B-address	d-character
xxx	T	aaaaa	bbbbb	x

(See Figure 27)

Function: This instruction causes the system to search for a table argument that is equal to, lower than, or higher than the search argument as specified by the d-character. A table lookup operation stops on the first table argument that satisfies the table lookup command. See Figure 27 for valid d-characters used for table lookup instructions.

The A-field contains the search argument. The B-address is the address of the low-order character of the entire table. The number of characters in the table argument must be equal to the number of characters in the search argument. However, the field in the table that contains the table argument and the function can be longer than the search argument.

At the end of the operation, one of the high, low, or equal compare indicators is turned on as a result of the last field compared.

Word Marks: The search argument (A-field) must have a word mark set to define the high-order position. The table field (including the argument in the low-order positions and the function in the high-order position) must have a defining word mark in its high-order position. The A-field word mark stops the comparison against the table argument. The system starts to compare again at the position immediately at the left of the word mark in the table field (low-order position of next table argument).

Timing: T = 4.5 (L + 1 + B + NA).
L = 1, 6 or 12.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	Address of the function at immediate left of the table argument that stopped the operation.

An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Note: If a table field is found that is shorter than the search argument, the B-address register will contain the address of the position at the immediate left of the short table field. Thus, a short table field can be used to signal the end of the table. This condition results in setting the high compare indicator on.

DESCRIPTION	MNEMONIC	d-CHARACTER	TABLE SEARCH RESULT
Lookup Low	LL	1	Lower than search argument
Lookup Equal	LE	2	Equal to search argument
Lookup Low or Equal	LLE	3	Equal to or lower than search argument
Lookup High	LH	4	Higher than search argument
Lookup Low or High	LLH	5	Lower than or higher than search argument
Lookup Equal or High	LEH	6	Equal to or higher than search argument
Lookup to Any	none	7	Stop on any
Lookup to End	none	blank	Search to end of table

● Figure 27. Valid d-characters for Table Lookup Instructions

Search Argument	TABLE	
	Function (Desired Factor)	Table Argument
1002	v598	1000
	v698	1002
	v297	1003
	v197	1004
	v198	1005
	v298	1006
	v395	1241
	v495	1242
	v695	1243

Figure 28. Table Lookup Operation

Example: Find the unit price of part number 1002. The unit price is the desired factor, and the part number is the search argument and the table argument (Figure 28).

The table is searched from low-order to high-order position. In this case, the search starts with the number 1243. When the search argument equals the table argument (1002), the search stops. The B-address register then holds the address of the units position of the function.

Editing

The IBM 1410 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 numeric output field. Also, unwanted zeros to the left of significant digits can be suppressed (Figure 29). A step-by-step editing process of this example is shown in Figure 35. Thus, editing in the IBM 1410 is the automatic control of zero suppression, inserting of identifying symbols, and punctuation of an output field.

In editing, two fields are needed: the data field and a control field. The data field is the data to be edited for output. The control field specifies the conditions for the edit operation (how the data field is to be edited). It specifies the location of punctuation and condition of special characters, and indicates where zero suppression is to occur.

The control field is divided into two parts: the body (used for punctuating the A-field) and the status portion (containing the special characters). The *body* of the control word begins with the rightmost blank or zero and continues to the left until the A-field word mark is sensed. The remaining portion of the control field is the *status* portion. Sign printing is partly controlled by the sign of the A-field.

An edit operation requires two instructions. A move instruction is used to transfer the control word and its word mark to the output area; the edit instruction moves the data to the output area and performs the editing function.

Example	Op Code	A-address	B-address
Edit Inst.	E	12163	04685
		A-field (data)	B-Field (control word)
Storage		00257426	$bbb,bb0.bb&CR&**
Result of Edit			B-field
Storage		00257426	$ 2,574.26 **

Figure 29. Editing

Move Characters and Edit

Instruction Form:

Mnemonic	Op Code	A-address	B-address
MCE	E	aaaaa	bbbbb

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 alternately, character by character, under control of the word marks and the editing specifications. See "Editing Specifications." Any sign in the units position of the data field is removed during the operation.

Word Marks: A word mark must be set in the high-order position of the B-field to control the edit operation. The A-field must also have a defining word mark. When the A-field word mark is sensed, the remaining commas in the B-field are set to blanks. The edited output field does not contain any A-field data that have not been moved before the word mark for the control field is sensed. The data field can contain fewer, but should not contain more, positions than the number of blanks and zeros in the body of the control word.

Timing: $T = 4.5(L + 1 + A + 1.5B + 1.5Z + 1.5D)$.
$L = 1, 6$ or 11.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	Varies with result of edit.

Editing Specifications

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

CONTROL CHARACTER	FUNCTION
b (blank)	Replaced with the character from the corresponding position of the A-field.
0 (zero)	Used for zero suppression. Replaced with a corresponding character from the A-field. The rightmost 0 in the control word indicates the rightmost limit of zero suppression.
. (point)	Remains in the edited field in the position where written, unless decimal control was in effect, and the data field did not contain a significant digit. (See "Decimal Control.")
, (comma)	Undisturbed in the output data field in the position where written, unless zero suppression takes place and no significant numeric character is found at the left of the comma.
CR (credit)	Body portion: undisturbed in the position where written. Status portion: if sign of the data field is plus, these two positions are

CONTROL CHARACTER	FUNCTION
	replaced by blanks. If the sign of the data field is minus, they are undisturbed in the output field in the positions where written. (See also "Sign Control Left.")
— (minus)	Same as CR.
& (ampersand)	Causes a blank space in the output field. It can be used in multiples.
* (asterisk)	Status Portion: undisturbed in the position where written. Body Portion: (See "Asterisk Protection.")
$ (dollar)	Status Portion: undisturbed in the position where written. Body Portion: (See "Floating Dollar Sign.")

Zero Suppression

Zero suppression is the deletion of unwanted zeros at the left of significant digits in an output field (Figure 30).

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

EXAMPLE:	
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 30. Zero Suppression

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.
2. When the special zero is detected in the control field, it is replaced by the corresponding digit from the A-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. All zeros and punctuation at the left of the first significant character (up to and including the zero suppression code position) are replaced by blanks in the output field.
2. When the automatically set zero-suppression word mark is sensed, it is erased and the operation ends.

Asterisk Protection

When it is necessary to have asterisks appear at the left of significant digits, the asterisk protection feature is used (Figure 31).

The control word is written with the asterisk in the body to the left of the zero-suppression code (if the asterisk appears in the body to the right of the zero-suppression code, it is treated as a blank).

EXAMPLE:	
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 31. Asterisk Protection

FORWARD SCAN

1. The normal editing process proceeds until the asterisk is sensed.
2. The asterisk is replaced (in the output field) by the corresponding digit from the A-field.
3. The editing process continues normally until the B-field word mark is sensed and removed.

REVERSE SCAN

1. Zeros, blanks, and punctuation to the left of the first significant digit are replaced by asterisks.
2. The word mark (set during the forward scan) signals the end of editing. It is erased, and the operation stops.

Note: Asterisk protection and floating dollar sign cannot be used in the same control word.

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 (Figure 32).

EXAMPLE:	
A-field	00257426
Control word (B-field)	bbbb,b$0.bb
First forward scan	b002,574.26
Reverse scan	bbb2,574.26
Second forward scan	$2,574.26
Results of edit	$2,574.26

Figure 32. Floating Dollar Sign

The control word is written with the "$" in the body to the left of the zero-suppression code (if the dollar sign appears in the body to the right of the zero-suppression code, it is treated as a blank).

Three scans are necessary to complete this editing operation.

FIRST FORWARD SCAN

1. The editing proceeds until the "$" is sensed.
2. The "$" is replaced (in the output field) by the corresponding digit from the A-field.
3. Editing continues until the B-field word mark is sensed and removed.

REVERSE SCAN

1. Zeros and punctuation to the left of the first significant digit are replaced by blanks.
2. The reverse scan continues until the word mark (set over the zero-suppression code 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.

Note: Floating dollar sign cannot be used at the right of the decimal point. Also, floating dollar sign and asterisk protection cannot be used in the same control word. If floating dollar sign protection is needed for data having cents but no dollars, decimal control must *not* be used at the same time as floating dollar sign control. For example, if both controls are in use and the control word is v bb$.b0, an A-field of 00025 is edited to .25 and not $.25. To include the floating dollar sign and a decimal point in the edited result without activating the decimal control, place the zero-suppression code to the *left* of the point in the control word (example: v b$0.bb).

Sign Control Left

CR or -- symbols can be placed at the left of a negative field (Figure 33).

EXAMPLE:	
A-field	v 00378940-
Control word (B-field)	v CR&bbb,bb0.bb
Forward scan	CRb003,v789.40
Reverse scan	CRbbb3,789.40
Results of edit	CR 3,789.40

Figure 33. Sign Control Left

The control word is written with the CR or – symbols in the high-order status position.

FORWARD SCAN

1. The scan proceeds until the zero-suppression code (0) 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. The scan proceeds until the B-field word mark is sensed, indicating the end of the body of the control word.
5. Editing continues and the CR or – 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, the CR or – is blanked.

REVERSE SCAN

1. Zeros and punctuation are replaced by blanks in the output field. 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 34).

The control word is written with a point in the body to the left of the zero-suppression code (0).

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

EXAMPLES:	
1. A–field	v 00000
Control word (B–field)	v bbb.b0
First forward scan	000.v00
Reverse scan	bbb.00
Second forward scan	bbb
Results of edit	(Blank Field)
2. A–field	v 29437
Control word (B–field)	v bbb.b0
First forward scan	294.v37
Reverse scan	294.37
Result of edit	294.37
3. A–field	v 00001
Control word (B–field)	v bbb.b0
First forward scan	000.v01
Reverse scan	bbb.01
Results of edit	.01

Figure 34. Decimal Control

FIRST FORWARD SCAN

1. When the zero-suppression code (0) is sensed during editing, this position is replaced by the corresponding digit from the A-field.

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.

REVERSE SCAN

1. Zeros and punctuation are replaced by blanks in the output field 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. The zeros at the right of the decimal point, and the decimal point itself, are replaced by blanks.

2. The operation stops at the decimal column.

Figure 35 is a step-by-step editing process of the example shown in Figure 29.

STEP	TYPE OF CYCLE	ADDRESS REGISTERS			DATA REGISTER		PUT BACK INTO STORAGE	B-FIELD AT END OF CYCLE	REMARKS
		I	A	B	B	A			
1	Iop	00002	?????	?????	v E	v E	v E	v $bbb,bb0.bb&CR&**	Read Instruction Op Code
2	I1	00003	1????	?????	1	1	1	Same	Load A-address register
3	I2	00004	12???	?????	2	2	2	Same	Load A-address register
4	I3	00005	121??	?????	1	1	1	Same	Load A-address register
5	I4	00006	1216?	?????	6	6	6	Same	Load A-address register
6	I5	00007	12163	?????	3	3	3	Same	Load A-address register
7	I6	00008	12163	0????	0	0	0	Same	Load B-address register
8	I7	00009	12163	04???	4	4	4	Same	Load B-address register
9	I8	00010	12163	046??	6	6	6	Same	Load B-address register
10	I9	00011	12163	0468?	8	8	8	Same	Load B-address register
11	I10	00012	12163	04685	5	5	5	Same	Load B-address register
12	I11	00012	12163	04685	v Op	v Op	v Op	Same	Op Code & next instruction
13	A	00012	12162	04685	6	6	6	Same	Execute EDIT instruction
14	B	00012	12162	04684	*	6	*	Same	
15	B	00012	12162	04683	*	6	*	Same	
16	B	00012	12162	04682	&	6	Blank	v $bbb,bb0.bb&CRb**	
17	B	00012	12162	04681	R	6	Blank	v $bbb,bb0.bb&Cbb**	
18	B	00012	12162	04680	C	6	Blank	v $bbb,bb0.bb&bbb**	
19	B	00012	12162	04679	&	6	Blank	v $bbb,bb0.bbbbbb**	
20	B	00012	12162	04678	b	6	6	v $bbb,bb0.b6bbbb**	
21	A	00012	12161	04678	2	2	2	Same	
22	B	00012	12161	04677	b	2	2	v $bbb,bb0.26bbbb**	
23	A	00012	12160	04677	4	4	4	Same	

Figure 35. Step-by-Step Editing Process

STEP	TYPE OF CYCLE	ADDRESS REGISTERS			DATA REGISTER		PUT BACK INTO STORAGE	B-FIELD AT END OF CYCLE	REMARKS
		I	A	B	B	A			
24	B	00012	12160	04676	•	4	•	Same	
25	B	00012	12160	04675	0	4	4	$bbb,bb4.26bbbb**	Zero Suppress
26	A	00012	12159	04675	7	7	7	Same	
27	B	00012	12159	04674	b	7	7	$bbb,b74.26bbbb**	
28	A	00012	12158	04674	5	5	5	Same	
29	B	00012	12158	04673	b	5	5	$bbb,574.26bbbb**	
30	A	00012	12157	04673	2	2	2	Same	
31	B	00012	12157	04672	,	2	,	Same	
32	B	00012	12157	04671	b	2	2	$bb2,574.26bbbb**	
33	A	00012	12156	04671	0	0	0	Same	
34	B	00012	12156	04670	b	0	0	$b02,574.26bbbb**	
35	A	00012	12155	04670	0	0	0	Same	
36	B	00012	12155	04669	b	0	0	$002,574.26bbbb**	
37	B	00012	12155	04668	$	0	$	$002,574.26bbbb**	Sense Word Mark — Rev. Scan
38	B	00012	12155	04669	?	0	?	$002,574.26bbbb**	Units Position of next Field
39	B	00012	12155	04670	$	0	$	Same	
40	B	00012	12155	04671	0	0	Blank	$b02,574.26bbbb**	
41	B	00012	12155	04672	0	0	Blank	$bb2,574.26bbbb**	
42	B	00012	12155	04673	2	0	2	Same	
43	B	00012	12155	04674	,	0	,	Same	
44	B	00012	12155	04675	5	0	5	Same	
45	B	00012	12155	04676	7	0	7	Same	
46	B	00012	12155	04677	4	0	4	$bb2,574.26bbbb**	

Figure 35. (Continued)

This section describes the store address register, set word mark, clear word mark, clear storage, halt, and no-operation instructions that are used to facilitate programming and prepare storage areas for processing data fields.

Store Address Register

This operation makes it possible to store the contents of the A- and B-address registers after any operation. The contents of the E- and F-address registers can be stored after any tape operation in the overlap mode. The store address register operation is particularly useful when fields or records of variable length are being processed, or when a method of linking a main routine with a subroutine is desired. For example, the address of the next sequential instruction is stored in the B-address register after a program branch to the I-address occurs. If the first step of the subroutine stores the contents of the B-address register in the last step of the subroutine (branch unconditional instruction), the program branches back to the next instruction of the main routine after the subroutine is executed.

Instruction Form:

Mnemonic	Op Code	C-address	d-character
xxx	G	ccccc	A, B, E, F

(See Figure 36.)

Function: The contents of the register specified by the d-character are stored in the C-field. The C-address specifies the low-order position of the field in core storage where the register contents will be stored.

Word Marks: Word marks in the C-field have no effect on the operation. *Note:* If there are zones in the C-field, they are not disturbed.

OPERATION	MNEMONIC	d-CHARACTER
Store A-Address Register	SAR	A
Store B-Address Register	SBR	B
Store E-Address Register	SER	E
Store F-Address Register	SFR	F

Figure 36. Store Address Register Mnemonics and d-Characters

Timing: T = 69.75.

Note: This instruction cannot be indexed.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

Set Word Mark

Instruction Form:

Mnemonic	Op Code	A-address	B-address
SW	,	aaaaa	bbbbb

Function: If this instruction is given as shown in the instruction form, a word mark is set in the specified A-address location and in the specified B-address location. The data characters in the specified locations are not disturbed.

If this instruction is given with only one address (A-address), a word mark is set in the specified A-address location only. The data character in the specified location is not disturbed.

If this instruction is given with no address specified (a no-address chained instruction), word marks are set in the address locations that are specified by the A- and B-address registers (contents from the previous operation).

Word Marks: Word marks are explained in the previous paragraph.

Timing: T = 4.5 (L + 4).

L = 1, 6, or 11.

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Two Addresses	NSI	A − 1	B − 1
One Address	NSI	A − 1	A − 1
No Addresses	NSI	Ap − 1	Bp − 1

Clear Word Mark

Instruction Form:

Mnemonic	Op Code	A-address	B-address
CW	□	aaaaa	bbbbb

Function: If this instruction is given as shown in the instruction form, a word mark is cleared, if present, from the specified A-address location and from the specified B-address location. The data characters in the specified locations are not disturbed.

If this instruction is given with only one address (A-address), a word mark, if present, is cleared from the specified A-address location only. The data character in the specified location is not disturbed.

If this instruction is given with no address specified (a no-address chained instruction), word marks, if present, are cleared from the address locations

that are specified by the A- and B-address registers (contents from the previous operation).

Word Marks: Word marks are explained in the previous paragraph.

Timing: T = 4.5 (L + 4).

L = 1, 6 or 11.

Address Registers after Operation:

	I-address Reg	A-address Reg	B-address Reg
Two Addresses	NSI	A – 1	B – 1
One Address	NSI	A – 1	A – 1
No Addresses	NSI	Ap – 1	Bp – 1

Clear Storage

Instruction Form:

Mnemonic	Op Code	B-address
CS	/	bbbbb

Function: A storage area is cleared of data and word marks, right-to-left, from the specified B-address location to, and including, the nearest hundreds position. For example, to clear storage from 12590 to 12500, use a / 12590 instruction. The B-address register, at the end of the operation, will hold 12499.

If this instruction is given with no address specified (a no-address chained instruction), the contents of the B-address register are used as the B-address location. (In this case, the A-address register is not loaded at instruction loading time and is undisturbed at the end of the clear storage operation.) By chaining the instruction in this manner, several blocks of 100 core storage positions can be quickly cleared.

For clearing larger blocks of core storage, a simple program loop (as shown in Figure 37) proves more efficient. This example clears the core storage area from positions 00500 to 36199. The first instruction sets a word mark in the low-numbered position of the core storage area being cleared.

The second instruction starts clearing the specified core storage area at the high-numbered position. This instruction clears the core storage area from 36199 to 36100. (At the end of the operation, the B-address register contains the number 36099.)

INSTRUCTION ADDRESS	INSTRUCTION																
	OP	A/I FIELD					B-FIELD										d
		d	x-ctrl fld			d	d										
00135	,	0	0	5	0	0											
00139	/						3	6	1	9	9						
00145	G	0	0	1	3	4	B										
00142	V	0	0	1	2	9	0	0	5	0	0						1

Figure 37. One Method for Clearing Core Storage

The third instruction stores the B-address register contents in core storage, starting at the address specified by the C-address register. The C-address register contains the core storage address that is the units position of the clear storage B-field. After the operation, core storage positions 00130-00134 contain the number 36099.

The fourth instruction tests core storage position 00500 for the word mark that was previously put there. If the word mark is still there, the program branches to the specified I-address. The I-address is the core storage address that contains the clear storage operation code. The next 100 positions of core storage are cleared.

When the last group of 100 core storage positions is cleared, the word mark is removed from core storage position 00500. The test instruction is performed, but no branch occurs. This signifies that the clear operation is complete, and the program proceeds with the next sequential instruction.

Word Marks: Word marks are cleared in the area specified.

Timing: T = 4.5 (L + 1 + B).

L = 1 or 6.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	B	bbb00-1

Clear Storage and Branch

Instruction Form:

Mnemonic	Op Code	I-address	B-address
CS	/	iiiii	bbbbb

Function: This instruction has the same effect as clear storage except that the next instruction is taken from the I-address. This is an unconditional branch instruction because the branching does not depend on any condition.

Word Marks: Word marks are cleared in the storage area specified by the B-address.

Timing: T = 58.5.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSIB	BI	NSI

Halt

Instruction Form:

Mnemonic	Op Code
H	.

Function: The system stops. Pressing the start key starts system operation with the next sequential instruction.

Word Marks: Word marks are not affected. If this is the last instruction in the program, a word mark must be preset in the storage location immediately to the right of the operation code.

Timing: T = 4.5.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

Halt and Branch

Instruction Form:

Mnemonic	Op Code	I-address
H	v	iiiii

Function: The system stops. When the start key is pressed, the program resumes with the instruction located at the I-address. This is an unconditional branch instruction, because the branching does not depend on any condition.

Word Marks: Word marks are not affected. If this is the last instruction in the program, a word mark must be preset in the storage location immediately to the right of the halt and branch instruction.

Timing: T = 36.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSIB	BI	NSI

No Operation

Instruction Form:

Mnemonic	Op Code
NOP	v N

Function: This operation code can be substituted for the operation code of any instruction to make that instruction ineffective.

Word Marks: Word marks are not affected.

Timing: T = 4.5 (L + 1).

L = 1, 2, 3 . . . . . No Limit

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

Through the IBM 1415 Console (Figure 38), the operator of a 1410 system can enter information, display core storage contents, and read records from tape or disk storage units. The console consists of an I/O printer, a control section, an indicator-light panel, and includes desk space.

Console I/O Printer

The I/O printer on the IBM 1415 Console can:

1. Provide an operating log of all major manual console I/O printer operations (reset key operations are not logged but are indicated by a carrier return and vertical space). All alterations to internal data and their addresses are logged by the console printer. Before the alteration is made, the data must be displayed on the console printer.
2. Provide display facilities for some registers and all storage locations.
3. Provide an inquiry mode of operation under control of the console operator.
4. Provide messages under program control. A programmed print-out can be overlapped with compute if the overlap feature is included.
5. Provide print-out of the instruction address register, A- and B-address registers, Op register, Op-modifier register, A- and B-channel contents, assembly channel contents, and the unit select and unit number register for both channels 1 and 2 on manual-stop, programmed-stop, and error-stop operations (Figure 39).

The printing mechanism of the console-I/O printer is an IBM Selectric® typewriter; it can print 64 characters (10 numeric, 26 alphabetic, 28 special), a word-mark symbol, and an underscore symbol (invalid bit parity print-out). This printer has no type bars or movable carriage. Instead, it has a sphere-shaped type head containing all of the characters. The type head moves from left to right across the paper during a printing operation. Maximum rate of the console-I/O printer is 932 characters per minute. The paper forms have feed holes in the left and right margins. Vertically, the holes must be ½ inch apart; horizontally, the rows must be 9⅜ inches apart.

Because this typewriter is used as the console-I/O printing mechanism, the functions of vertical spacing (indexing or line spacing), backspacing, and type head carrier return are inoperative from the keyboard.

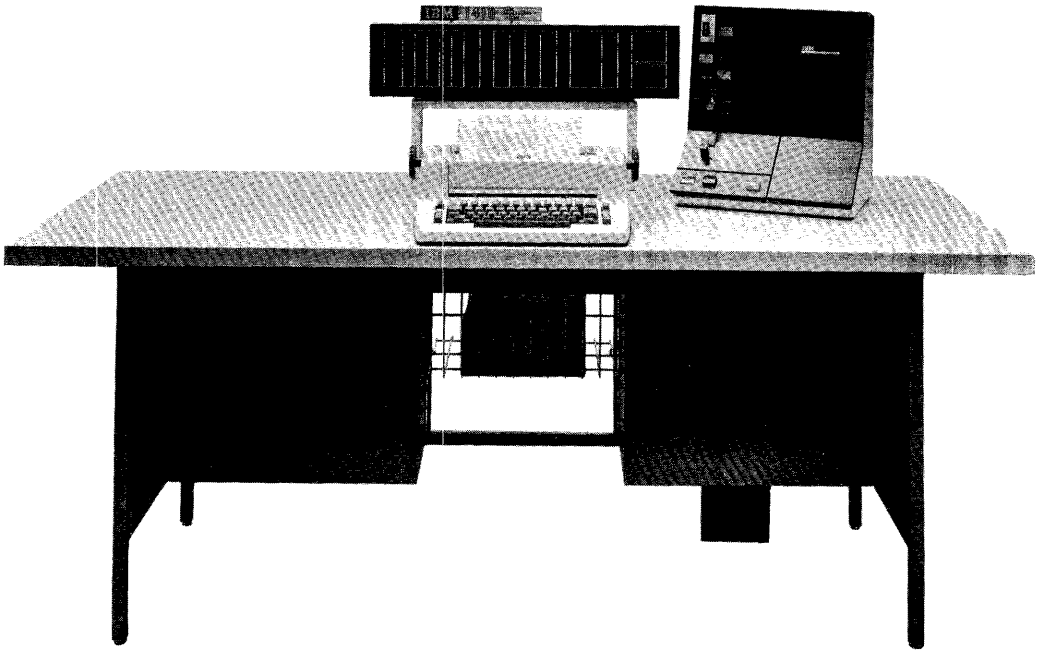


Figure 38. IBM 1415 Console

OPERATION AND VERTICAL SPACES BEFORE PRINT-OUT	PRINTOUT IDENT.	SPACE	I A R	SPACE	A A R	SPACE	B A R	SPACE	OPERATION CODE OP MODIFIER	SPACE	A CHANNEL CONTENTS B CHANNEL CONTENTS ASSEMBLY CHANNEL	SPACE	CH. #1 UNIT SEL REG UNIT NUM REG	CH. #2 UNIT SEL REG UNIT NUM REG
NORMAL STOP (Double Space)	S		XXXXX		XXXXX		XXXXX		XX		XXX		XXXX	
HALF-CYCLE (Double Space)	C		XXXXX		XXXXX		XXXXX		XX		XXX		XXXX	
ERROR STOP (Double Space)	E		XXXXX		XXXXX		XXXXX		XX		XXX		XXXX	
ADDRESS SET (Single Space)	B (Note)		XXXXX											
STORAGE SCAN SET (Single Space)	# (Note)		XXXXX											
DISPLAY (Single Space)	D D		XXXXX XXXXXXXX											
ALTER (Single Space)	A		XXXXXXXX		Not IAR read-out; length of printing line determined by margins set, operator's line length, and write field length.									
CONSOLE INQUIRY (Single Space)	I		XXXXXXXX											
CONSOLE REPLY (Single Space)	R		* XXXXXXXX											

* ___ Indicated Invalid Character (Underlined)

Note: Print-out B if address entry switch is set on Normal; all other positions of the switch cause a print-out #.

● Figure 39. IBM 1415 Printing Layout

Console Printer Control Keys and Levers

INQUIRY KEYS

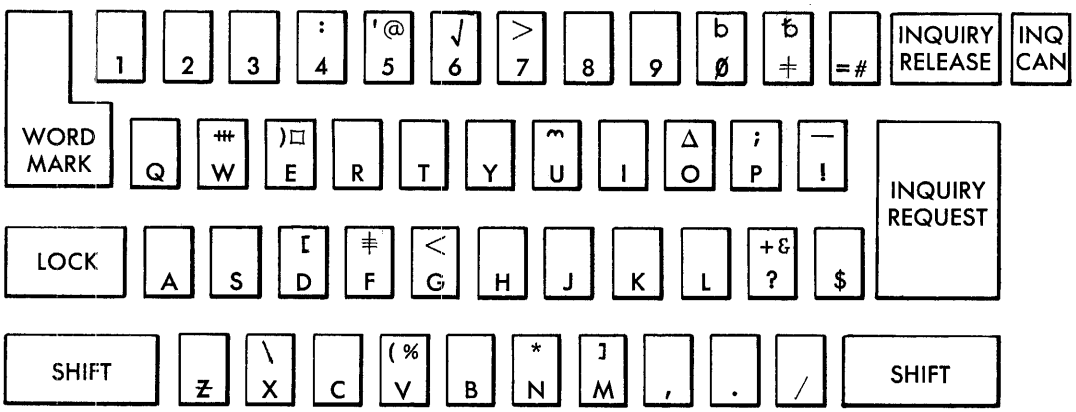
The IBM 1415 Console can be used as an inquiry station by using the console inquiry keys (Figure 40). The entry of inquiries and the print-out of their replies are under program control, and can occur while the system is operating in either the run mode or the I/E-cycle mode.

The use of each key is discussed in the order of use during an inquiry request operation.

REQUEST KEY

A console inquiry is initiated by pressing the inquiry-request key. A signal requesting permission to process a console inquiry is sent to the 1411 Processing Unit (turns on the inquiry status latch in the 1411).

The inquiry request is discovered by the 1410 system when the program tests the inquiry status latch with a test and branch instruction (J̌ iiii Q). If the latch is set on, the program branches to a subroutine that contains the read console printer instruction M̌ %T0 bbbbb R (Figure 41). Acknowledgment of the inquiry request by the 1410 system causes the character I to be printed, after which the console-I/O printer takes one space and then the keyboard unlocks. Manual entry of the inquiry request, character by character, can now proceed. The first inquiry character is placed in the storage address specified by the



Note: Where two characters are shown in a case (three on a key top), the actual characters printed depend on the type head in use (see Figure 2).

Figure 40. Console-I/O Printer Keyboard

I/O UNIT	X-CTRL FIELD	DESCRIPTION	MNE-MONIC	d-CHARACTER		OPERATION	NOTES
				CHAR-ACTER	CONTROL		
Console-I/O Printer	% T0	Read from console-I/O printer without word marks	RCP	R	Read	Transfer data direct from the console-I/O printer to storage	1. Data transfer is operator-controlled. See console operating features. 2. Word marks in storage are undisturbed.
	% T0	Read from console-I/O printer with word marks	RCPW	R	Read		1. Data transfer is operator-controlled. See console operating features. 2. Word marks in storage are erased and entered during a load operation.
	% T0	Write on console-I/O printer without word marks	WCP	W	Write	Transfer data direct from storage to the console-I/O printer and print	1. Group-mark—word-mark is not printed with message. 2. Word marks are not indicated.
	% T0	Write on console-I/O printer with word marks	WCPW	W	Write		1. Group-mark—word-mark is not printed with message. 2. Word marks are indicated.

Figure 41. Console-I/O Printer Control Instructions

read console instruction B-address. Subsequent characters are placed in the next higher storage positions.

If a request is made but has not yet been recognized, pressing the cancel key resets the inquiry status latch. If an error is recognized while the message is being printed, pressing the cancel key sets the condition i/o channel-status indicator on, and the program continues. Figure 42 shows the conditions that set the i/o channel-status indicators and turn on their associated lights during a console printer read operation.

Error Condition: Any system error stops the system and initiates an error print-out operation; the inquiry request operation is ended.

INDICATOR	d-CHARACTER BIT	CONDITION
Not Ready	1	Never set
Busy	2	Never set
Data Check	4	Processing unit detects input character validity error
Condition	8	Cancel Key operated during inquiry
Wrong Length Record	B	Wrong length record
No Transfer	A	No message request — Cancel Key operated before inquiry

Figure 42. i/o Channel-Status Indicators Set During Console Printer Read Operation

RELEASE KEY

The inquiry is released to the processing unit by pressing the release key after the correct number of characters have entered storage. The programmer has already specified the length of the inquiry (a certain number of characters that occupy specific storage locations). The next higher storage location must contain a previously inserted group-mark — word-mark. As the last inquiry character is entered in its storage location, the addressing circuitry is set up to read out the group-mark — word-mark. The operator must press the release key at this time to obtain a correct-length record, and to ensure the processing of the inquiry.

If it is desirable to request a second inquiry while entering a first inquiry, this can be accomplished by holding down the inquiry request key while pressing the inquiry release key to release the first inquiry. This causes the inquiry status latch to remain on. After a request is initiated, the inquiry status latch can be reset before any characters are entered, by pressing the release key.

Operating the release key also initiates a carrier-return and vertical-space operation, and locks the keyboard.

Wrong-Length Record (Inquiry). Operating the release key, when the number of characters printed is less than the prescribed form on an inquiry request, causes:

1. A carrier-return and vertical-space operation.
2. The wrong-length record i/o channel status indicator to be set on.
3. The program to go to the next instruction.

When the number of characters entered is more than the prescribed form on an inquiry request, additional characters are not accepted by the processing unit. When the operator presses the release key or the cancel key, the wrong-length record i/o channel status indicator turns on but the program continues.

CANCEL KEY

Operating the cancel key ends the inquiry routine in process at that time. Operating the cancel key during the inquiry-request-message printing sets the condition i/o channel status indicator on, releases the system, causes a carrier-return and vertical-space operation, and allows the normal program to resume.

If it is desirable to request a second inquiry while cancelling a first inquiry, this can be done by holding down the inquiry request key while pressing the cancel key to cancel the first inquiry. This causes the inquiry status latch to remain on.

After a request is initiated, the inquiry status latch can be reset off, by pressing the cancel key, before any characters are entered.

WORD MARK KEY

Pressing this key prints a word mark; after the word mark is printed, the carrier is backspaced one position. Pressing a character key then prints the character under the word mark and enters both the word mark and the character into storage. The word mark key must be pressed first in order to enter a character with a word mark into storage.

SHIFT KEYS

Pressing either one of the two shift keys shifts the console printer into upper case. Figure 40 illustrates the console printer keyboard. The characters shown at the top of the keys are upper-case characters and require a shift key to be operated before the character key is pressed. The printer automatically returns to lower-case shift when the key is released.

LOCK KEY

Pressing this key activates the shift keys, and locks the console printer in upper-case shift until released (by pressing one of the shift keys or by locking the keyboard by completing an input operation).

COPY CONTROL LEVER

Operating the copy control lever (at the left rear of the console printer) positions the type-head carrier forward or backward so that various thicknesses of printing material are accommodated. The copy-control lever can be set in five different positions. Moving the lever forward decreases the distance between the platen and printing mechanism; moving the lever to the rear increases this distance.

PAPER RELEASE LEVER

Pulling forward on the paper release lever (the outer lever at the right rear of the printer) releases the pressure of the front and rear feed rolls on the platen. This permits accurate paper positioning and easy paper removal. This lever should be forward when the pin-feed platen is used. It should only be pushed back when it is desired to move the paper vertically backward through the paper feed.

MARGIN-SET LEVERS

The left and right margins are determined by the position of the margin stops. The left or right margin is set by operating the associated margin-set lever (at the top of the keyboard). The margin-set lever is operated by exerting pressure toward the rear of the console printer and sliding the lever to the right or left.

INDEX SELECTOR LEVER

When the index selector lever (the inner lever at the right rear of the console printer) is set toward the rear, the platen double-spaces each line of printing (three lines per inch). With the index selector lever set toward the front of the machine, the platen single-spaces each line of printing (six lines per inch).

Console Reply Routine

A reply routine or programmed print-out can occur at any time. The console write instruction M̆/L̆ %T0 bbbbb W causes:

1. The character R to print.
2. A space.
3. Data to be transferred from storage and printed by the console-I/O printer until a group-mark – word-mark is sensed in storage. (A valid blank in storage causes the console printer to space.)
4. A carrier return and vertical-space operation.
5. The program to continue with the next instruction.

Figure 43 shows the conditions that set the I/O channel-status indicators on and that turn on their associated lights during a console printer write operation.

INDICATOR	d-CHARACTER BIT	CONDITION
Not Ready	1	Never set
Busy	2	Carriage returning
Data Check	4	I/O Printer detects output character validity error
Condition	8	Never set
Wrong Length Record	B	Never set
No Transfer	A	Never set

Figure 43. I/O Channel-Status Indicators Set During Console Printer Write Operation

CPU Processing Error: A CPU processing error during the data transfer ends the reply routine and causes an error print-out operation. Operation of the start key is necessary to complete the programmed print-out.

I/O Printer Error: If a parity error is sensed in the console printer, the error character is printed and underlined. The data-check I/O channel status indicator is also set on. The reply routine continues until a group-mark – word-mark is sensed in storage.

Console Load Read and Write Operations

If the console I/O printer is addressed on a load-write operation (L Op code), blank characters in storage are printed as small b's, and each word mark is printed as an inverted circumflex over the character associated with the word mark.

It is possible to enter word marks into storage during a console-inquiry routine if the instruction calls for a load-read operation. The word mark prints on the log sheet and enters storage. A console printer space operation generates a blank character in storage.

Control Section

The control section (Figures 44, 45, and 53) contains the power keys and lights and other keys and switches that control the 1410 system.

Computer Reset

Operating this key resets the check circuits, resets the program to 00001, resets all timing clocks, and resets all machine indicators (overflow latches, compare triggers, etc.). The inquiry latches (except the console inquiry latch) and the tape density latch are not reset.

Power Keys, Lights, and Switches

The power keys, lights, and switches (Figure 44) control the application of power to the 1410.

EMERGENCY POWER OFF

This pull switch should be used only in case of emergency, when all power must be shut off immediately to prevent injury to an individual or damage to the system. Pulling the switch removes all power from all units.

If this switch is used, only a customer engineer should turn on the power again.

POWER ON

Operating this switch normally provides full operating power to the 1410 system, either from a power-off or a dc-off condition. (Power is applied to the 1414, the tape adapter unit, and the disk-storage control unit only if their respective CE panel power local-remote switches are set to REMOTE.) Also, a power-on reset operation is initiated when power is first applied or reapplied to the system, and causes all system registers, latches, rings, etc., to be reset. The first operation of the switch starts the internal sequencing of power to the system and turns on the illuminated portion of the power-on key. The key light remains on until either the emergency power-off switch or the power-off key is operated.

When system power is fully on, the ready light is turned on.

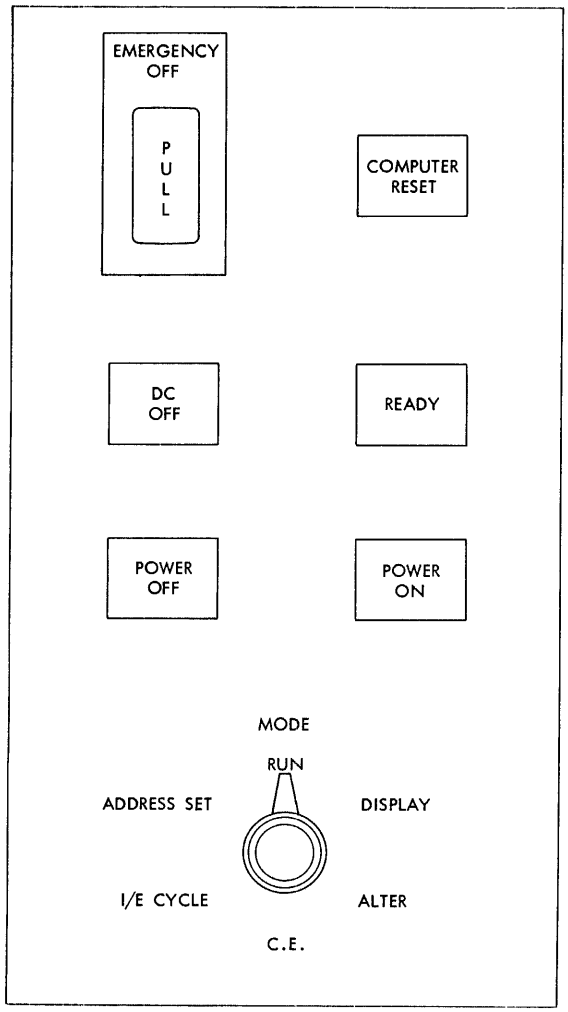


Figure 44. Power Keys, Lights, and Switches

POWER OFF

Pressing the power-off key removes all power from the system, except on units with CE panel power local-remote switch set to LOCAL. Removal of system power also turns off the ready light and the illuminated portion of the power-on key.

To restore full operating power to the system, the power-on key must be pressed.

DC OFF

Pressing this key turns off the system dc power only, except in those units with CE panel power local-remote switch set to LOCAL. The key is used when the system will be idle for a short time. Operating the key turns off the ready light, but the light comes on again as soon as full power is restored to the system. The power-on light remains on.

READY LIGHT

The ready light is turned on when full operating power is applied to the system. It takes a short time for the machine to reach ready status because the power is supplied to the system units or frames in a specified sequence. The ready light turns on immediately if the power-on key is pressed while the machine is in the dc-off mode.

Mode Switch

The six modes of machine operation are selected by the mode switch. The six modes are modified by the CE controls. Usually, these CE controls are set to the normal or off operating mode. Any change in the mode switch setting causes a stop print-out operation, which begins as soon as the execution of the previous setting is complete.

CE (CUSTOMER ENGINEER)

When the mode switch is set to CE, the customer engineering function of storage scan is available for use.

I/E CYCLE

With the mode switch set to I/E CYCLE, the first operation of the start key causes the system to read one complete instruction from storage, then stop and print out. Because this print-out occurs while the machine is in the I/E cycle mode, the print-out is preceded by the printing of a C. The console printer then spaces and prints out the contents of the instruction address register, A- and B-address registers, Op-register, Op-modifier register, A-data register, B-channel contents, assembly channel output, and the unit select and unit number registers for channel 1 and channel 2.

The second operation of the start key causes the execution of that instruction (called the execution phase), and then system operation stops. Another C print-out operation occurs exactly as previously described.

Subsequent operation of the start key results in the system going through alternate instruction and execution cycles.

Exceptions are branch and I/O operations, which may not always follow the above description.

ADDRESS SET

This setting of the mode switch is used to start a program at a specific place in storage. Pressing the stop key or turning the mode switch to the ADDRESS SET position causes a normal stop print-out operation.

The start key is then pressed, and a B-character is printed on the console-I/O printer. (See Note in Figure 39.) The printer then takes a single vertical space. The address that is then entered (by operating the console printer keys) enters the I-address register and is followed by an automatic carrier return and vertical-space operation.

The mode switch may then be positioned at either the RUN setting or the I/E CYCLE setting. Pressing the start key starts the program with the instruction located at the entered address.

This switch setting, when used with the address entry switch on the console CE panel, permits altering the contents of the A-, B-, C-, D-, E-, or F-register, depending on the setting of the switch. With this switch in the NORMAL position, the contents of the I-address register are altered. If any address register is altered, other than the I-address register, the address entry switch must be returned to the NORMAL position before pressing the start key. Pressing the start key starts the program at the unaltered address in the I-address register.

RUN

When the mode switch is set to RUN, pressing the start key causes the system to run continuously under control of the stored program.

A display operation must precede an alter operation. Resetting the mode switch to RUN or I/E CYCLE, and pressing the start key, lets the program proceed at the address in the I-address register.

DISPLAY

Any portion of storage may be displayed on the console-I/O printer log sheet by using the DISPLAY setting of the mode switch. The display may be of any length, from one field to a multiple line print-out.

Operation: During a display operation, this sequence takes place:

1. The system is stopped by operating the stop key or setting the mode switch to DISPLAY.
2. With the mode switch set to DISPLAY, operating the start key results in printing a character D, followed by a space and the unlocking of the keyboard.
3. The high-order address of the field to be displayed is manually entered on the console-I/O printer by operating the appropriate keys.
4. An automatic keyboard-lock, carrier-return, and vertical-space operation takes place following the printing of the fifth address character.
5. A character D is automatically printed, followed by a space.
6. The contents of storage, starting at the high-order position previously printed, are printed until a word mark is recognized. The word mark and its associated character are printed (first character of the adjacent field). The adjacent field can be displayed if the start key is pressed again. A continuous display results from holding the start key in its operated position. The display operation can be ended at any time by pressing the stop key. The display operation is momentarily held up if an end-of-printing-line signal is encountered. An automatic carrier return and vertical-space operation takes place, followed by a resumption of the display operation.

No Wraparound on Display – 10K Core Storage Only: When the last character in storage is printed, the display operation ends and the carrier returns.

Wraparound on Display – 20K, 40K, 60K, 80K Core Storage: When the last character in storage is printed and has no word mark, the display operation continues and the next character printed will come from storage location 00000. The display continues to the next word mark encountered. If the last character in storage is printed and has a word mark, the display operation ends, but may be resumed by pressing the start key again.

Error Conditions: Characters with invalid parity in storage are underscored on the console-I/O print-out. Channel errors are ignored. The carrier returns when the stop key is pressed or the error is reset.

ALTER

By using the ALTER setting of the mode switch, in combination with the console-I/O printer, it is possible to alter the information in any storage location. However, a display operation of the specific storage location must be completed before an alter operation can be performed. This display operation prerequisite ensures having a record of the storage location contents before the alteration takes place.

After the display operation, the alter operation is started by rotating the mode switch from DISPLAY to ALTER and pressing the start key. The character A is printed, signifying an alter operation, followed by a space and the unlocking of the keyboard. Unlocking the keyboard allows the manual-alter printing operation to proceed.

If one or more fields (but less than a full line) were previously displayed, only the first displayed field can be altered. Only the first line from a multiple-line display can be altered.

The correct characters are printed and replace the previously-displayed incorrect data. Correct data are kept by reprinting all the correct characters. Any previously displayed word mark must be re-entered into storage. Valid blanks are entered in storage by operating the space bar, or the blank-character key (b).

The alter operation continues until a word mark is sensed if one or more fields (but less than one line) were displayed. An alter operation ends when the end-of-line condition is sensed if a multi-line display preceded the alter operation. Either one of these conditions locks the keyboard and initiates a carrier-return and vertical-space operation.

If an error other than a data error occurs, the alter routine ends and the carrier returns.

No Wraparound on Alter — 10K Core Storage Only: When a character is entered into the last location of storage, the alter operation stops. The carrier is not returned, and the operator is able to continue typing characters; however, these additional characters are not entered into storage.

Wraparound on Alter — 20K, 40K, 60K, 80K Core Storage: When a character is entered into the last location of storage, the alter operation continues and the next location altered is 00000, unless:

1. The last character in storage is printed at the end of a line, or
2. The last character in storage contains a word mark and the previous operation ended at a word mark.

Conditions 1 and 2 are normal; whenever either exists, the alter operation stops and the carrier returns.

Control Keys

Control keys (Figure 45) include start, stop, and program reset.

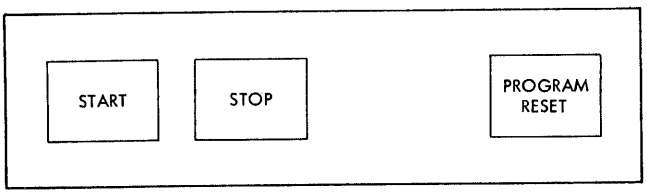


Figure 45. Control Keys

START KEY

With the mode switch set to RUN the operation of the start key resets the parity check circuits and causes the system to begin executing instructions at the address contained in the I-address register. Also, the start key initiates the operation when the mode switch is set to I/E CYCLE, DISPLAY, or ALTER. It is also active in some customer engineering operating modes.

STOP KEY

Operation of the stop key, while the program is running, stops the program after execution of the current instruction. The character S is printed, followed by a space and then a print-out (Figure 39) of the contents of the following: IAR, (space), AAR, (space), BAR, (space), Op register, Op-modifier register, (space), A-channel, B-channel, assembly channel, (space), unit select register for channel 1, unit number register for channel 1, unit select register for channel 2, and unit number register for channel 2.

PROGRAM RESET KEY

Operating this key resets the check circuits, resets the program to 00001, and resets the A- and B-data registers, Op register, Op-modifier register, and console inquiry latch.

Indicator Light Panel

CPU Control Indicator Lights

See Figure 46.

I RING

These lights show the 13 steps of the instruction ring (OP and 1-12).

A RING

These lights show the six steps of the A ring (1-6).

CLOCK

These lights show the ten steps of the main clock (A-K).

SCAN

These lights show what type of address modification is taking place.

The N Light shows that a storage location is operating in a +0 modification cycle.

The 1 Light shows that the CPU is operating in a − 1 address-modification cycle.

The 2 Light shows that the CPU is operating in a + 1 address-modification cycle.

The 3 Light shows that storage is being re-addressed and the CPU is operating in a − 1 address modification cycle.

SUB SCAN

These lights show what portion of a field is being addressed during arithmetic operation and certain other system executions.

U (Units) shows that the units position of the field is being addressed.

B (Body) shows that the body of the field (excluding units position of the field) is being addressed.

E (Extension) shows that the extension portion of the field is being addressed.

MQ (Multiplier-Quotient) shows that the multiplier or quotient is being addressed during a multiply or divide operation. It is also used to indicate special conditions during an edit operation.

CYCLE

These lights show the eight types of cycles in which the CPU can operate (A, B, C, D, E, F, I, X).

ARITHMETIC LIGHTS

Carry In shows that the carry latch has been set on.

Carry Out shows that the adder has a carry output.

A Compl (A Complement) shows that channel A data are being complemented.

B Compl (B Complement) shows that channel B data are being complemented.

Status Lights

See Figure 47.

B < A (LOW)

This light shows that the B-field is less than the A-field. A computer reset operation or a power-on reset operation turns the light on. The light remains on until the condition is reset by a program operation.

B = A (EQUAL)

This light shows that the B-field is equal to the A-field. The light remains on until the condition is reset by a stored-program operation, a computer reset operation, or a power-on reset operation.

B > A (HIGH)

This light shows that the B-field is greater than the A-field. The light remains on until the condition is reset by a program operation, a computer reset operation, or a power-on reset operation.

OVERFLOW

This light shows that an arithmetic-overflow condition has been detected. The overflow condition can be detected only during an add or subtract operation, and not during a zero and add, zero and subtract, multiply

CENTRAL PROCESSING UNIT						
I RING		A RING	CLOCK	SCAN	CYCLE	ARITH
OP		1	A	N	A	CARRY IN
1	6	2	B	1	B	
2	7	3	C	2	C	CARRY OUT
3	8	4	D	3	D	
4	9	5	E		E	A COMPL
5	10	6	F	SUB SCAN	F	B COMPL
	11		G	U		
	12		H	B	I	
			J	E	X	
			K	MQ		

Figure 46. CPU Control Indicator Lights

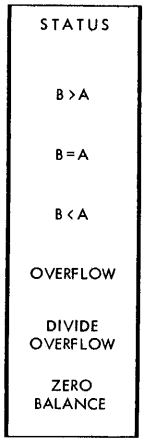


Figure 47. Status Indicator Lights

or divide operation. The light remains on until the condition is reset off by a programmed test operation, a computer reset operation, or a power-on reset operation.

DIVIDE OVERFLOW

This light shows the occurrence of a divide-overflow condition. The light remains on until the condition is reset off by a programmed test operation, a computer reset operation, or a power-on reset operation.

ZERO BALANCE

When on, this light shows the occurrence of a zero-balance condition. It is set by the result (which is zero) of any add, subtract, zero and add, zero and subtract, or multiply operation. The light remains on until the condition is reset by the computer reset, or power-on reset. It is also reset by the result (which is not zero) of any add, subtract, zero and add, zero and subtract, or multiply operation.

I/O Channel Control

There are two sets of I/O channel control lights (Figure 48). One set shows channel 1; the other set shows channel 2, if channel 2 is present. The description applies to both channels, except that the Op code for channel 2 is X instead of R.

INTERLOCK

This light shows that either an I/O read or write operation has been called for. The light is turned off when the status test is satisfied following a read or write operation. The status test is satisfied if either:

1. A branch if any I/O channel status indicator on instruction $\overset{v}{R}$ (I) $\equiv$ is given before encountering the next I/O unit instruction on the same channel, or;

I/O CHANNEL CONTROL	
CH 1	CH 2
INTERLOCK	INTERLOCK
RBC INTERLOCK	RBC INTERLOCK
READ	READ
WRITE	WRITE
OVERLAP IN PROCESS	OVERLAP IN PROCESS
NOT OVERLAP IN PROCESS	NOT OVERLAP IN PROCESS

Figure 48. I/O Channel Control Indicator Lights

2. A specific $\overset{v}{R}$ (I) d instruction (see Figure 19) is given, which results in a branch before encountering the next I/O unit instruction.

If the status test is not satisfied before the next I/O instruction for that particular channel is called for, the system is interlocked and the interlock light remains on.

RBC INTERLOCK (READ BACK CHECK INTERLOCK)

This light shows that the system has completed a successful write operation, but has not called for a read-back check (write-disk check) operation. A write operation must be followed by a read-back check when an IBM 1405 Disk Storage is installed but this is not required for the IBM 1301 Disk Storage.

READ

This light shows that an I/O read operation is in process.

WRITE

This light shows that an I/O write operation is in process.

OVERLAP IN PROCESS

This light is turned on at the beginning of any I/O operation that is performed in the overlap mode. If the system stops because of an error during the I/O operation, the light remains on to indicate what type of I/O operation was in process when the error occurred. When no error occurs, the light turns off at the end of the data transfer.

NOT OVERLAP IN PROCESS

This light turns on at the beginning of any I/O operation that is not performed in the overlap mode. It is turned off at the end of the data transfer. The light signifies what type of I/O operation was in process when the system stopped because of an error.

I/O Channel Status Indicator Lights

The I/O channel status indicator lights (Figure 49) indicate the setting of their associated indicators. The indicators were set as a result of the last I/O operation on that particular I/O unit. Whenever the not ready, busy, data check, condition, wrong-length record, or no transfer I/O channel status indicator is set on, the corresponding indicator light is also turned on. One set of indicator lights is associated with channel 1; another set of identical lights is available for use with the channel 2 special feature. Figures 42 and 43 show the conditions that set the indicators on and turn on their associated lights during a console printer read or write operation. See also Figures 62, 63, 90, 91, and 99 for the specific conditions in other I/O units that will turn on these same indicator lights.

NOT READY

The not-ready light shows that one of the input or output units on that channel is not capable of taking a cycle. Refer to the individual I/O unit write-up for specific conditions that turn on the not-ready light.

BUSY

The busy light shows that one of the input or output units on that channel has not completed a previous operation. Refer to the individual I/O unit write-up for specific conditions that turn on the busy light.

DATA CHECK

The data check light, when on, shows that one of the input or output units on that channel has detected a data parity condition. Refer to the individual I/O unit write-up for specific conditions that turn on the data check light.

CONDITION

The condition light shows that one of the input or output units on that channel has encountered an end-of-file condition or a data-transfer control error condition relating to that unit. Refer to the individual I/O unit write-up for specific conditions that turn on the condition light.

WRONG-LENGTH RECORD

The wrong-length record light shows that one of the input or output units on that channel has encountered or sent a wrong-length record. Refer to the individual I/O unit write-up for specific conditions that turn on the wrong-length record light.

NO TRANSFER

The no-transfer light shows that an operation of one of the input or output units on that channel has resulted in a no-transfer condition. Refer to the individual I/O unit write-up for specific conditions that turn on the no-transfer light.

Systems Check Indicator Lights

See Figure 50.

PROCESS LIGHTS

A-Channel shows that an A-channel parity error has been detected.

B-Channel shows that a B-channel error has been detected.

Assembly Channel indicates an error at the assembly output or an error when merging zones, numeric information, and word marks during any operation.

Address Channel shows that a validity error has been detected on the channel that supplies data to the address registers.

Address Exit is only active during an indexing or store address register operation, and shows that a validity error has been detected at the address register exit channel.

A-Register Set shows that the A-data register has failed to reset.

B-Register Set shows that the B-data register has failed to reset.

Op-Register Set shows that the Op register has failed to set.

Op-Modifier Set shows that the Op-modifier register has failed to set.

A-Character Select shows that no character is, or extra characters are, gated on the A-channel.

B-Character Select shows that a malfunction in the storage-character selection and regeneration circuitries has been detected.

I/O CHANNEL STATUS	
CH 1	CH 2
NOT READY	NOT READY
BUSY	BUSY
DATA CHECK	DATA CHECK
CONDITION	CONDITION
WRONG LENGTH RECORD	WRONG LENGTH RECORD
NO TRANSFER	NO TRANSFER

Figure 49. I/O Channel Status Indicator Lights

SYSTEM CHECK		
PROCESS		PROGRAM
A CHANNEL	A REGISTER SET	I/O INTERLOCK
B CHANNEL	B REGISTER SET	ADDRESS CHECK
ASSEMBLY CHANNEL	OP REGISTER SET	RBC INTERLOCK
ADDRESS CHANNEL	OP MODIFIER SET	INSTRUCTION CHECK
ADDRESS EXIT	A CHARACTER SELECT	
	B CHARACTER SELECT	

Figure 50. System Check Indicator Lights

PROGRAM LIGHTS

I/O Interlock shows that the program has failed to test the I/O channel status indicator before the next I/O instruction on that channel.

Address Check shows that an incorrect storage address has been given by the program or that an operation goes beyond the limits of core storage (see "Addressing").

RBC Interlock (Read-Back Check Interlock) shows that the read-back check (write disk check) operation had not been completed before another operation of that disk storage channel was called for (a requirement for the IBM 1405 Disk Storage only).

Instruction Check shows that an incorrect instruction has been given by the program.

Power Indicator Lights

See Figure 51.

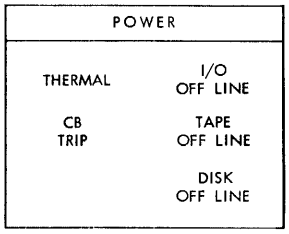


Figure 51. Power Indicator Lights

I/O OFF-LINE

This light shows that:

1. The off-line switch on the 1414 Model 3, 4, 5, or 8 is on.
2. Power is removed from the 1414 Model 3, 4, 5, or 8.

THERMAL

When the internal temperature of the system exceeds the allowable limit or a blower circuit breaker trips, power turns off and the light turns on.

CB TRIP (CIRCUIT-BREAKER TRIP)

When one of the circuit breakers in the system trips, all dc power turns off and the light turns on.

TAPE OFF-LINE

This light shows that:

1. Either one or both tape transmission channels are operating off-line (CE use only).
2. Power is shut down for either one or both tape transmission channels.
3. The stop-on-error switch on the 1414 Model 1, 2, or 7 CE panel is on.

DISK OFF-LINE

This light shows that:

1. The off-line switch on a disk storage CE panel is not in RUN position.
2. Power is shut down for either one or both disk transmission channels.

System Controls Indicator Lights

See Figure 52.

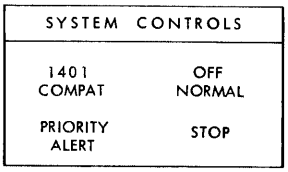


Figure 52. System Controls Indicator Lights

1401 COMPATIBILITY LIGHT

This light shows that the system is in the 1401 mode of operation (capable of running IBM 1401 programs). It is turned on when the compatibility switch is in the 1401 (ON) position.

OFF NORMAL LIGHT

This light shows that certain CE switches on the console are not in the correct position for normal operation. The light is on if:

1. Print-out control switch is set to INHIBITED.
2. Asterisk insert switch is set OFF.
3. Cycle control switch is not set OFF.
4. Check control switch is not set to STOP NORMAL.
5. Storage scan switch is not set OFF and mode switch is at CE.
6. Address entry switch is not set to NORMAL.

STOP

This light, when on, shows that the system has stopped and that operator intervention is required to start a new operation.

PRIORITY ALERT

This light, when on, shows that the system is operating in the priority alert mode and is ready for an interruption (priority feature must be installed).

IBM 1415 Console CE Test Panel

The IBM 1415 Console customer engineering test panel (Figure 53) is provided primarily for customer engineering use in diagnostic testing and performing preventive maintenance routines. However, certain functions of the panel can be used advantageously by customer personnel when checking new program routines. Only the switches and functions of use to the customer are described in this section of the manual.

Console CE Controls Having Customer Uses

CHECK CONTROL SWITCH

The check control switch is a three-position rotary switch. When it is set to STOP NORMAL, any CPU error or input parity error, with the asterisk-insert switch set OFF, results in an immediate stop and an error print-out operation. For normal operation, this switch is set to the STOP NORMAL position, with the asterisk insert switch on.

When the check control switch is set to RESTART, any of the previously mentioned errors also results in an

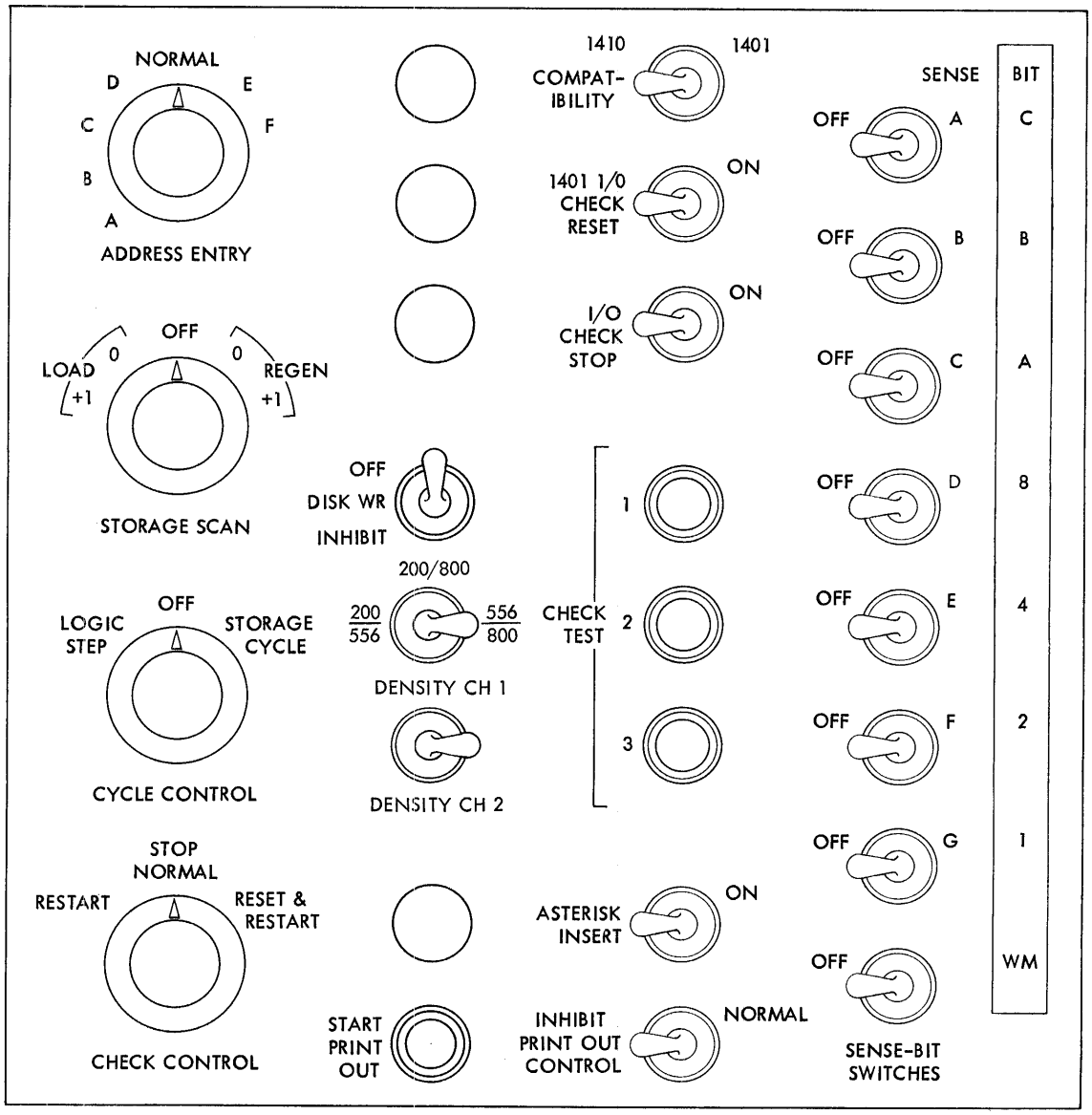


Figure 53. Console CE Test Panel

immediate stop. Following the error print-out operation, the program is restarted automatically. If the error print-out is bypassed (print-out control switch), the program is restarted immediately following the stop.

When the check control switch is set to RESET AND RESTART, any of the previously mentioned errors also results in an immediate stop in the same manner as for the RESTART setting. An error print-out operation is followed by a computer reset operation. When the computer reset operation is completed, the program is restarted. If the error print-out is bypassed (print-out control switch), computer reset and the program start follow the stop.

PRINT-OUT CONTROL SWITCH

This toggle switch controls all stop print-out operations, including error print-out. When this switch is set to NORMAL, the print-out takes place. (See Figure 39.) When the switch is set to INHIBITED, the print-out does not take place.

START PRINT-OUT SWITCH

This switch is used with the print-out control switch. Pressing this switch, with the print-out control switch on NORMAL, results in a stop print-out operation. Printing the contents of the various registers aids in determining the cause of failure. This switch can also be used to initiate occasional print-outs while single cycling.

ASTERISK INSERT SWITCH

This toggle switch, when set to ON, converts any input unit character of incorrect parity to an asterisk, and enters it into storage in place of the invalid character. When the toggle switch is set to OFF, a wrong-parity character from any input unit stops the operation and initiates an error print-out operation, unless inhibited by the print-out control switch. If the asterisk insert switch is OFF and the check control switch is set on STOP NORMAL, the data transfer stops. If the asterisk insert switch is OFF and the check control switch is set on RESTART (and the print-out control switch is set on INHIBITED), the full record can be entered into storage and used for diagnostic purposes or for the reconstruction of the incorrect record.

CYCLE CONTROL SWITCH

This is a rotary three-position switch used with any setting of the mode switch.

When the cycle control switch is set to OFF, system operation is not controlled by this switch.

When the cycle control switch is set to STORAGE CYCLE, pressing the start key advances the program by single storage cycles. A print-out operation, as described in the "I/E CYCLE" mode switch setting, occurs at the end of each cycle, unless inhibited.

When the cycle control switch is set to LOGIC STEP, pressing the start key advances the program by single logic steps.

ADDRESS ENTRY SWITCH

This is a seven-position rotary switch (A, B, C, D, E, F, and NORMAL). This switch enables a console-printed address to enter the selected address register (A, B, C, D, E, F, or IAR if the switch is set to the NORMAL position). To activate this switch, the console mode switch must be positioned to the ADDRESS SET setting.

For normal system operation, the switch must be set to NORMAL.

DISK WRITE SWITCH

The disk-write switch, an optional feature, facilitates testing programs on an IBM 1410 system containing disk storage. It prevents writing test data on permanent records in disk storage. When this switch is set to OFF, normal disk storage operations can be performed.

When the switch is set to INHIBIT, all disk storage instructions, except write disk and write disk with word marks, are performed normally. When these two instructions are encountered, data are transferred from core storage to disk storage and parity and record length are checked; however, no data are written on the surface of the disk. Automatic comparison of the record address in core storage and the address on the disk record is performed, however, and the unequal-address compare indicator turns on if an unequal condition occurs.

When the disk write switch is on INHIBIT, a write operation results in an error condition because no data were written on the disk.

TAPE DENSITY SWITCHES

The tape density switches, one for each channel in use, are an optional feature for selecting the high or low recording density combinations desired for any 729 V or 729 VI tape units on the designated channel.

STORAGE SCAN SWITCH

The REGEN +1 and REGEN 0 settings of the storage scan switch (Figure 53) are used by customer engineers for system testing. However, the LOAD +1 and LOAD 0 settings are of use to the operator; in particular, the LOAD +1 is needed to perform a core clear operation, as follows:

1. Place the mode switch (Figure 44) at CE.
2. Place the storage scan switch at LOAD +1 (see Note).
3. All positions of core storage can easily be loaded with the same character by using the sense-bit switches

(Figure 53). To clear all core storage, position the sense-bit switches to prepare for writing blanks (C bits).

4. Press the start key.

5. At the console-I/O printer, type any valid address. After the fifth character is typed, blanks will be loaded into every position of core storage – starting with the address typed and increasing to the highest address, then wrapping around to the lower addresses.

6. Press the stop key. Because of the speed of core storage, it is not necessary to pause between typing the fifth character and pressing the stop switch for all positions of core storage to be affected.

NOTE: If the storage scan switch is set at LOAD +0, only the address typed in step 5 is cleared or otherwise affected.

SENSE-BIT SWITCHES

The sense-bit switches, used as bit switches, select the character to be loaded into core storage during a CE-mode storage load operation, as described above under "Storage Scan Switch."

The sense-bit switches are active as sense switches (A through G) only when operation is in the 1401 mode. As sense switches, they are tested by the program, and when on, they can cause a branch in the program.

Compatibility Controls

COMPATIBILITY SWITCH

The compatibility switch (Figure 53), when at the 1401 setting, makes it possible to run IBM 1401 programs on the 1410. Ordinarily, the switch should be in the 1410 setting.

I/O CHECK STOP SWITCH

This switch is operative only when the 1410 is operating in 1401 mode. The I/O check stop switch, when set to ON, stops programming at the completion of an I/O operation if the error occurs during that operation. Error conditions that can cause this are: hole count check in the card reader or card punch, validity error in the card reader, print check, or any one of a number of control errors.

The program controls the system in the event of an I/O error when this switch is in the OFF position.

1401 I/O CHECK-RESET SWITCH

This switch is operative only when the 1410 is in 1401 mode and is used with the I/O check stop switch. Operating this momentary switch resets those error conditions than can be bypassed when the I/O check stop switch is set OFF. (The switch is primarily used by customer engineers for diagnostic testing.)

IBM 1414 Input-Output Synchronizer

The IBM 1414 Input-Output Synchronizer (Figure 54) contains the circuitry necessary to transmit data to and from the 1411 Processing Unit and the i/o units indicated in Figure 55. The 1414 is available in many models; the model used with a specific 1410 system depends on the kinds of i/o devices in the system.

Model 1 controls up to ten IBM 729 II, IV, or V Magnetic Tape Units (in any combination) on a single channel (IBM 7330 units can also be controlled if the Tape Intermix Feature is installed). Model 2 of the 1414 controls up to ten IBM 7330 Magnetic Tape Units. Model 7 has the same control capabilities as Model 1 but can also control IBM 729 VI tape units.

Model 3 houses the 80-character card read and punch core buffers and a 100- or 132-character print core buffer, for intermediate storage between the 1402 and 1403 and the 1411 Processing Unit. (Buffering is described in the *1410 Systems Summary*, Form A22-0524.) Model 8 is similar to Model 3 but lacks card read and card punch buffers and controls. Model 4, a two-module unit, includes the 1414 Model 3 features and has additional facilities for telecommunication features, including controls and core buffers for the IBM 1011 Paper Tape Reader, the IBM 1009 Data Transmission Unit, up to 20 IBM 1014 Remote Inquiry Units, and Telegraph Input-Output. Model 5 is similar to Model 4 but lacks card read, card punch, and print buffers and controls.

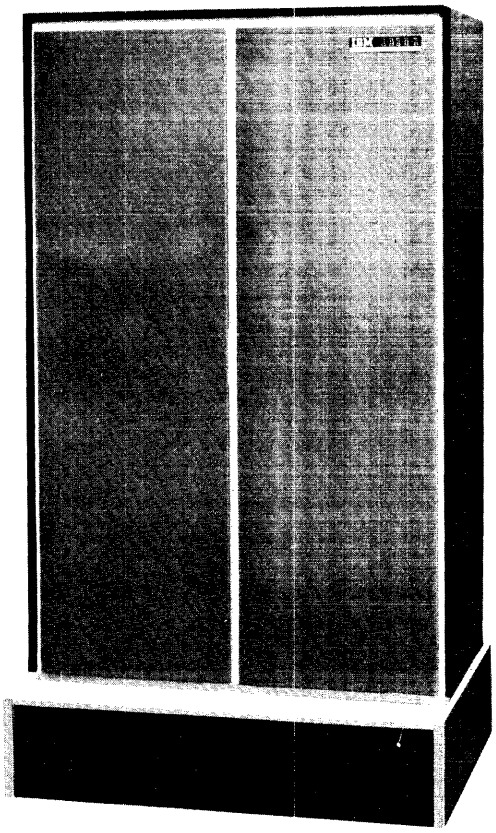


Figure 54. IBM 1414 Input-Output Synchronizer

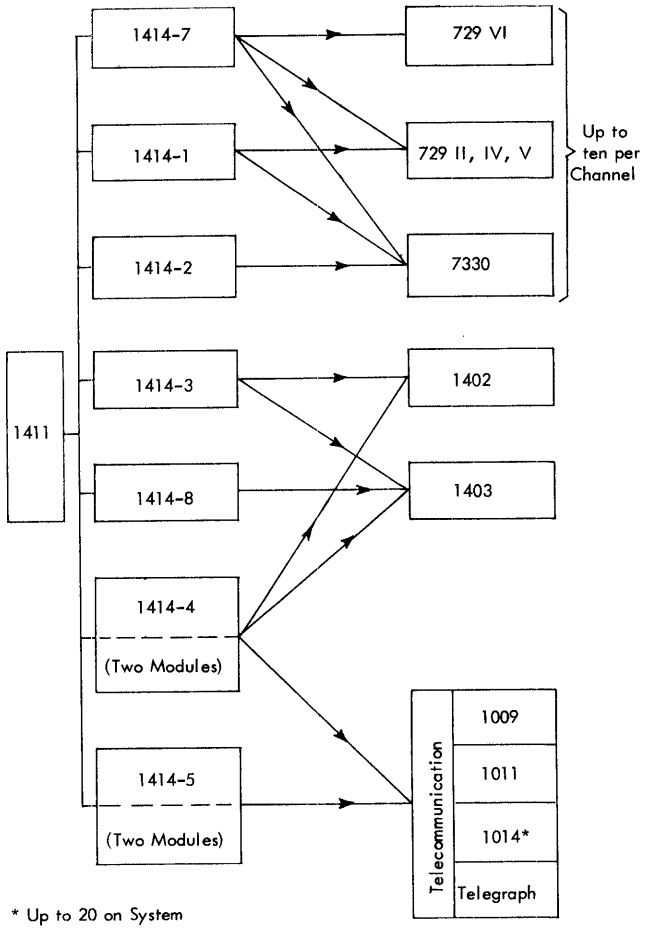


Figure 55. Synchronizer Models Required for Each i/o Device

The tape intermix feature, required to mix IBM 729's and 7330's, and any other features or adapters that are optional or required to complete an installation, are shown in the *IBM 1410 Data Processing System Configuration Bulletin,* Form A22-6688.

Input-Output Instructions

Input-output instructions are commands to specific I/O devices to act as input or output for the 1411 Processing Unit. The form and meaning of an I/O instruction are described under "Instructions" in the first part of this manual and on the *IBM 1410 Reference Card,* Form X22-6502. I/O instructions may be made in the move or load mode; the only difference is in the processing of word marks and word separators.

Move Mode

INPUT

Word separators in incoming data are stored unchanged as word separators in core storage.

OUTPUT

Word separators in core storage are written unchanged as word separators on output devices. No core storage word marks are transferred to the output medium.

Load Mode

INPUT

A single word separator in incoming data is stored in core storage as a word mark over the next incoming character. Two *consecutive* word separators enter core storage as a single word separator. Thus, the input record length is shortened one position in either of these operations: storing one word mark, or storing one word separator.

NOTE: Disk storage units can handle word marks directly in eight-bit mode.

OUTPUT

Core storage word marks are converted to word separators when transferred to output devices. The character with the word mark, as stored in the processing unit, is moved in the output medium to the position following the word separator. A word separator in core storage is converted to two word separators in the output.

Example:

INPUT → CORE STORAGE		
⌒A⌒B⌒⌒C	⌒A⌒B⌒⌒C	(move mode)
	ÅB⌒C	(load mode)
CORE STORAGE → OUTPUT		
ÅB⌒C	AB⌒C	(move mode)
	⌒A⌒B⌒⌒C	(load mode)

⌒ symbolizes a word separator (formerly =).

Checking Execution of I/O Instructions

Six status indicators on the console (Figure 49) automatically reveal, for a specific data channel, two conditions: (1) the inability of an I/O device to execute an I/O instruction, and (2) the detection of data errors during the execution of the I/O instruction. These indicators are common to all I/O devices on that channel. Some I/O devices, such as the 1403 Printer, do not require the use of all six indicators for a full status check. The significance of each status indicator varies with the kind of I/O device and is described in this manual with the given unit.

Certain status indicators are associated with the mechanical state of the I/O unit; the others are associated with data transfer. Status checking occurs twice during the completion of an I/O instruction. The first check determines whether the unit is mechanically capable of executing the instruction. If it is not, the system does not stop; it ignores the instruction and reads out the next sequential instruction. The second check follows the data transfer (for example, from the 1414 to core storage), and determines whether the unit has satisfactorily executed the instruction. The check is for a parity error, a wrong-length record, etc. If one of the status indicators checking data transfer is set, the effects vary with the I/O device.

The operator cannot be sure that the I/O instruction was executed except by programming to test the status indicators.

Example: Checking Execution of a Card Read Instruction

The meaning of the channel status indications can be understood best by examining a specific I/O operation — for example, a card-read operation — for the status checking sequence and the significance of each check.

First, it is necessary to understand the way an image of the first card enters the buffer. This begins with manual operations at the card reader: the operator

places cards in the feed hopper and presses the start key. No I/O instruction is needed yet. Three cards feed into the machine when the start key is pressed, and an image of the first card enters the buffer.

The program can now be executed. When a card-read instruction is encountered, the results should be: (1) the image of the first card transfers from the buffer to core storage, (2) the first card is stacked, and (3) the image of the second card enters the buffer.

Three simultaneous and automatic status checks are made before the instruction is executed:

1. Is the reader ready? If it is not, turn on the not-ready indicator. Conditions causing this indicator to turn on are: a card jam, no cards, 1402 power off, cover open, stacker full, omission of the above manual operations, etc.
2. Is the reader busy? If it is, turn on the busy indicator. The condition causing this indicator to turn on is: the reader is mechanically in motion (buffer is being filled).
3. Is there a "condition"? If there is, turn on the condition indicator. The condition causing this indicator to turn on is: end of file, last card has been processed and stacked (provided the end-of-file key on the 1402 had been pressed).

If any of the three indicators turns on, the I/O instruction is ignored and the system goes on to the next sequential instruction. If none of the three indicators turns on, the card image in the buffer is transferred unconditionally to core storage and three additional, simultaneous, automatic checks are made:

1. Is there an error in the data transferred? If there is, turn on the data check indicator. Conditions causing this indicator to turn on are: a parity error, a timing error, a hole count check, or an invalid card code.
2. Is the record of "wrong length"? If it is, turn on the wrong-length record indicator. The condition causing this indicator to turn on, for a card read, is: a group-mark – word-mark is not correctly located in core storage.
3. Were data transferred under normal or legal circumstances? If not, turn on the no transfer indicator. Conditions causing this indicator to turn on are: (1) a card image has already been transferred from buffer to core storage and the same card image is transferred to core storage again, or (2) two successive cards entered the buffer without a command to read the first into core storage (in this case, the data from the first card would be lost).

Following the I/O operation, and before the next I/O instruction, the condition of all status indicators can be (and usually is) tested by the program to satisfy a system status-check requirement. The alternative to testing all six status indicators is to test at least one status indicator but, in this case, the status check is satisfied only if a branch results from the test.

Role of the 1414

The words displayed on the console by a lighted status indicator depend for meaning on the kind of I/O device addressed by the I/O instruction. The I/O devices operating through the IBM 1414 Input-Output Synchronizer are linked to their common channel status indicators by the 1414.

If the program calls upon a 1414-attached I/O device that is mechanically incapable of executing the instruction, the 1414 is the unit that causes the appropriate I/O channel status indicator lamps to light. If no indicators turn on, the data are transferred. If the data transfer was unsatisfactory, the 1414 turns on the appropriate remaining I/O channel status indicators.

Testing the I/O Channel Status Indicators

Before the next I/O instruction, the on or off condition of the status indicators must be tested by an X- or R-type branch instruction, branch if I/O channel status indicator on. (See Note.) The d-character in the branch instruction selects the indicator or combination of indicators that are tested. An $\check{R}$ (I) ≡ or $\check{X}$ (I) ≡ resulting in a branch shows that at least one indicator is on, because the group mark contains all bits necessary to check all six status indicators. The I-address is usually the beginning of a subroutine that defines the error condition.

Interrogation of the status indicators with an X- or R-type branch instruction does *not* reset the indicators; they are reset only as the next I/O instruction is read out.

NOTE: If the system has the processing overlap feature, the overlap-in-process indicator should first be tested with a test and branch instruction; otherwise, an overlapped operation may become non-overlapped.

Results of Omitting the Status Check

During the reading out of an I/O instruction, an interlock indicator is turned on; if it is still on when the next I/O instruction is read out, the system will stop. This forces the program to check that each I/O instruction is executed as planned. Only two provisions will remove the interlock, and both are program instructions: (1) any of the X- or R-type branch instructions that checks one or more indicators, *if it also results in a branch,* or (2) an $\check{X}$- or $\check{R}$ (I) ≡ instruction, with no requirement of an actual branch.

Summary in Actual Sequence

These steps occur automatically when the system encounters an I/O instruction in the program:

1. Recognize an I/O instruction.
2. Test the interlock indicator. If it is on, the system stops.
3. Reset all six I/O status indicators associated with the instruction.
4. Test the mechanical ability of the addressed I/O device to execute an I/O instruction. If it is not able, set the status indicator showing the reason.
5. Turn on the interlock indicator, blocking further I/O instructions.
6. If any status indicator is on, prevent the execution of the I/O instruction (skip steps 7 and 8) and proceed to the next sequential instruction.
7. Transfer data.
8. Test for errors in the data transfer. If there are any, set the status indicator showing the kind of error (or programmed situation).
9. Proceed to the next sequential instruction. This can never be an I/O instruction (see "Results of Omitting the Status Check").

Units present in every card-oriented 1410 system are an IBM 1411 Processing Unit, an IBM 1414 Input-Output Synchronizer, and the IBM 1415 Console.

Input-output units that can be added to these components to form a card-oriented 1410 system (Figure 56) are the IBM 1402 Card Read Punch, Model 2, and the IBM 1403 Printer (if printed results are desired).

IBM 1402 Card Read Punch, Model 2

The 1402-2 Card Read Punch (Figure 57) is used with the IBM 1410 Data Processing System. The card reader enters data into the system through an 80-position read buffer, and the card punch receives data from the system through an 80-position punch buffer. (These two buffers are located in an associated IBM 1414 Input-Output Synchronizer.)

The card reader has a rated speed of 800 cards per minute (actual card speed realized is governed by the program instructions). The card reader is equipped with a large-capacity, card-loading device, called a file feed. With this device, the read feed can be loaded with as many as 3,000 cards.

Cards pass through the 1402-2 read feed face down, 9-edge first, from right to left, past two sets of reading brushes and a stacker-select station (Figure 58). The read-check brushes read the card to establish a hole-count check. The read brushes also read the entire card for a hole-count check (comparison of the same card as read by the read-check brushes and the read brushes), and direct the data into the read buffer for later transmission to storage.

The card punch has a rated speed of 250 cards per minute (actual card speed realized is governed by the program instructions). Cards pass through the 1402-2 punch feed face down, 12-edge first, from left to right, past a blank station, the punch station, the punch-check brushes, and a stacker-select station (Figure 58). The punch-check brushes read the entire card to establish a hole-count check (comparison of the same card as read by the punch-check brushes against the impulses received by the punch magnets).

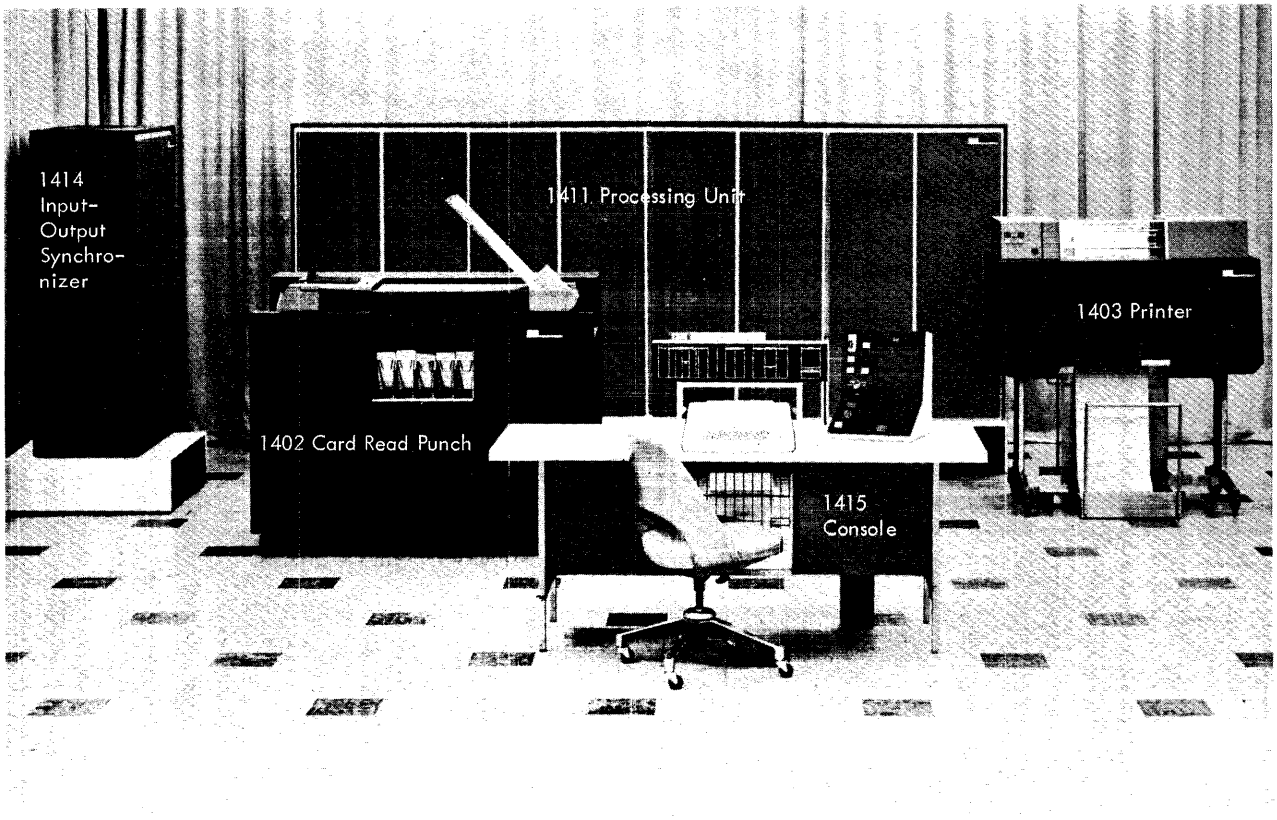


Figure 56. Card-Oriented IBM 1410 System

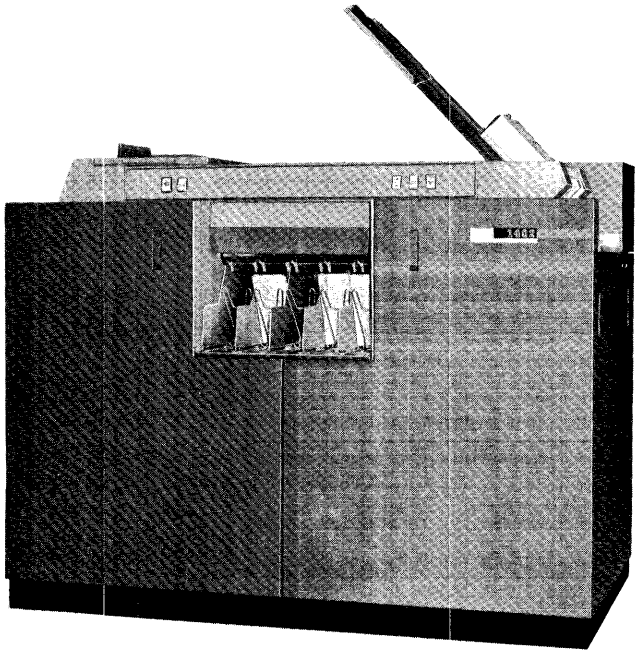


Figure 57. IBM 1402 Model 2 Card Read Punch

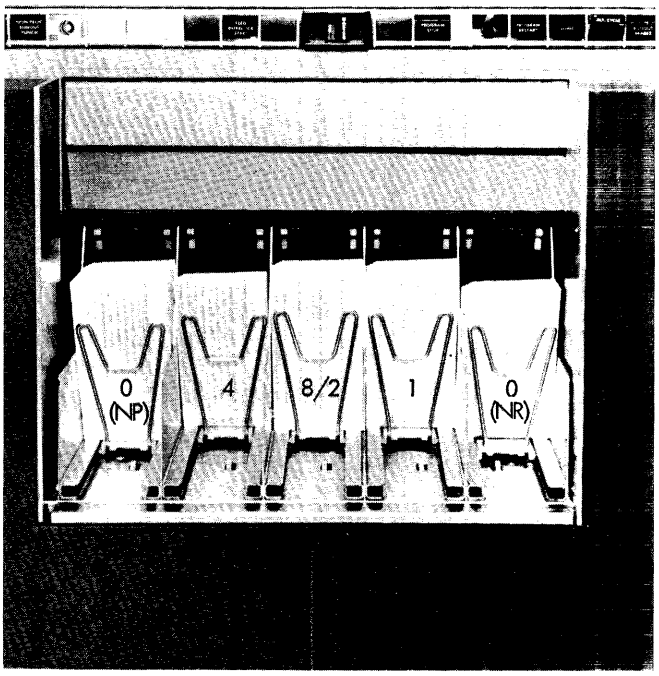


Figure 59. Radial Stackers

The IBM 1402 Card Read Punch, Model 2, is equipped with five radial-type stackers (Figure 59), with a capacity of 1,000 cards each. Cards from each feed can be directed, under program control, to three of the five pockets.

The cards in the card reader can be directed to the 0 (normal read) pocket, the 1 pocket, or the 8/2 pocket. The cards in the card punch can be directed to the 0 (normal punch) pocket, the 4 pocket, or the 8/2 pocket.

NOTE: Cards, in either the punch or reader, which result in validity errors or a hole-count check are automatically stacked in the NP or NR pocket.

Card Read Punch Lights

Several lights on the IBM 1402 Card Read Punch, Model 2, refer to the entire 1402 rather than only to the reader or punch. These lights (Figure 60) are:

Stacker shows that one or more pockets are full. Both the reader and the punch units stop.

Fuse shows that a fuse has blown in the reader or punch unit.

Power shows that power is being supplied to the 1402.

Transport shows that a card jam has occurred in the stacker area. Card feeding is stopped in the rest of the 1402 until the jam is removed.

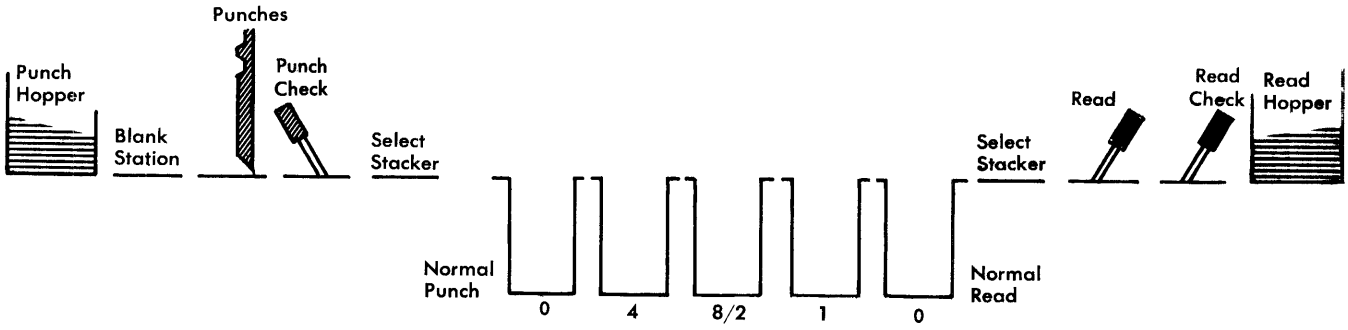


Figure 58. IBM 1402, Card Transport Schematic

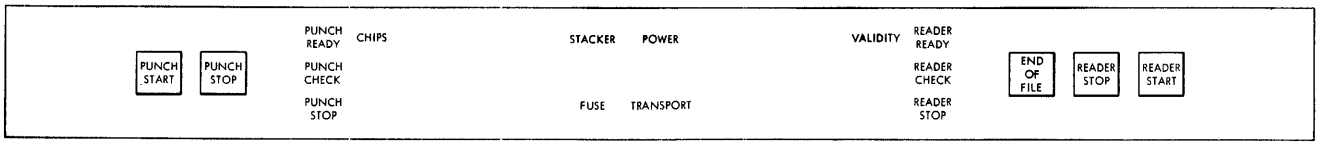


Figure 60. IBM 1402 Card Read Punch Keys, Lights, and Switches

Reader Keys and Lights

Card reader keys and lights (Figure 60) are:

Reader Start: Operating this key feeds three cards into the read feed, fills the reader buffer with the contents of the first card, and turns on the reader-ready light. When the reader has been stopped, pressing the start key turns on the reader-ready light, and allows the cards to continue feeding under program control. When the cards are removed from the read feed hopper and the end-of-file key is not operated, pressing the start key moves the remaining two or three cards, unprocessed, to the stacker area.

Reader Stop: Operating this key stops the reader and turns off the reader-ready light.

End-of-File: Operating this key activates circuits that signal a last-card condition in the central processing unit. The last-card condition can be used by the program to initiate an end-of-file routine. The end-of-file latch is turned on following the data transfer of the last card. The next card-read instruction is interpreted as a NO OP.

The end-of-file key, which can be pressed at any time, causes the card reader to operate in one of these ways:

1. With four or more cards in the read hopper, all cards are processed and run into a stacker. Operating the stop key or processing the last card causes the end-of-file condition to be reset.
2. With three cards remaining in the feed, a card read or card-feed instruction before operation of the end-of-file key causes the program to set the not-ready I/O channel status indicator. Pressing the end-of-file key and then the start key allows the last three cards to be processed and run into a stacker. Operating the stop key or processing the last card causes the end-of-file condition to be reset.
3. With the one, two, or three cards to be processed in the read hopper, pressing the end-of-file key and then the start key feeds the card or cards and turns on the reader-ready light after the first card passes the second read station. The card or cards are processed and run into a stacker. Operating the stop key or processing the last card causes the end-of-file condition to be reset.

Reader Ready (light) shows that the reader is under program control.

Validity (light) shows that an invalid character has been detected during a feed operation. The light remains on until the next feed instruction is started. During the read instruction, the invalid character is transferred from buffer to storage.

Reader Stop (light) shows a feed failure or card jam during a feed operation. This error stops the reader and turns off the reader-ready light.

Reader Check (light) shows the detection of a hole-count error, parity error, or buffer-timing error during a feed operation. The light remains on until the next feed instruction is started. During the read instruction, the data are transferred from buffer to storage, and the CPU sets the data check I/O channel status indicator on and the program can test it.

Punch Unit Keys and Lights

Card punch keys and lights (Figure 60) are:

Punch Start: Operating this key feeds two cards into the punch feed and turns on the punch-ready light. When the punch has been stopped, pressing the start key turns on the punch-ready light, and allows card punching to resume under program control. When the cards have been removed from the punch feed hopper, pressing the start key moves the three cards remaining in the punch feed to the normal-punch pocket. The first card that enters the normal-punch pocket is unchecked.

Punch Stop: Operating this key stops the punch and turns off the punch-ready light.

Punch Ready (light) shows that the punch is under program control.

Punch Stop (light) indicates a feed failure or card jam during a punch operation. This error stops the punch and turns off the punch-ready light.

Punch Check (light) shows the detection of a hole-count error, parity error, or buffer timing error during a punch operation.

Chips: This light shows that the chip receptacle is full or not in place.

IBM 1402 Card Read Punch Operation Codes

See Figure 61 for IBM 1402 operation codes.

Read a Card

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
R or R1 (Ch 1)	v M	%1x³	bbbbb	R
R2 (Ch 2)	v M	□1x³	bbbbb	R
RW or R1W (Ch 1)	v L	%1x³	bbbbb	R
R2W (Ch 2)	v L	□1x³	bbbbb	R

Function: This instruction causes the transfer of 80 characters from the card-read buffer to core storage. If the units position (x³) in the x-control field is 0, 1, or 2, a card feed is initiated and the card from which this data came (which is now at the select station) is directed to the 0 (NR), 1, or 2 (8/2) pocket. During the feed cycle the card-read buffer is refilled and this card is positioned at the select station. If the units position in the x-control field is 9, the contents of the card-read buffer are transferred to core storage, but there is no card feed and stacker-select operation.

The B-address specifies the high-order (starting) position of the data record in core storage. Data records are transferred from high-order to low-order position (left to right). The operation is stopped by the first group-mark – word-mark sensed in core storage.

Word Marks: A group-mark – word-mark must appear in the core-storage position to the immediate right of the data record. If the v L Op code is used, word separators are read into storage as word marks. The character in the adjacent card column is also stored in the core storage position that contains the word mark. This combination of word marks and their associated characters affects the record length.

The programmer must determine in advance and set the correct field length for each card to be processed. If the v ≠ is incorrectly positioned, the wrong-length record channel status indicator will be turned on.

Timing: T = 49.5 + I/O.

(See "IBM 1402, Model 2, Timing Considerations.")

Note: A branch if any I/O channel status indicator on instruction, v R (I) ≠, must be given before the next

I/O UNIT	OP CODE	X-CTRL FIELD	DESCRIPTION	MNEMONIC	d-CHARACTER		OPERATION	NOTES
					CHARACTER	CONTROL		
CARD READER	v M or v L	%10	Read a card	R or R1 (Ch 1), R2 (Ch 2) or RW or R1W (Ch 1), R2W (Ch 2)	R	Read	Initiate feed cycle. Transfer 80 characters from read buffer to core storage. Read card into read buffer. Stack card in pocket 0.	If v L op code is used, word separators are read into storage as word marks, and each is associated with the following character.
		%11					Same as above, except card is stacked in pocket 1.	
		%12					Same as above, except card is stacked in pocket 2.	
		%19					Transfer 80 characters from read buffer to core storage. THERE IS NO CARD FEED AND STACKER SELECT OPERATION.	
CARD READER	v K (Ch 1) v 4 (Ch 2)	None	Select stacker and feed	SSF or SSF1 (Ch 1), SSF2 (Ch 2)	0 1 2	Select stacker 0 Select stacker 1 Select stacker 2	Initiate feed cycle. Read card into read buffer. Stack card in the designated pocket.	Should be used only after a Read-a-Card instruction having a 9 in the units position of the x-control field.
CARD PUNCH	v M or v L	%40	Punch a card	P or P1 (Ch 1), P2 (Ch 2) or PW or P1W (Ch 1), P2W (Ch 2)	W	Write	Transfer 80 characters from core storage to punch buffer. Punch a card. Stack card in pocket 0.	If v L op code is used, word marks are translated to word separators and each is punched ahead of its associated character.
		%44					Same as above, except card is stacked in pocket 4.	
		%48					Same as above, except card is stacked in pocket 8.	

● Figure 61. IBM 1402 Card Read Punch Op Codes

channel 1 I/O unit operation to be sure that the data were correctly transferred.

(A specific R̆ (I) d instruction may be substituted for the R̆ (I) ≠ instruction. This will not cause a system interlock as long as a branch to the I-address, specified in the R̆ (I) d instruction, occurs.) See Figure 62.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

Select Stacker and Feed

Instruction Form:

Mnemonic	Op Code	d-character
SSF or SSF1	K̆ (Ch 1)	0, 1, or 2
SSF2	4̆ (Ch 2)	0, 1, or 2

Function: This instruction (used after a read a card instruction that had a 9 in the units position of the x-control field) stacks the card that was read on the last card-read cycle into pocket 0 (NR), 1, or 2 (8/2),

INDICATOR	d-CHARACTER	CONDITION
Not Ready	1	Card jam Reader out of cards (not EOF) Reader not on line Reader power off Reader stacker full Cover interlock open Feed clutch failure (clutch chk) Joggle switch open (file feed door) Input/Output Synchronizer off line Input/Output Synchronizer power off
Busy	2	Read buffer being filled Card being stacked
Data Check	4	Hole count check Input/Output Synchronizer detects parity error Input/Output Synchronizer detects timing error Processing Unit detects parity error *Never set on Select Stacker and Feed Instruction*
Condition	8	EOF (last card has been stacked) (EOF latch turned off as this indicator turned on) *Never set on Select Stacker and Feed Instruction*
Wrong Length Record	—(B-bit)	Wrong length record *Never set on Select Stacker and Feed Instruction*
No Transfer	&(A-bit)	Card has been transferred previously. This indicator will be set ON if two Select Stacker and Feed Instructions are given without an intervening Read a Card Instruction with a 9 in the units position (x^3) of the X-control field. It will also be set ON if two Read a Card Instructions with a 9 in the units position of the X-control field are given without an intervening Select Stacker and Feed Instruction.

● Figure 62. I/O Channel Status Indicators Set During Card Reader Operations

depending on the d-character of 0, 1, or 2. A card feed is initiated that refills the card-read buffer and positions this card at the select station.

Word Marks: Word marks are not affected.

Timing: T = 13.5 + I/O.

(See "IBM 1402, Model 2, Timing Considerations.")

Note: A branch if any I/O channel status indicator on instruction, R̆ (I) ≠, must be given before the next channel 1 I/O unit operation to be sure that the data were correctly transferred.

(A specific R̆ (I) d instruction may be substituted for the R̆ (I) ≠ instruction. This will not cause a system interlock as long as a branch to the I-address, specified in the R̆ (I) d instruction, occurs.) See Figure 62.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

Punch a Card

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
P or P1 (Ch 1)	M̆	%4x^3	bbbbb	W
P2 (Ch 2)	M̆	□4x^3	bbbbb	W
PW or P1W (Ch 1)	L̆	%4x^3	bbbbb	W
P2W (Ch 2)	L̆	□4x^3	bbbbb	W

Function: This instruction causes the transfer of 80 characters in core storage to the punch buffer. The units position of the x-control field (x^3) can contain 0, 4, or 8, which directs the card to the 0 (NP), 4, or 8 (8/2) pocket.

The B-address specifies the high-order (starting) position of the data record in core storage. Data records are transferred from the high-order to the low-order position (left to right). The operation is stopped by the first group-mark — word-mark sensed in core storage. At the end of the data transfer, the punch is started and punches a card with the data just transferred.

Word Marks: A group-mark — word-mark must appear in the core storage position to the immediate right of the data record. If the L Op code is used, word marks are translated to word separators for punching. Use of the load mode causes the word separator to be punched ahead of its associated character and affects the resultant field length.

The programmer must determine in advance and set the correct field length for each card to be processed. If the ≠̆ is incorrectly positioned, the wrong length record channel status indicator will be turned on.

Timing: T = 49.5 + I/O.

(See "IBM 1402, Model 2, Timing Considerations.")

Note: A branch if any I/O channel status indicator on instruction, R̆ (I) ≠, must be given before the next

channel 1 I/O unit operation to be sure that the data were correctly transferred.

(A specific $\check{R}$ (I) d instruction may be substituted for the $\check{R}$ (I) ≠ instruction. This will not cause a system interlock as long as a branch to the I-address, specified in the $\check{R}$ (I) d instruction, occurs.) See Figure 63.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

INDICATOR	d-CHARACTER	CONDITION
Not Ready	1	Card jam Punch out of cards Punch stacker full Punch power off Punch not on line Chip basket full or not in place Cover interlock open
Busy	2	Previous card still being punched
Data Check	4	Input/Output Synchronizer detects parity error (card not punched)
Condition	8	Parity error detected during punching or hole-count check. Error card goes to 0 pocket.
Wrong Length Record	—(B-bit)	Wrong length record (this card not punched)
No Transfer	⅋ (A-bit)	Never set

● Figure 63. I/O Channel Status Indicators Set During Card Punch Operations

IBM 1402, Model 2, Timing Considerations

The I/O units are "busy" transferring data to or from their input-output buffers for these times:

1. Read a card instruction, when the units position of the x-control field is 0, 1, or 2.

I/O = 0-75,000 μs (access time) + 880 μs (time needed by buffer to accept 80 positions of data at 11 microseconds per position) + 65,000 μs (read cycle).

2. Read a card instruction, when the units position of the x-control field is 9.

I/O = 880 μs (time needed by buffer to accept 80 positions of data at 11 microseconds per position).

3. Select stacker and feed instruction.

I/O = 0-75,000 μs (access time) + 65,000 μs (read cycle).

4. Punch a card instruction.

I/O = 0-60,000 μs (access time) + 880 μs (time needed by buffer to accept 80 positions of data at 11 microseconds per position) + 217,500 μs (punch cycle).

Special Feature — Interchangeable 51-Column Read Feed

The interchangeable 51-column read feed (including file feed) permits feeding either 51-column cards or standard 80-column cards in the read feed of the IBM 1402 Card Read Punch, Model 2. Modifying the read file feed and stackers readily adapts the IBM 1402-2 for processing 51-column cards.

The 51-column card is used for charge sales slips, postal money-order forms, installment payments, inventory cards, and many other applications.

Using an interchangeable feed allows direct entry to the data processing system from the stub card. This eliminates the need for reproducing 51-column cards into standard 80-column cards.

To adapt the read feed for 51-column-card operation, the operator installs a tray and hopper side plates on the read file feed, and adjusts the stackers on the read side.

Normal operations of the IBM 1402 Card Read Punch, Model 2, can be performed with 51-column cards in the read feed. For example, a file of 51-column cards can be processed in the read feed while the results are punched in 80-column cards in the punch feed. However, when the stackers are adjusted to accept 51-column cards, no cards from the read feed can be selected into stacker 8/2.

MODIFYING THE FILE FEED

An adapter tray (Figure 64), placed on the file-feed magazine, accommodates the 51-column cards. A modified card weight enables feeding the last cards from the hopper. Inserting two hopper side plates (Figure 64) positions the 51-column cards at the center of the feed. Thumbscrews fasten the side plates to the hopper. Jogglers align the cards in the hoppers, as in standard operation.

In 51-column-card operation, the first column of the card corresponds to column 15 of an 80-column card, and is therefore read by brush 15; the last column corresponds to column 65 and is read by brush 65. A factor of 14 relates the card column to the reading brush. A switch for regulating the storing of information from a 51-column card is physically located in the 1402-2. It is automatically turned on when the stacker guide is pulled forward for stacking of 51-column cards.

When the switch is ON, the information from a 51-column card is read into positions 15 through 65 of the read buffer. Positions 1-14 and 66-80 of the read buffer are filled with valid blanks.

To check for proper transfer of the data from the read buffer to core storage, a group-mark — word-mark

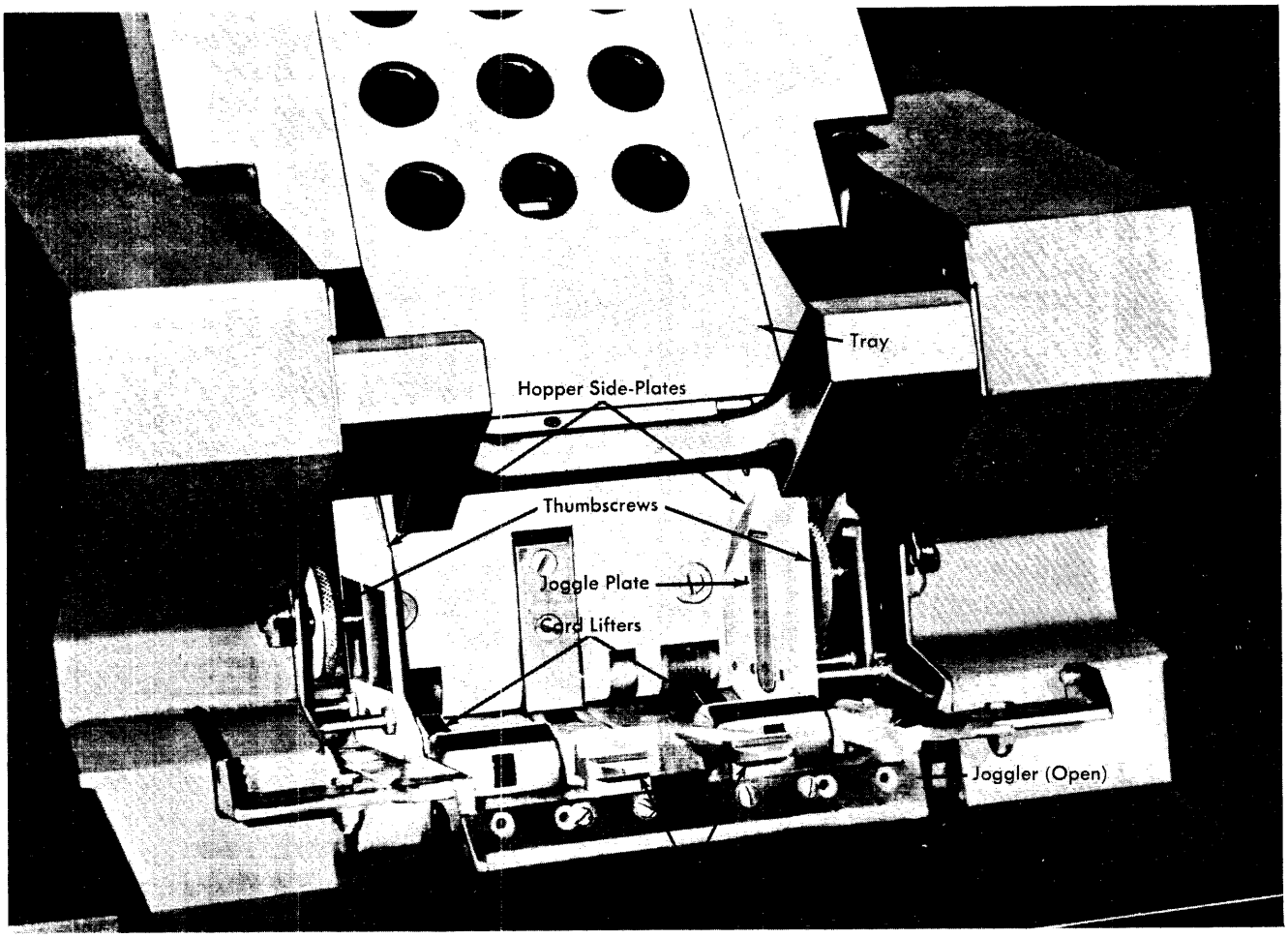


Figure 64. Interchangeable 51-Column Read Feed

must be inserted in the 52nd position of the core storage read-in area. The 51 active positions, but not the blanks, are transferred from the read buffer to core storage.

ADJUSTING THE STACKERS

The operator adjusts the stacker guide (Figures 65 and 66) at the rear of stackers NR and 1 to accommodate 51-column cards. A finger hole permits pulling the guide forward to reduce the depth of the stacker. A spring latch holds the guide securely in either the 51- or 80-column-card position.

A pivot-plate assembly (Figures 65 and 66) adapts the front of stackers NR and 1 for stacking either 51- or 80-column cards. The 51-column pivot plate with card retaining levers swings down and fastens to the stacker separators. This assembly provides a lower pivot for properly stacking the 51-column cards.

For standard 80-column operation, the operator pulls each auxiliary pivot-plate assembly forward and then places it under the cover.

Modified card-deck supports (Figures 65 and 66) for stackers NR and 1 permit stacking 51-column cards, standard cards, and the scored cards processed by the machine. The capacity of each of these stackers is 800 cards.

SETUP OPERATION

To set up the IBM 1402 Card Read Punch, Model 2, to feed 51-column cards in the read feed:

1. Position the side plates in the hopper, and fasten firmly by turning the knurled thumbscrews. Be careful not to interfere with the card lifters.
2. Place the 51-column-card tray over the file-feed magazine.
3. Reach into stackers NR and 1 and, using the finger hole, pull the guide forward until it latches.
4. Raise the cover over the auxiliary pivot-plate assemblies, lower one assembly partially, and then slide the main pivot-plate to the rear until it latches.

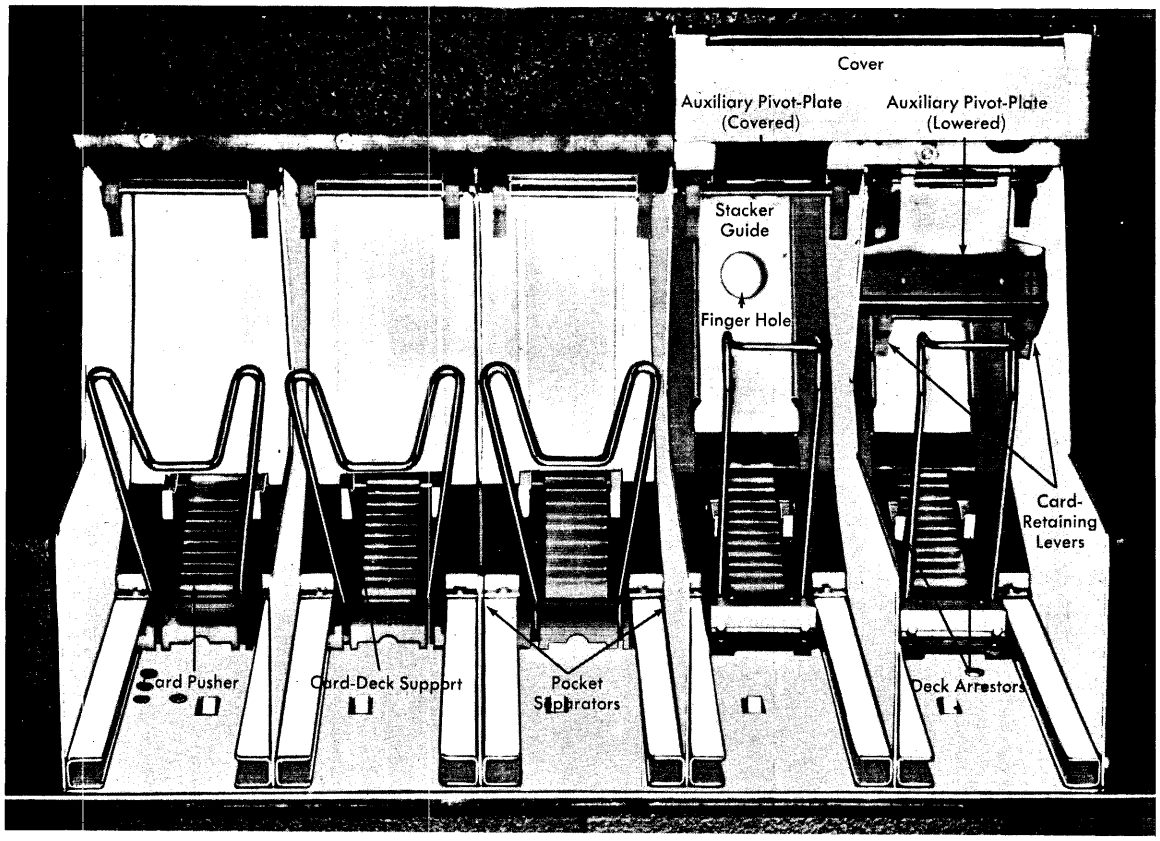


Figure 65. 51-Column Adjustable Stackers

5. Swing the auxiliary pivot-plate assembly down until it latches to the stacker separators. (Repeat steps 4 and 5 for the other pivot-plate assembly.)

Reverse this procedure to return to standard card feeding. (NOTE: Handle and store the adapter tray and hopper side plates carefully to avoid damaging them.)

IBM 1403 Printer

The IBM 1403 Printer (Figure 67) is another output unit for the IBM 1410 Data Processing System. The standard printing line is 100 positions long, with an additional 32 positions available as a special feature. Each position can print 48 different characters: 26 alphabetic, 10 numeric, and 12 different special characters (see Figure 2). Ten characters are printed per inch. For information on the numeric print feature, refer to "Special Features" at the end of this section.

METHOD OF PRINTING

All characters are serially assembled in a closed chain that revolves horizontally (Figure 68). A magnet-driven hammer taps the paper form against the chain

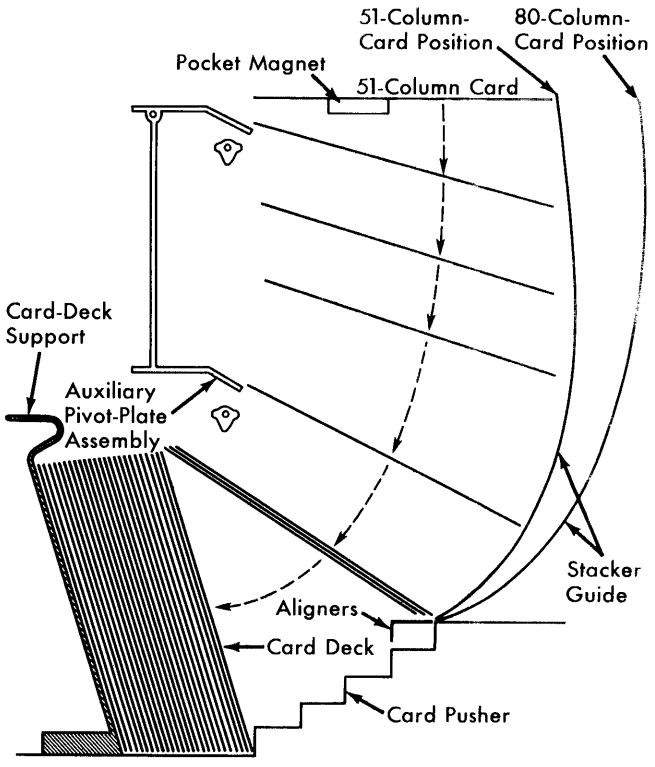


Figure 66. 51-Column Stacker Schematic

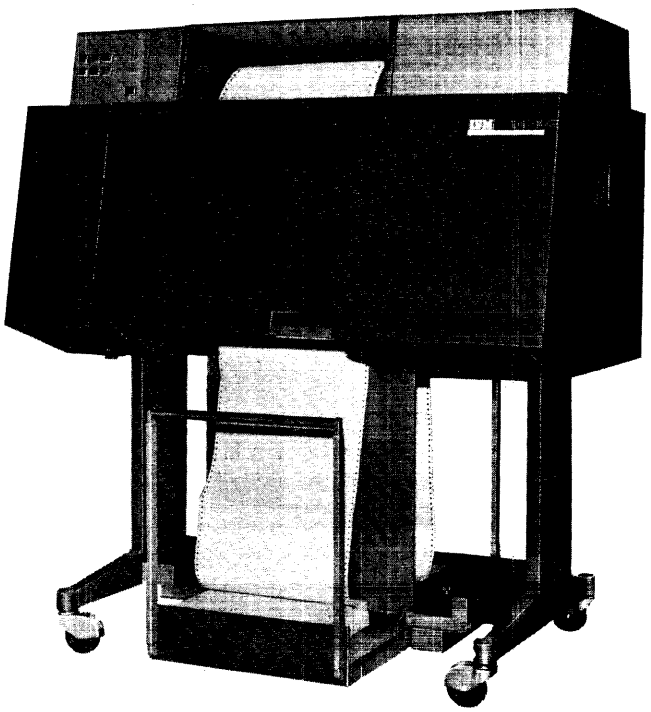


Figure 67. IBM 1403 Printer

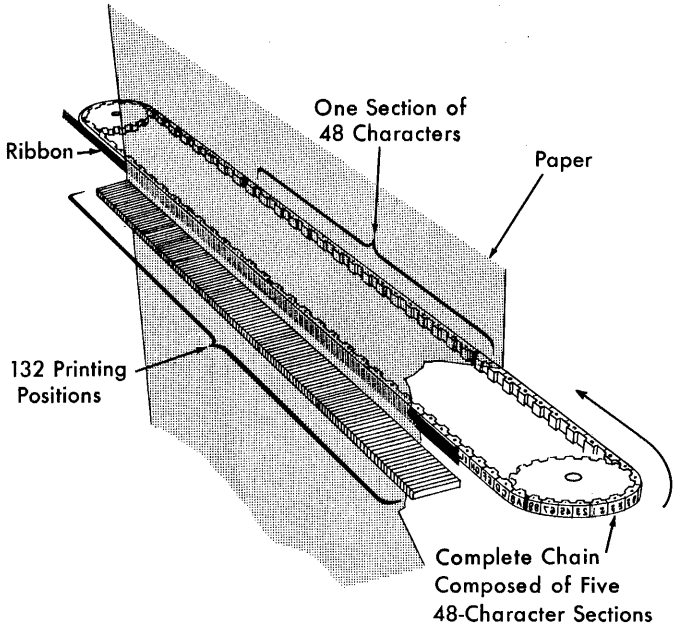


Figure 68. Printer Mechanism, Schematic

as each character to be printed travels past an appropriate print position.

As each character is printed, it is checked against the corresponding position in the print buffer to ensure that printed output is accurate. The machine also checks to ensure that the character is printed in the correct print position, that only valid characters are printed, and that over-printing does not occur.

Printer Keys and Lights

These keys and lights are shown in Figures 69 and 70.

Print Start (Front and Back): Operating this key turns on the ready light.

Print Stop (Front and Back): Operating the stop key turns off the ready light. If the program attempts to execute a print instruction, the program automatically sets on the not-ready I/O channel status indicator in the 1411 and turns on its associated light in the 1415.

Check Reset (key) resets a printer error indication. The print-start key is then pressed to resume operation.

Print Ready (light) shows that the printer is ready to print.

End-of-Forms (light) indicates an end-of-forms condition and the machine stops.

Forms Check (light) shows there is paper-feed trouble in the forms tractor or that the carriage stop has been used. This light must be cleared by the check reset key before the print-start key is effective.

Print Check (light) indicates a print error.

Sync Check (Synchronism Check) (light) comes on to show that the chain was not in synchronism with the printer-compare counter. The timing is automatically corrected. The light is extinguished by operating the printer-start key.

IBM 1403 Carriage Controls

The carriage controls are shown in Figure 69.

Carriage Restore: Pressing this key positions the carriage at channel 1 (home position). If the carriage feed clutch is disengaged, the form does not move. If it is engaged, the form moves in synchronization with the control tape.

Carriage Stop: Pressing this key stops carriage operation and turns on the forms-check light.

Carriage Space: Each time it is pressed, this key causes carriage tape and the form to advance one space.

Single Cycle: This key causes the printer to operate for one print cycle on each pressing of the key when the end-of-form light is on and no paper jam exists. This allows printing of the last line of a form.

IBM 1403 Manual Controls

The manual controls are shown in Figure 71.

Feed Clutch controls the carriage-tape drive and form-feeding mechanism. If it is set to neutral, automatic form-feeding cannot take place. It is also used to select six- or eight-lines-to-the-inch spacing.

Paper-Advance Knob positions the form vertically. It can be used only when the feed clutch is disengaged.

Vertical-Print Adjustment makes possible fine spacing adjustments of forms at the print line. Carriage tape is not affected by this knob.

Lateral-Print Vernier provides fine horizontal positioning.

Print-Density Control Lever: As many as six forms can be printed at one time, and the print hammer unit is designed to adjust automatically for different thicknesses of forms. However, to provide a vernier control for print impression, a print-density control lever is used. When this lever is set at position E, print impression is lightest. When set at position A, print impression is darkest. Between these two settings are intermediate settings. Position C is considered the normal

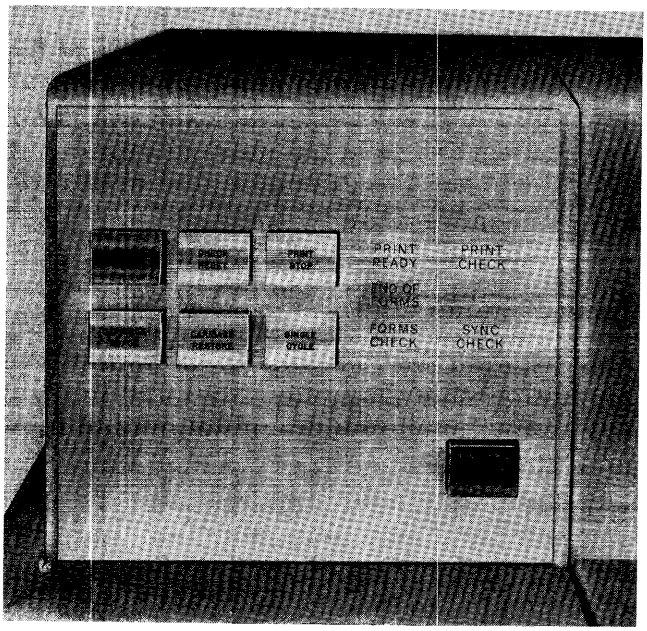


Figure 69. IBM 1403 Printer, Operating Keys and Lights

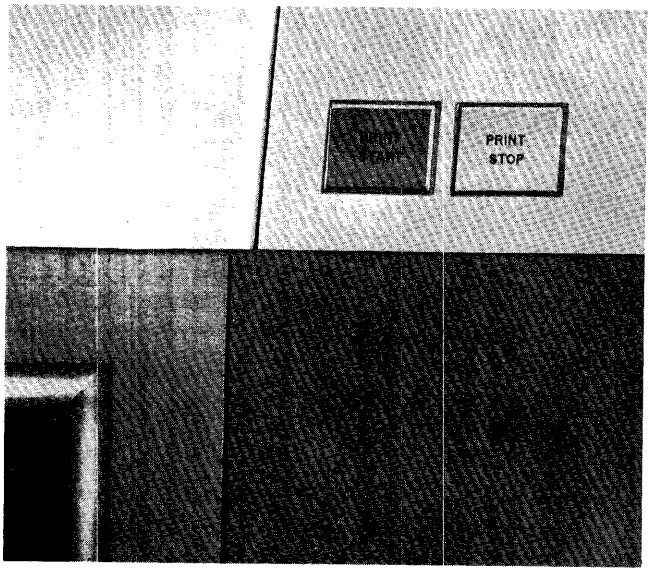


Figure 70. Printer Keys (Rear)

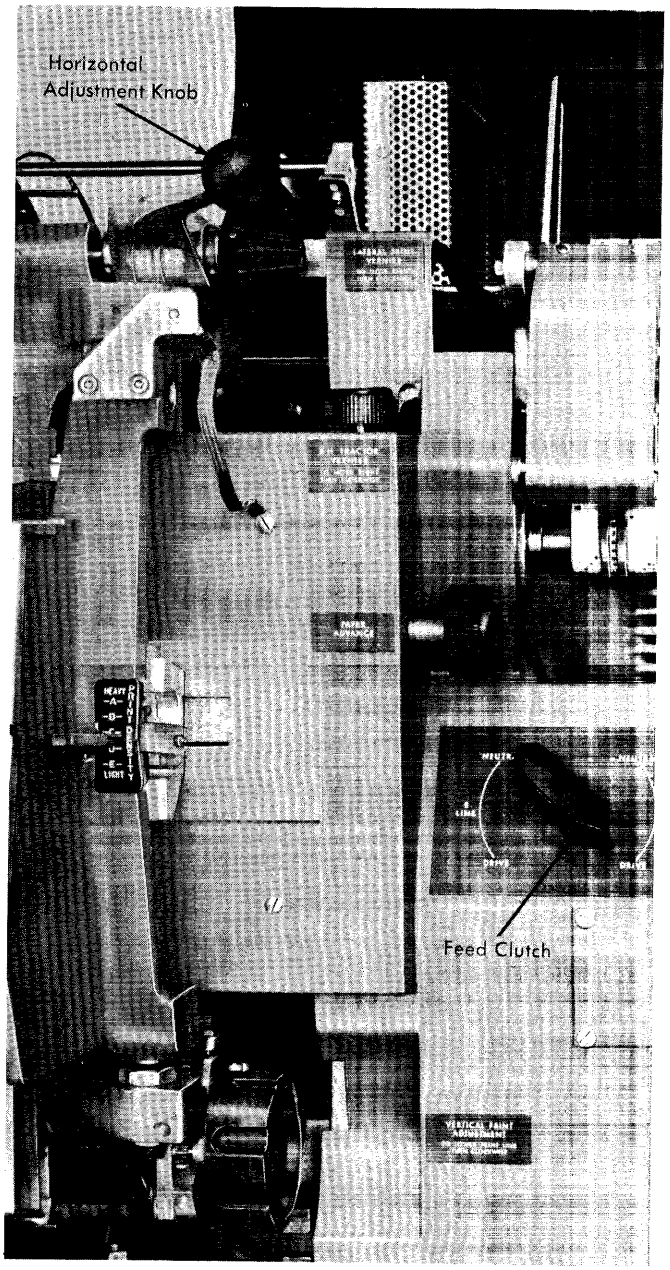


Figure 71. Carriage Controls

setting. This lever moves the type chain closer to or farther from the hammer unit. The setting of this lever must be considered, together with the forms thickness, to determine the normal setting of the print-timing dial (Figure 72). A chart is provided to determine the normal setting (Figure 73).

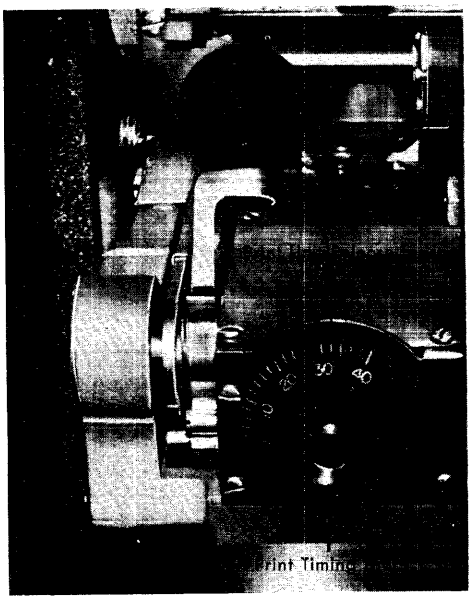


Figure 72. Print-Timing Dial and Print Unit Release Lever

Print-Timing Dial: A movable dial is set to a fixed indicator. Numbers around the dial provide a means of setting the print timing for a specific operation. The setting of the print-density control lever must be set before the print timing dial is set. The nominal setting is read from a chart. The chart (Figure 73) should give the correct setting of the print timing dial. However, this setting can be checked (by rotating the dial slowly in each direction from the normal setting) to determine the limits of good print quality.

Print-Unit Release Lever permits access to the form transport area (see Figure 72).

Print-Line Indicator and Ribbon Shield: The lower ribbon shield is also used as a print-line indicator. It pivots along with the ribbon mechanism. The front side of this shield is marked to show print position location (Figure 74). When used as a print-line indicator, the shield indicates where the lower edge of characters will print. When the printer frame is open, the indicator pivots against the forms so that the print line may be set with respect to the forms.

Horizontal Adjustment: (Figure 71) positions the printing mechanism horizontally. When the lever is raised, the print mechanism unlocks and can be positioned horizontally within its 2.4-inch travel.

PRINT TIMING DIAL SETTING

PRINT DENSITY	FORM THICKNESS .003	.006	.009	.012	.015	.018	.021	.024
A	21	18	15	12	9	6	3	0
B	25	22	19	16	13	10	7	4
C	29	26	23	20	17	14	11	8
D	33	30	27	24	21	18	15	12
E	37	34	31	28	25	22	19	16

OBTAIN DIAL SETTING BY MATCHING "FORM THICKNESS" TO "PRINT DENSITY"

Figure 73. Print-Timing Dial Chart

R. H. Tractor Vernier (Figure 71) allows for fine adjustments in paper tension. It can be used for adjustments of up to one-half inch.

Tractor Slide Bar: Two tractor slide bars, upper and lower (Figure 74), serve as mountings for the forms tractors. The forms tractors are movable, and notches in the tractor slide bar facilitate this movement. A procedure for proper adjustment of these notches, according to the form being used, is given for the upper tractor slide bar. The description would be the same for the lower slide bar.

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

The first notch is used for forms from 5½ to 18¾ inches wide. When this notch is used, the print unit's lateral movement is limited to .4 inch.

The second notch is used for forms from 4½ to 17¾ inches in width. When this notch is used, the print unit's lateral movement is limited to 1.4 inch.

The third notch is used for forms from 3½ to 16¾ inches wide. When this notch or notches 4 through 9 are used, full lateral print unit movement (2.4 inches) is possible.

The ninth (last) notch can be used for forms from 3½ to 10¾ inches wide. When this notch is used, the first usable print position is 38.

The right-hand tractor is locked in place by spring-loaded pins snapped into any one of 27 holes, located one-half inch apart on the tractor slide bar.

The movement of the tractor slide bar, in which the holes are located, is controlled by the right-hand tractor vernier. Movement of up to ½ inch can be made by using the vernier knob.

Indicator Panel Lights

Gate Interlock turns on when the print unit is not locked in position (Figure 75).

Brush Interlock is on if the carriage tape brushes are not latched in position for operation.

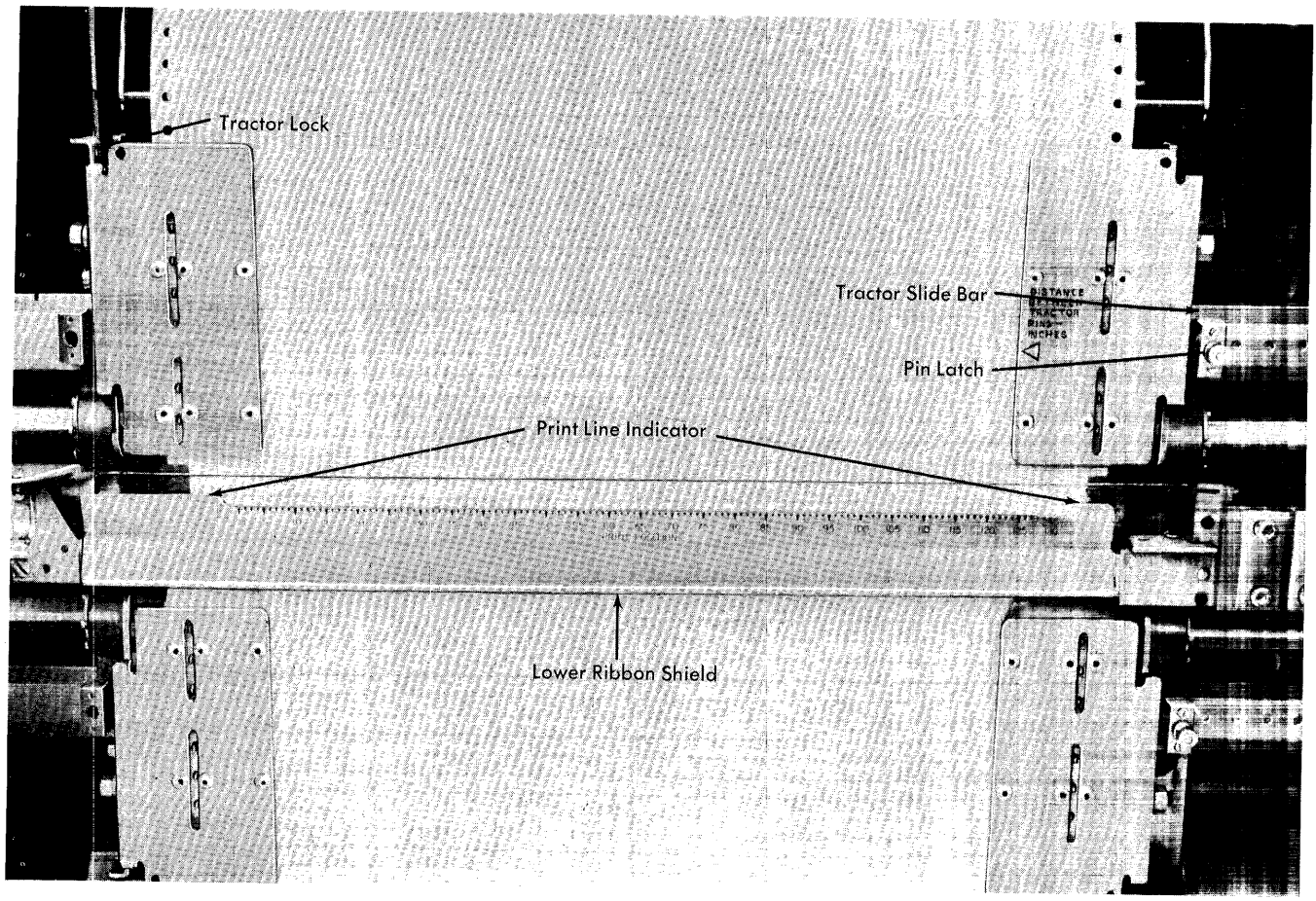


Figure 74. Print-Line Indicator and Ribbon Shield

Shift Interlock turns on to show that the manual feed clutch is not properly positioned.

Thermal Interlock shows that a temperature above the operating limit has been sensed in the hammer unit or chain-drive unit; the light remains on until the temperature drops to an acceptable level. The 1403 is interlocked during this time.

High-Speed Start turns on when a high-speed skip has been initiated.

Low-Speed Start turns on when a low-speed skip or line spacing has been initiated.

High-Speed Stop turns on to show that high-speed skipping is to be stopped.

Low-Speed Stop turns on to show that a low-speed skip stop has been initiated. It is on when the carriage is not in motion.

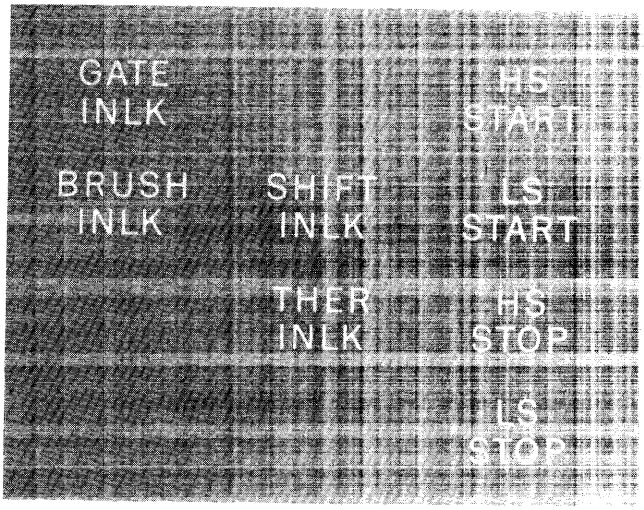


Figure 75. Printer Indicator Panel

Tape-Controlled Carriage

The tape-controlled carriage (Figure 76) controls high-speed feeding and spacing of continuous forms. The carriage is controlled by punched holes in a paper tape that corresponds in length to the length of one or more forms. Holes punched in the tape stop the form when it reaches any predetermined position.

Carriage skip channels 1-12 are standard. The tape circuits initiate special signals that are sent to the CPU when channels 9-12 are sensed. Program testing of carriage channels 9 and 12 is standard.

Vertical spacing and skipping are initiated by the program. Horizontal spacing is 10 characters to the inch. Vertical spacing of either six or eight lines to the inch can be manually selected by the operator.

Forms skip at the rate of 33 inches per second. With the dual-speed carriage, distances of eight lines or less are skipped at 33 inches per second, and those of more than eight lines at 75 inches per second. The last eight spaces skipped in a high-speed skip are skipped at 33 inches per second.

The carriage accommodates continuous forms, to a maximum of 22 inches in length (at 6 lines per inch) or 16½ inches (at 8 lines per inch). The minimum length is 1 inch. For efficient stacking of forms, the recommended maximum forms length is 17 inches. The width of the form can vary from a recommended minimum of 3½ inches to a maximum of 18¾ inches, including punched margins.

Forms can be designed to permit printing in practically any desired arrangement. Skipping to different sections of the form can be controlled by the program and by holes punched in the carriage tape.

CONTROL TAPE

The control tape (see Figure 76) has 12 columnar positions indicated by vertical lines. These positions are called channels. Holes can be punched in each channel throughout the length of the tape. A maximum of 132 lines can be used to control a form, although for convenience, the tape blanks are slightly longer. Horizontal lines are spaced six to the inch for the entire length of the tape. Round holes in the center of the tape are pre-punched for the pin-feed drive that advances the tape in synchronism with the movement of a printed form through the carriage. The effect is exactly the same as though the control holes were punched along the edge of each form.

PUNCHING THE TAPE

A small, compact punch (Figure 77) is provided for punching the tape. The tape is first marked in the channels in which the holes are to be punched. This can be done easily by laying the tape beside the left edge of the form it is to control, with the top line (immediately under the *glue* portion) even with the top edge of the form. A mark is then made in the first channel, on the line that corresponds to the first printing line of the form. Additional marks are made in the appropriate channels for each of the other skip stops, and for the overflow signal required for the form.

The marking for one form should be repeated as many times as the usable length of the tape (22 inches) allows. With the tape thus controlling several forms in one revolution through the sensing mechanism, the life to the tape is increased. Finally, the line corres-

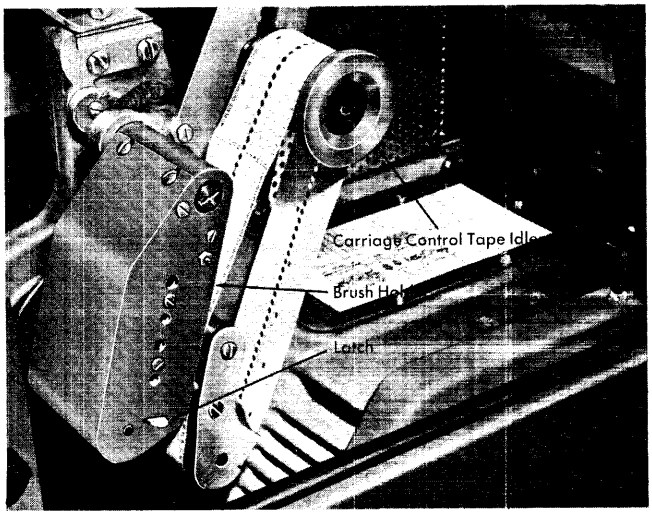


Figure 76. Tape-Controlled Carriage

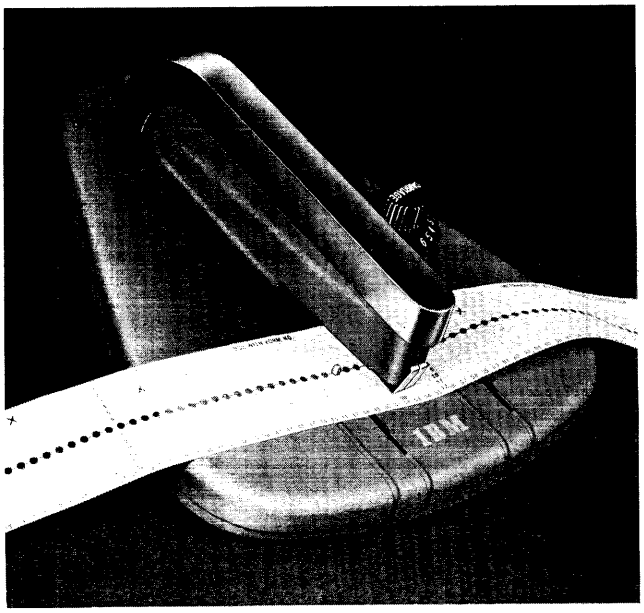


Figure 77. Tape Punch

ponding to the bottom edge of the last form should be marked for cutting after the tape is punched.

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

The last hole punched in the tape should be at least four lines from the cut edge, because approximately the last half inch of the tape overlaps the GLUE section when the two ends are spliced. If it is necessary to punch a hole lower than four lines from the bottom of the form, the tape should be placed with the top line (immediately under the GLUE portion) four lines lower than the top edge of the form, before marking the channels. To compensate for the loss, the tape should then be cut four lines lower than the bottom edge of the form.

8-LINES-PER-INCH SPACING

The control tape for 8-lines-per-inch spacing is punched as it would be for normal 6-lines-per-inch spacing. Each line on the tape always equals one line on the form, regardless of whether the latter be 6 or 8 lines-per-inch. In measuring a control tape for a document printed eight lines to the inch, every ⅛ inch on the form represents one line on the tape.

CARRIAGE TAPE BRUSHES

Two sets of reading brushes (Figure 78), mounted on the same frame, are used to sense holes in the carriage control tape. A small contact roll is used for each set of brushes. One set is called the slow brushes. The other set is called the stop brushes. Seven spaces, as measured by the control tape, separate the brush sets. The slow brushes are positioned ahead of the stop brushes.

The slow brushes are used to control high-speed skipping. They regulate the speed of the last eight spaces of a high-speed skip.

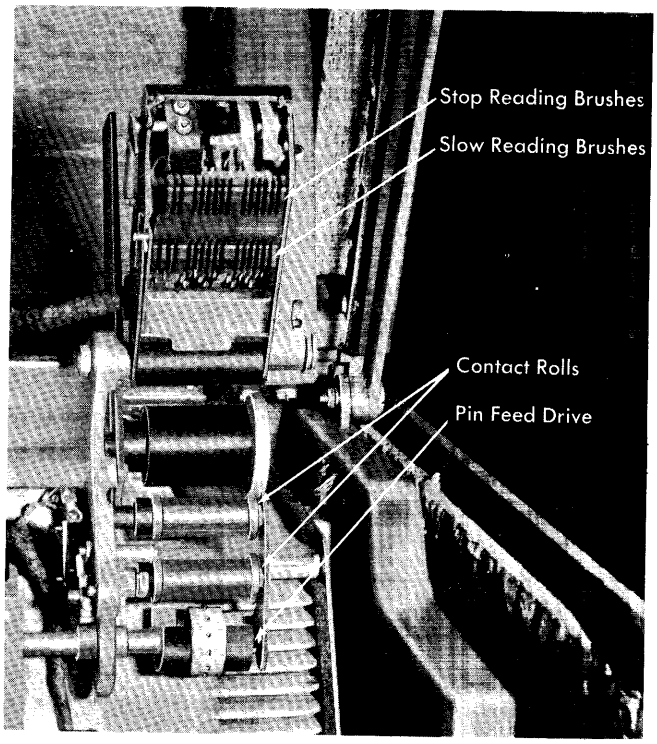


Figure 78. Carriage-Tape Brushes

All carriage tape brushes can function to stop a carriage skip under control of the stored program.

INSERTING CONTROL TAPE IN CARRIAGE

1. Raise the counter-balanced cover of the printer to gain access to the tape-reading mechanism.
2. Turn the feed clutch to a disengaged (neutral) position (see Figure 71).
3. Raise the brushes by moving to the left the latch located on the side of the brush holder.
4. Place one end of the tape loop, held so that the printed captions can be read, over the pin-feed drive wheel so that the pins engage the center drive holes.
5. Place the opposite end of the loop around the adjustable carriage control tape idler.
6. Remove the excess slack from the tape by loosening the locking knob on the idler and moving the idler in its track. Tighten the knob when the desired tension is reached. The tape should be just tight enough so that it gives slightly when the top and bottom portions of the loop are pressed together (see Figure 76). If it fits too tightly, damage occurs to the pin-feed holes.
7. Press the brushes down until they latch, and close the printer cover, when the tape is in position.
8. Press the carriage restore key to bring the tape to its home position, and turn the feed clutch knob back to the engaged position. The carriage is ready to operate.

RIBBON CHANGING

To change the ribbon (Figure 79) on the IBM 1403 Printer:

1. Turn off the power in the printer.
2. Lift up the printer cover.
3. Pull back and unlock the print unit release lever. Swing the print unit out.
4. Open the top ribbon cover.
5. Unlatch the print-line indicator ribbon shield and swing it against the form.
6. Push the top ribbon roll to the right (hinged side of print unit), lift out the left end of the ribbon roll, and remove roll from the drive end of mechanism.
7. Slip ribbon out from under correction roll.
8. To remove the bottom roll, press the ribbon roll to the right, and lower the left end of the ribbon roll and remove it from the drive end of the mechanism.

When replacing the ribbon in the machine, hand-tighten the ribbon to remove slack from in front of the printing mechanism. Ribbons are available in widths of 5, 8, and 11 inches in addition to the standard 14 inches. The ribbon width lever (Figure 80) can adjust the ribbon-feed mechanism to accommodate various ribbon widths.

FORMS INSERTION

1. Raise the counterbalanced cover of the printer to gain access to the print and forms area.
2. Turn the feed clutch knob to a neutral position.
3. Unlock and swing back the print unit by using the print unit release lever.
4. Unlock the paper guide bars by pulling out on the raised handles (upper and lower). On some 1403 Printers, this step is skipped; see step 12.
5. Open the upper and lower forms tractors (Figure 81).
6. Set the left forms tractors slightly to the left of the first unit position by pulling up or down in the tractor lock (upper and lower tractor). See Figure 74.
7. Insert form on pins and close tractor cover.
8. Pull out on right tractor pin and move tractor to desired location to line up the right side of form. The pin should latch in one of the recessions in the tractor slide bars. See Figure 74.
9. Insert form on pins and close tractor covers.
10. Use the tractor vernier knob to tighten the tension on the form. This knob is used for adjustments of up to one-half inch.
11. Check the position and line where printing will occur, by swinging the ribbon shield against the form (it is marked with each print position). If the horizontal alignment is not correct, it can be adjusted by using the horizontal adjustment knob and/or the lateral print vernier knob for slight adjustments. The

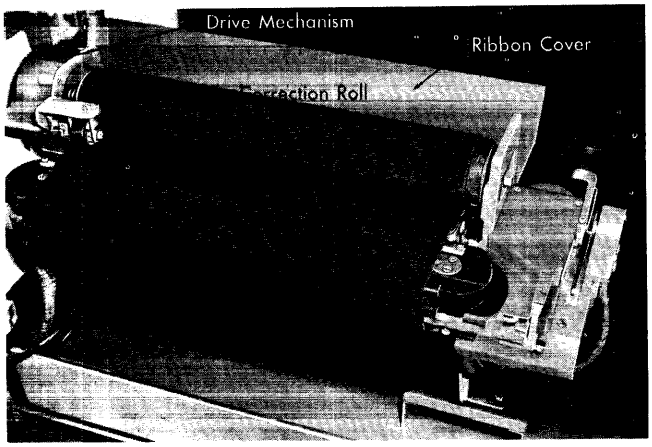


Figure 79. Ribbon Mechanism

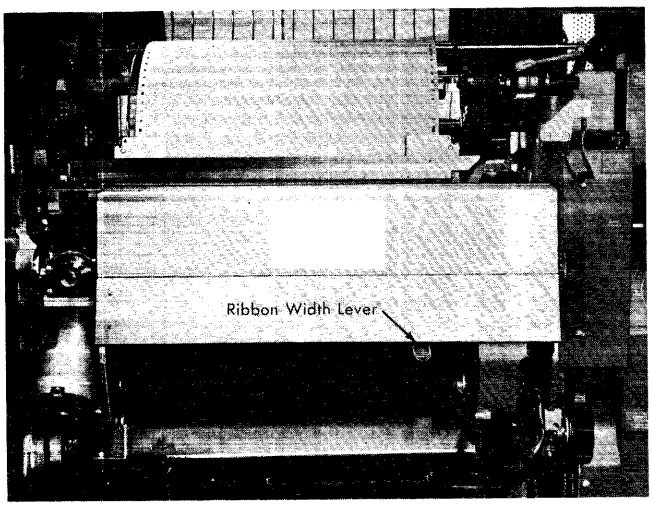


Figure 80. Printer Front Cover, Open

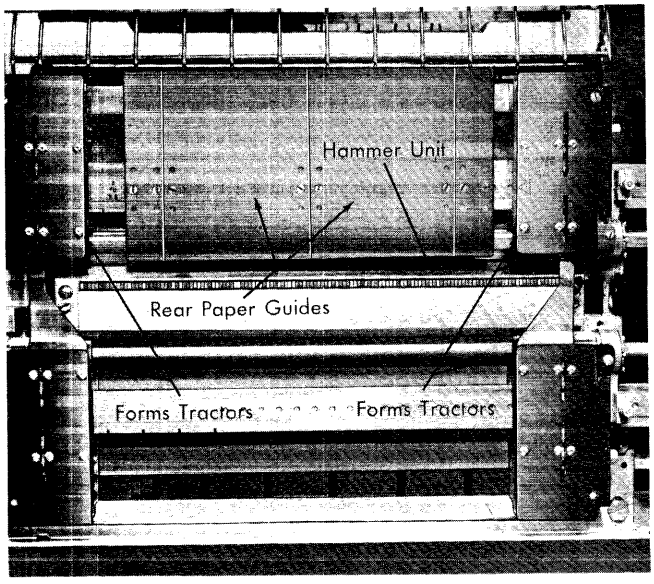


Figure 81. Forms Tractor

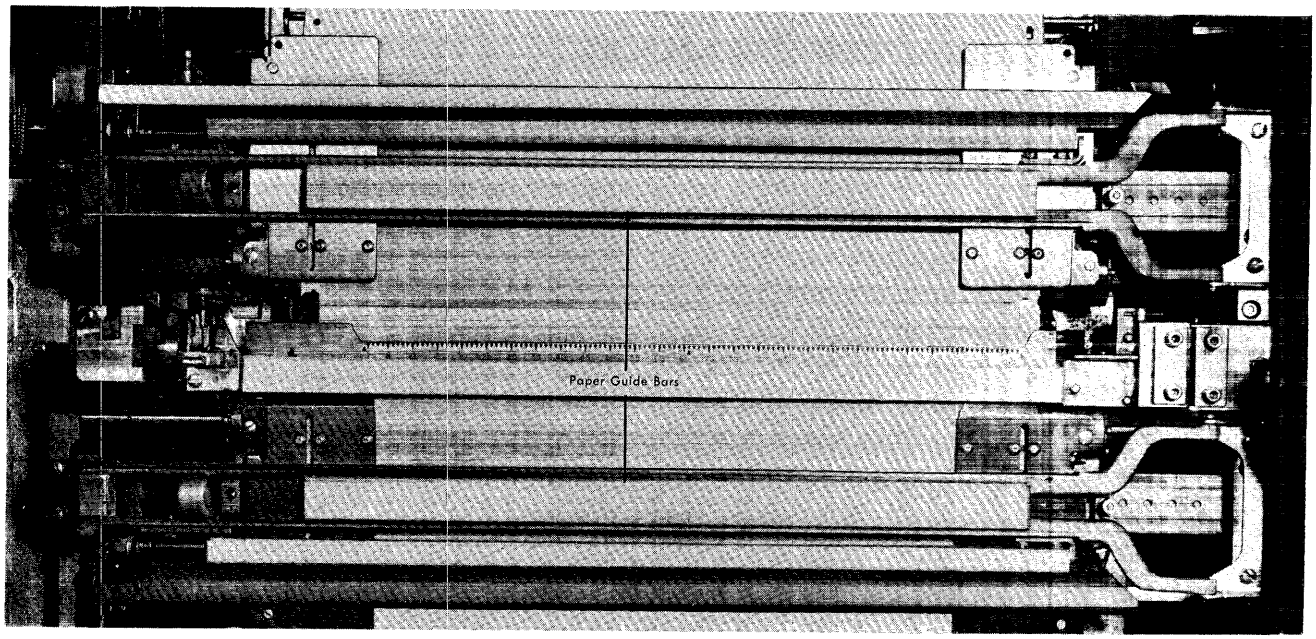


Figure 82. Paper Guide Bars

vertical adjustment can be made by using the paper advance knob and/or vertical print adjustment knob.

12. Return the upper and lower paper guide bars to the closed positions (Figure 82). Some 1403 printers have the tractor-mounted jam detection device which, together with elimination of front "clip on" paper guides, eliminates the need for the upper and lower paper guide bars. The forms insertion procedure for a 1403 with the tractor mounted jam detection device instead of the upper and lower tape guides is the same except that steps 4 and 12 are skipped.

13. Return the print unit to its normal position and lock it in place.

14. Restore the carriage tape to the first printing position by pressing the carriage restore button.

15. Return the feed-clutch knob to a drive position at either six or eight lines-per-inch, depending on the form to be printed.

16. Close the outside cover of the printer.

PAPER STACKER

The paper stacker provides a manual control for optimum stacking of paper at the rear of the printer. Two controls (Figure 83) permit the operator to set up the paper stacker for each individual run.

The upper lever controls the position of the paper guide at the stacker. This lever is indexed (0-6) so that the setup position can be recorded for reference in the operator's procedures.

The lower lever is a speed control that is set to keep light tension on the paper form feeding into the

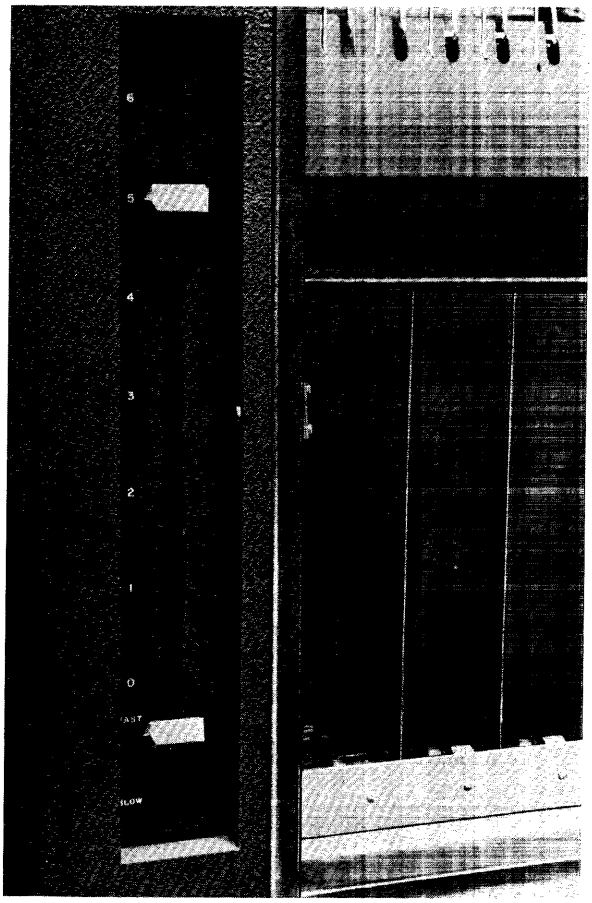


Figure 83. Paper Stacker Controls

stacker. The speed control has five settings. The setting of this control is selected according to the carriage operation being used. For example, if the job is a listing operation with no long skips, the slow position is selected. However, this must also be conditioned by the kind of forms being used because of varying weight of the paper.

Form Design

Some of the customary rules for designing forms should be reconsidered in the light of the many new features introduced by the IBM 1403 Printer.

1. The print unit contains 100 print positions in a 10.0-inch width or a maximum of 132 print positions (special feature) in a 13.2-inch width. Each print position can print any character.

2. Editing, high-speed skipping and other features are included in the system.

One of the basic tools used in designing forms is the spacing chart shown in Figure 84. The numbers across the top from 0 to 13 represent the tens and hundreds positions of the print-position number, and the numbers directly beneath represent the units position of the print-position number. Print-position 42 can be located by referring first to the 4 column and then to the digit 2 within the 4 column. Print-position 9 can be located by referring to the 0 column and then to the digit 9 within that column.

A facsimile of the carriage-control tape is shown at the left (in Figure 84) for marking the control punching for a specific form. Notations have been included relative to standard form-widths and form-depths, lateral movement of the carriage, and instructions to forms manufacturers.

The IBM 1403 Printer carriage is designed to feed marginally-punched continuous forms satisfactorily under the conditions and specifications outlined in Figure 85. These specifications, if followed, give maximum operating efficiency when the 1403 carriage is used. They are not intended to be restrictive but are intended to permit customers to purchase their continuous forms from the manufacturer of their choice.

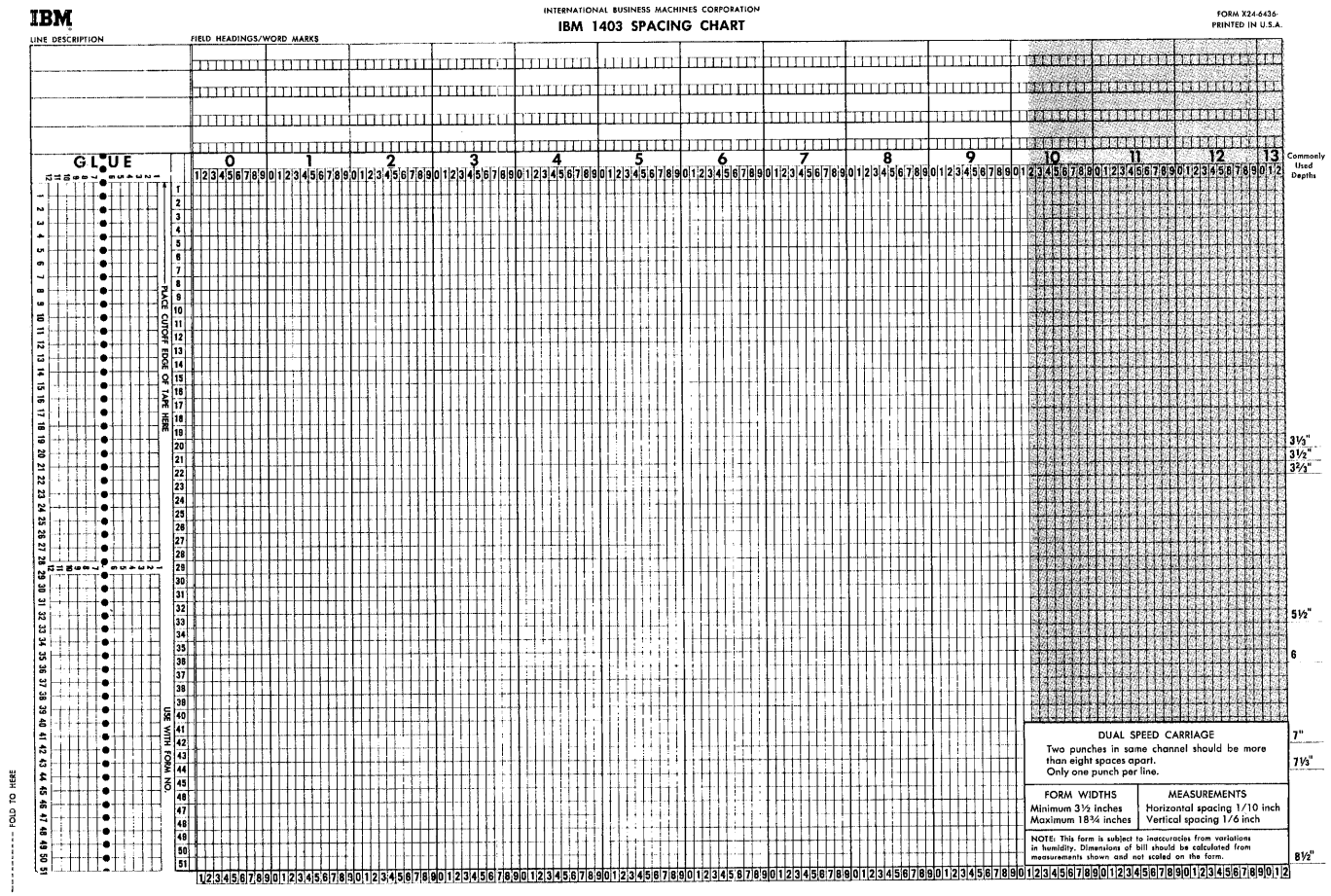


Figure 84. Forms Spacing Chart

FORM DESIGN AS AFFECTED BY THE PRINT UNIT

In view of the 100 or 132 print positions and the 13.2-inch print unit, these factors should be considered when designing forms to be used on the IBM 1403 Printer:

1. The maximum form width is 18¾ inches, and the minimum is 3½ inches (see Figure 85).

2. The maximum form length is 22 inches at six-lines-per-inch spacing, or 16½ inches at 8 lines per inch. For efficient stacking of forms, the recommended maximum forms length is 17 inches.

3. Because all print positions can print all characters, form depth can be reduced, and carbon paper eliminated, by the use of side-by-side printing. For

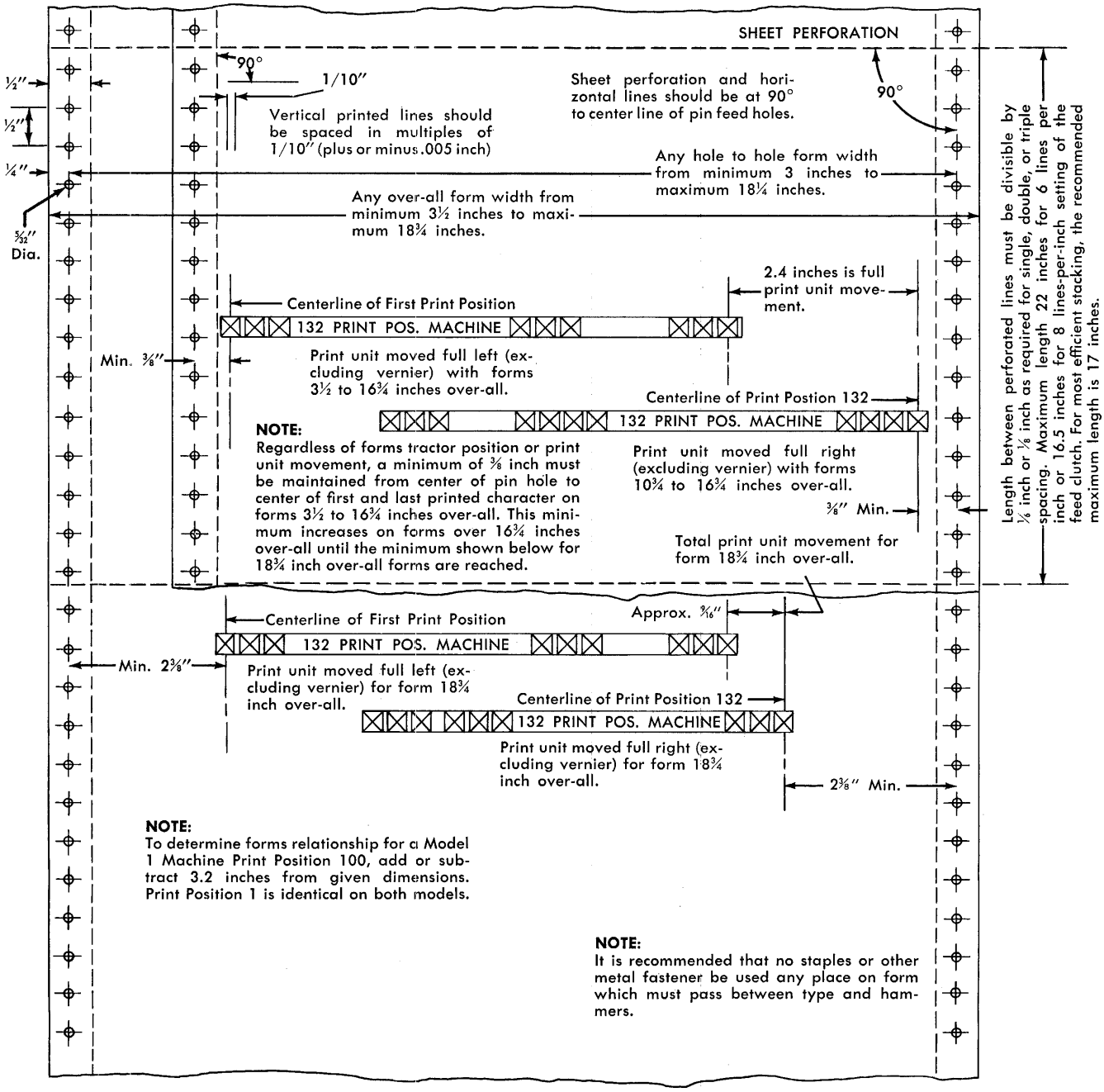


Figure 85. Form Specifications

example, *sold to* and *ship to* names can be printed on the same line, one on the left side of the form and the other on the right.

4. Forms can be designed for printing six or eight lines to the inch. Single-space, eight-lines-per-inch printing is not recommended when the registration between lines is critical.

5. Forms can be designed for variable line spacing within a form by use of single-, double-, or selective-space control.

6. It is possible to dispense with many vertical lines, because the system can be programmed to print commas, decimals, slashes, dashes, and other symbols.

7. A vertical line should not be printed between two adjacent printing positions because there is an over-all maximum tolerance of only .013 inch between adjacent characters.

8. The number of legible copies that can be produced depends on the weight of the paper used for each form, and on the carbon coating. Because the striking force of the print hammers is not adjustable, paper and carbon should be tested in conjunction with the print-density control lever and the print timing dial.

9. The CR (credit symbol) prints from two print positions and the minus sign prints from one. For this reason the minus sign is recommended as a credit symbol intead of the CR symbol.

10. The dollar symbol does not have to be preprinted on a check form, because this symbol can be programmed to print immediately to the left of significant digits.

Forms Specifications and Dimensions

PAPER CHARACTERISTICS

The paper used for continuous forms must be of sufficient weight and strength to prevent the holes from tearing out during feeding or ejecting the form. This is particularly important when single-part forms are being used.

The paper must not be so stiff as to cause improper feeding or excessive bulging, particularly at the outfold.

Paper must be as free from paper dust or lint as possible.

WEIGHT

The number of legible copies required is a factor in determining the weight of the paper to be used in a multiple-part set.

Best results on multiple-copy forms require a lightweight paper of 13 pounds or less, except for the last copy. Again, the number of copies, as well as the distance of the form away from the hammers (this distance can be varied by use of the print-density control-lever), affects the determination of paper weight.

Feeding and legibility performance can best be determined by making test runs of sample sets of forms.

FRICTION

During the feeding operation, friction on marginally-punched continuous forms should be eliminated by the following means:

1. Place the pack of forms directly beneath the front of the printer on the forms stand, in a position that eliminates any abnormal drag on the forms.

2. Allow enough clearance between the hammers and the print chain, to permit the forms to be fed by the pins freely, and without interference. This can be accomplished by properly setting the print density control lever.

PERFORATED LINES

The perforations between forms should be deep enough to permit easy separation, but not so deep as to tear in ordinary handling or feeding through the machine.

The perforated lines at the end of the form should always be located at 90 degrees to a vertical center line through the marginal holes.

Cut and uncut portions should be uniformly accurate in length and spacing to insure proper and efficient tearing.

Vertical perforations, at the margin for removal of the marginally punched strip, can vary, depending upon requirements. The distance from the edge of the form to the marginal perforations is usually ½ inch.

MARGINAL HOLES

Continuous forms should have holes in both right and left margins, $\frac{5}{32}$ inch in diameter, spaced vertically ½ inch apart from center to center, the full length of the form. The holes should be located this way on all copies of all sets throughout each pack of forms.

It is possible, however, to use holes of any size, shape, and spacing that accomplish the equivalent feeding conditions.

Vertical lines passing through the two vertical rows of pin holes must be parallel. It is recommended that the edges of the form be ¼ inch from the vertical center lines through the holes.

A horizontal line passing through the center of any two marginal holes on the same line should be at a 90-degree angle to either vertical center lines through the marginal holes.

Spacing between holes, center-to-center, must be such that the pins in the forms tractor, ⅛ inch in dia-

meter and spaced ½ inch apart, enter and leave the holes in the paper, freely without tearing the paper.

WIDTH OF FORMS

Although forms of any width within the extremes of those shown in Figure 85 can be used, it is recommended that form widths be confined to the standard sizes shown in Figure 86.

LENGTH OF FORMS BETWEEN PERFORATED LINES

The 1403 accommodates marginally-punched continuous forms up to a maximum length of 22 inches, at 6-lines-per-inch. It is recommended, however, that form lengths be confined to regular lengths, such as 3, 3⅓, 3½, 3⅔, 4, 4¼, 5, 5½, 6, 7, 8, 8½, 10, 11, 12, 14, 16, and 17 inches.

LINE SPACING

The forms tractor of the IBM 1403 can be set by the operator for single-space printing, 6- or 8-lines-per-inch. For 6-lines-to-the-inch spacing, the length of the form must be evenly divisible by ⅙ inch for single spacing, by ⅓ inch for double spacing, and by ½ inch for triple spacing. Similarly, 8-lines-to-the-inch spacing requires that the length of the form be evenly divisible by ⅛ inch for single spacing, by ¼ inch for double spacing, and by ⅜ inch for triple spacing.

Single-space, 8-lines-per-inch printing on the 1403 is not recommended when the registration between lines is critical.

OVER-ALL WIDTH (INCHES)	HOLE-TO-HOLE (INCHES)
4¾	4¼
5¾	5¼
6½	6
8	7½
8½	8
9½	9
10⅝	10⅛
11	10½
11¾	11¼
12	11½
12 27/32	12 11/32
13⅝	13⅛
14⅞	14⅜
15½	15
16	15½
16¾	16¼
17 25/32	17 9/32

Figure 86. Standard-Size Forms

MULTIPLE COPIES

Multiple-copy forms consisting of more than four parts, and forms with the first part made of paper of more than 13-pound weight, should be tested under operating conditions to determine the suitability of feeding and legibility.

If multiple-copy forms are not fastened together, the carbon paper must be kept in line with the form by an acceptable method. One such method is center carbon without pin holes, glued to the set, or full-width carbon paper punched with substantially larger marginal holes that are approximately centered with the corresponding holes in the form. Marginal holes in the carbon that are substantially larger than the corresponding holes in the forms make allowance for carbon shrinkage and provide the processing tolerance necessary for some of the commonly used form structures.

One-time carbon paper or carbon-backed paper can be used. The carbon paper or coating should produce the required number of legible copies without excessive smudging. This can be determined best by making test runs with sample sets of forms containing different qualities of carbon papers.

FASTENING OF MULTIPLE-COPY FORMS

The width, length, and number of copies of the form determine the fastening requirements for satisfactory feeding through the forms tractor. For most efficient stacking, however, it is recommended that a suitable fastening method always be used with multiple copy forms.

If the construction of the form is such that the parts are of different widths, the necessity for, and the method of, fastening the form should be determined by the width of the parts, the depth of the form (shown in Figure 87), and weight of paper.

Forms of fanfold construction can be used on the IBM 1403 Printer.

When card-tag or rag-content paper stock is used, a test of sample sets of forms should be made to deter-

FORM DEPTH (Inches)	MAXIMUM DISTANCE BETWEEN FASTENINGS (Inches)
1 to 5	5
5-1/5 to 11	11
11 to 14	7
14 to 17	8½

Figure 87. Fastening Requirements for Multiple-Copy Forms

mine the exact fastening requirements. The fastening may consist of any satisfactory method, such as stitching or gluing, that prevents the copies from shifting. It is essential, however, that whatever fastening medium is used should not impair the feeding or printing alignment of the form.

REGISTRATION OF FORMS

The assembly of multiple-copy forms should insure that the punching and printing of all copies of the form are in absolute registration with the material printed by the 1403. The following tolerances should be maintained.

Vertical Lines: Vertical columns of print positions are spaced 1/10 inch apart. There are 50 printing spaces in 5 inches. Vertical rules printed on a form should be spaced in multiples of 1/10 inch.

The center line of any one character, with reference to any other character on the same line, may have a plus or minus tolerance of .0065 inch, or a maximum over-all tolerance of .013 inch. From a forms viewpoint, it is practically impossible to guarantee that the cumulative tolerance of printing-plate shrinkage, paper shrinkage, and marginal-hole perforations does not exceed .0065 inch. This precludes the possibility of retaining satisfactory registration if vertical rules are spaced to split between print positions.

Where vertical lines are required, such rules should split the respective print position, thereby assigning that particular position for the columnar field (dollars and cents, for example) separation. However, in view of the fact that the 1403 can print special characters such as period and comma in every print position, the use of these symbols as decimal points, etc., avoids the need for vertical lines for such separations.

Vertical printed lines should parallel a vertical center line passing through the marginal holes.

Horizontal Lines: Horizontal printed lines on the form should be at a 90-degree angle to the vertical center line passing through the paper-feed pin holes. The spacing should conform to the setting of the 1403 forms tractor — 6- or 8-lines-to-the-inch.

Margins: It is recommended that no staples or other metal fasteners be used with multiple-copy forms. If such practice is unavoidable, it is important that either the left or right margin (whichever has the staples) be set outside the print hammer area, so that staples or other metal fasteners do not pass between the chain and hammer unit.

IBM 1403 Operation Codes

See Figure 88 for 1403 operation codes.

Write a Line (v M)

Write a Line, Word Marks Create Blanks in Printing (v L)

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
W or W1 (Ch 1)	v M	%20	bbbbb	W
W2 (Ch 2)	v M	□20	bbbbb	W
WW or W1W (Ch 1)	v L	%20	bbbbb	W
W2W (Ch 2)	v L	□20	bbbbb	W

Function: This instruction transfers 100 (or 132) characters in core storage to the print buffer. The hundreds position of the x-control field specifies the channel (% is channel 1). The tens position of the x-control field specifies the input-output unit used (2 is printer). The units position of the x-control field specifies the operation (0 specifies a write operation without word marks). The B-address specifies the high-order (starting) position of the data record (B-field) in storage. Data records are transferred from high-order to low-order position (left to right). The operation is stopped by the first group-mark — word-mark sensed in core storage. At the end of a correct data transfer, the printer is started and prints a line with the data just transferred. If the v L Op code is used, word marks are translated to word separators during the transfer to the print buffer. There is no character on the chain for a word separator, and a blank space re-

I/O UNIT	OP CODE	X-CTRL FIELD	DESCRIPTION	MNEMONIC	d-CHAR	OPERATION	NOTES
PRINTER	v M	%20 (Ch 1) □20 (Ch 2)	Write a line	W or W1 (Ch 1) W2 (Ch 2)	W	Transfer 100 or 132 characters from storage to print buffer and print a line.	Word marks in storage area are not transferred.
	v L		Write a line, word marks create blanks in printing	WW or W1W (Ch 1) W2W (Ch 2)			Word marks in storage area transfer as word separators and "print" as blanks ahead of associated characters.
	v M	%21 (Ch 1) □21 (Ch 2)	Write word marks as 1's	WM or WM1 (Ch 1) WM2 (Ch 2)		Transfer word marks from storage to print buffer and print as 1's.	Positions without word marks transfer and print as blanks. (With v L op code, word marks transfer as word separators and also "print" as blanks; thus no printing results.)
CARRIAGE	v F (Ch 1)	None	Carriage control	CC or CC1	See Figure 90 for the list of d-characters and operations.		
	v 2 (Ch 2)			CC2			

● Figure 88. IBM 1403 Op Codes

sults during the printing. When operation is in the load mode, a blank space (corresponding to the position that contained the word separator) appears ahead of its associated character. This *increases* the result field length in the print buffer and may cause the loss of some of the field.

Word Marks: A group-mark – word-mark must appear in the core storage position to the immediate right of the data record.

Timing: T = 49.5 + I/O.

(See "IBM 1403 Printer Timing Considerations.")

Note: A branch if any I/O channel status indicator on instruction, $\check{R}$ (I) ≡, must be given before the next channel 1 I/O unit operation, to be sure that the data were correctly transferred.

(A specific $\check{R}$ (I) d instruction may be substituted for the $\check{R}$ (I) ≡ instruction. This will not cause a system interlock as long as a branch to the I-address, specified in the $\check{R}$ (I) d instruction, occurs.) See Figure 89.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

Write Word Marks As 1's

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
WM or WM1 (Ch 1)	$\check{M}$	%21	bbbbb	W
WM2 (Ch 2)	$\check{M}$	□21	bbbbb	W

Function: This instruction transfers all word marks in the data field to the print buffer as 1's. The 1 in the units position of the x-control field specifies the write word mark operation. Positions without word marks are transferred to the print buffer as blanks. At the end of the data transfer, the printer is started and prints a line with the word mark data just transferred. If the $\check{L}$ op code is used, the effect on printing is the same as using the $\check{L}$ op code in write a line; consequently, no printing will result.

Word Marks: A group-mark – word-mark must appear in the core-storage position to the immediate right of the data record.

Timing: T = 49.5 + I/O.

(See "IBM 1403 Printer Timing Considerations.")

Note: A branch if any I/O channel status indicator on instruction, $\check{R}$ (I) ≡, must be given before the next channel 1 I/O unit operation, to be sure that the data were correctly transferred.

(A specific $\check{R}$ (I) d instruction may be substituted for the $\check{R}$ (I) ≡ instruction. This will not cause a system interlock as long as a branch to the I-address, specified in the $\check{R}$ (I) d instruction, occurs.) See Figure 89.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

Control Carriage

Instruction Form:

Mnemonic	Op Code	d-character
CC or CC1	$\check{F}$ (Ch 1)	x
CC2	$\check{2}$ (Ch 2)	x

Function: The tape-controlled carriage is instructed to skip to the channel specified by the d-character. The numeric portion of the d-character specifies the number of spaces to be taken or the tape channel hole that terminates the skip. The d-characters and the operations they initiate are shown in Figure 90. An instruction for an immediate skip to a channel at which the carriage tape is already positioned causes the

INDICATOR	d-CHARACTER	CONDITION
Not Ready	1	Printer not ready, Printer not on line, Printer power off, Printer out of forms
Busy	2	Previous line still being printed
Data Check	4	Print buffer detects parity error (line is not printed)
Condition	8	Print buffer detects timing error, Print buffer detects hammer fire check (line is not printed)
Wrong Length Record	– (B-bit)	Wrong length record (line following error line is not printed)
No Transfer	⅋ (A-bit)	Never set

● Figure 89. I/O Channel Status Indicators Set During Write a Line and Write Word Marks Operations

d	IMMEDIATE SKIP TO	d	SKIP AFTER PRINT TO	d	IMMEDIATE SPACE
1	Channel 1	A	Channel 1	J	1 Space
2	Channel 2	B	Channel 2	K	2 Spaces
3	Channel 3	C	Channel 3	L	3 Spaces
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	SPACE AFTER PRINT
/	1 Space
S	2 Spaces
T	3 Spaces

Figure 90. d-Characters for Control Carriage Instructions

carriage to move to the next punch of that same channel number.

An automatic single-space is initiated at the completion of a successful data transfer from the CPU, only if the forms have not been moved since the last print line. To prevent the automatic space, forms may be moved by either a forms operation or by pressing the space or restore keys on the 1403.

See Figure 91 for I/O channel status indicators that may be set during a carriage operation.

Word Marks: Word marks are not affected.

Timing: T = 13.5.

(See "IBM 1403 Printer Timing Considerations.")

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

INDICATOR	d-CHARACTER BIT	CONDITION
Not Ready	1	Printer not ready, Printer not on line, Printer power off, Printer out of forms
Busy	2	Forms in motion forms instruction waiting to be executed
Data Check	4	Never set
Condition	8	
Wrong Length Record	B	
No Transfer	A	

Figure 91. I/O Channel Status Indicators Set During IBM 1403 Carriage Operation

IBM 1403 Printer, Timing Considerations

The transfer of data from the print area of core storage to the print buffer requires 1,100 microseconds for 100 print positions, and 1,452 microseconds for 132 print positions. The printer is not busy at this time; "busy" comes on at the successful completion of the transfer. It remains on for a minimum of 82,420 microseconds if there is not an automatic space, or a minimum of 103,820 microseconds if there is an automatic space. In case of an unsuccessful transfer, the printer may be readdressed immediately by the CPU; however, the second data transfer will not actually start until 1,463 microseconds after the initiation of the first transfer.

Special Features

Numeric Print Feature

The numeric print feature for the IBM 1403 Printer has been designed for businesses having certain 1410 applications that require no alphabetic printing. For example, banks, insurance companies, and utilities prepare many reports with only numeric printing. With this feature, the time required to produce these reports can be reduced by as much as 50 per cent. The manufacturing, wholesaling, and retailing levels of other industries can also use this feature for the many applications in which reports are (or can be) numerically coded.

With this feature, the system user can switch from the alphameric to the numeric mode, simply by changing the chain cartridge in the 1403. The numeric chain is composed of 15 character sets, with 16 characters (digits 0 through 9 $. , * - □) in each set. In the numeric mode, the 1403 can print 1285 lines per minute – more than twice as fast as in the alphameric mode.

To change from one mode to another, an operator, using no special tools, removes one chain and replaces it with the other. Before locking the new cartridge in place, it is only necessary to move the chain enough to permit the chain drive to engage. When a chain cartridge is placed in the 1403, the corresponding mode is selected automatically. If the printer is in the numeric mode, characters other than the 16 specified for numeric printing cause a print check error.

Interchangeable Chain Cartridge Adapter

Many scientific and commercial applications require distinctive type styles for particular printing jobs. This special feature for the IBM 1403 Printer allows chain cartridges to be interchanged.

With this feature, an operator can insert an interchangeable chain cartridge with a different type font, type style, or special character arrangement.

The procedure for changing a cartridge is:

1. Turn off system power.
2. Lift up the printer cover.
3. Pull back and unlock the print unit release lever.
4. Unlatch the ribbon shield and swing it against the paper.
5. Open the ribbon cover and remove the lower ribbon spool. Slide ribbon from under the skew roll and store the lower ribbon spool on the ribbon cover.
6. Grasp the cartridge handles and raise them to a vertical position. (This unlocks the cartridge from the T-casting.)
7. Lift straight up on the handles and raise the cartridge until it clears its locating pins. At this point it is free from the machine. Place the cartridge on a surface that will tolerate oil and ink. (A container is provided for storing the cartridge that is not in use.)

8. Grasp the handles of the second interchangeable cartridge and, raising them to a vertical position, lift the cartridge into position over the locating pin. (Check for foreign matter clinging to underside of cartridge.)

9. Lower the cartridge gently into position over its guide pins and release the handles *(do not force either handle down at this point).* The 132-hammer end of the cartridge should settle fully down to the base. The 1-hammer end will not be down in position at this time.

10. Rotate the chain in the normal printing direction (counterclockwise, as viewed from the top). The chain can be rotated by pressing your finger against a character on the chain. At the same time, apply pressure to the button (located between the print-timing dial and the cartridge) on the top cover. Rotate the chain slowly until the drive key drops into the drive slot. The chain will stop and the cartridge will settle correctly into position on the 1-hammer end.

11. Lower the cartridge handles to their horizontal position. *Do not force.* If force is required, the cartridge is not fully seated; repeat steps 8 to 10.

12. Replace the ribbons; latch the ribbon shield into place; close the T-casting and the top cover; apply power to the system and resume printing.

Ten 729 or 7330 Magnetic Tape Units (Figure 92) can be attached to each data channel of the 1410 system. Four models of the 729 (Model II, IV, V, or VI) are available for 1410 use. Different 729 models can be intermixed and, if the Tape Intermix Feature is installed, 729's may be intermixed with 7330's (Figure 55).

The 729 and 7330 operating principles are practically identical; Figure 93 shows the essential differences in operating characteristics. Details on the operating principles and physical controls for the 7330 and 729 are in the *IBM Magnetic Tape Units Reference Manual,* Form A22-6589.

Tape Characteristics

Data are recorded in a seven-bit code, in seven parallel channels along the tape. Figure 94 shows tape characters and their corresponding codes.

Records are separated from each other by about ¾ inch of blank (unrecorded) tape, called an interrecord gap.

Each tape character is composed of an even number of magnetic bits. A check bit (labeled C in Figure 94) is written if the number of bits in the other six positions is odd. An even-parity check on each character ensures accuracy for tape-read and tape-write operations.

Tape Checking

The IBM 729 and 7330 tape units verify the validity of recorded information at the time it is written. The relative positions of the read and write gaps (Figure 95) are such that a character recorded by the write gap passes the corresponding read gap; thus, when each character of a record is written, it is read, and a parity check is applied. If an error is detected, the program receives a signal, and corrective action can be taken.

Detection of a validity check sets on a validity-check indicator, which sets on the data check I/O channel status indicator. The data check I/O channel status indicator can be interrogated by use of the

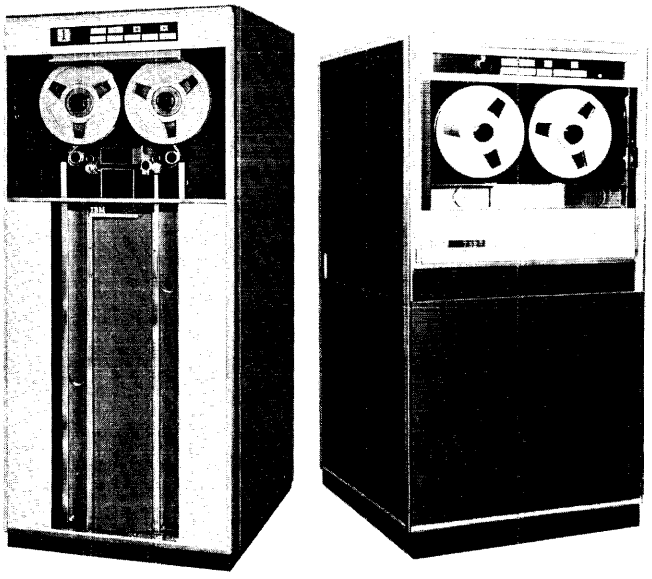


Figure 92. IBM 729 and 7330 Magnetic Tape Units

OPERATING CHARACTERISTICS	7330	729 II	729 IV	729 V	729 VI
Characters Per Inch (Recording Density)	200 or 556	200 or 556	200 or 556	200 or 556 or 800	200 or 556 or 800
Inches Per Second	36	75	112.5	75	112.5
Characters Per Second (Data Rate)	7,200 or 20,016	15,000 or 41,667	22,500 or 62,500	15,000 or 41,667 or 60,000	22,500 or 62,500 or 90,000
High Speed Rewind, Minutes	2.2	1.2	0.9	1.2	0.9
Regular Rewind, Inches Per Second	36	75	112.5	75	112.5

Figure 93. Tape Unit Characteristics

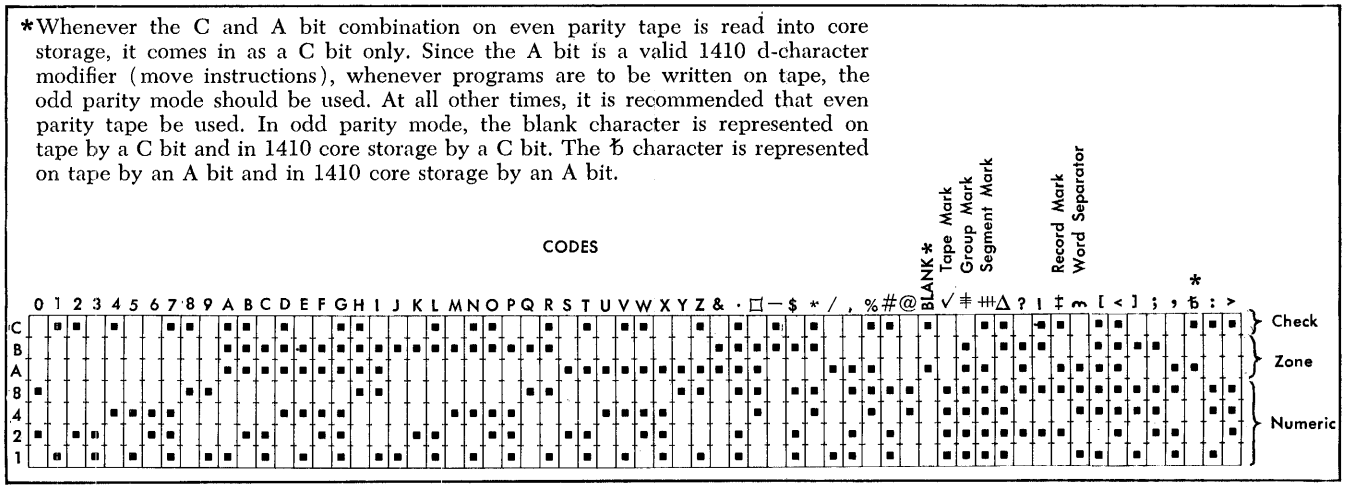


Figure 94. Magnetic Tape Seven-Bit Coding (Even Parity)

d-character 4 in the instruction, branch if data check I/O channel status indicator on.

If a tape error is detected, the tape unit can be backspaced by programming, and the record can be re-read. If the error persists, the operator can intervene, or the program can branch to an error routine.

Dust or damage to the magnetic tape is the most frequent cause of errors detected during write operations. Such imperfections are usually isolated. To skip the defective section, the 1410 has instructions to: (1) backspace over the area where the writing of the record was attempted, then (2) cause the tape to space forward about 3.5 inches when the next write operation is initiated. During the space operation, this 3.5 inch area is erased so that extraneous data are not sensed during succeeding read operations. Writing on tape is enabled to continue after the skip is completed.

During writing from load point, a space of 3.5 inches occurs before the record is written, and start time is increased by about 27 milliseconds.

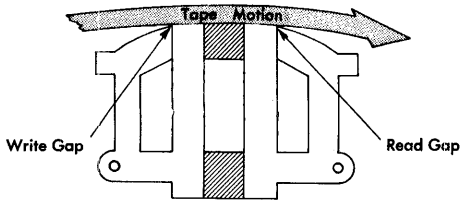


Figure 95. Read and Write Gap

Magnetic Tape Control Instructions

These instructions control reading and writing magnetic tape (with or without word marks), backspacing tape, writing tape marks, rewinding tape reels, and skipping over defective areas. There are three types of magnetic tape control instructions — unit control, read and write tape, and read and write tape with word marks (Figure 96).

In magnetic tape operation, the character in the hundreds position of the x-control field indicates the channel and overlap mode: %, Ch 1, non-overlap; @, Ch 1, overlap; □, Ch 2, non-overlap; *, Ch 2, overlap.

Unit Control

Instruction Form:

Mnemonic	Op Code	X-control field	d-character
(See Figure 96)	U	$\%x^2x^3$	x

Function: The tape unit, specified by the unit-position character in the x-control field, performs the operation indicated by the d-character. The d-characters and the operations they initiate are:

OPERATION	d-CHARACTER
Backspace tape	B
Skip and blank tape	E
Write tape mark	M
Rewind	R
Rewind and unload	U

Erasing Tape: In bypassing a defective section where tape writing has been attempted, the tape is backspaced one record and erased forward about 3½

	OPERATION CODE	MNEMONIC	d-CHARACTER	OPERATION	NOTES
v U	Backspace tape	BSP	B	Tape unit backspaces over one complete tape record.	A tape mark is considered a tape record. The 1410 is not interlocked during the operation.
	Skip and blank tape	SKP	E	Erases 3.5 inches of tape before next tape-write. A tape-read or backspace-tape cancels the SKP operation.	The next instruction should be a tape-write operation for the same tape unit.
	Write tape mark	WTM	M	A tape mark is written on tape as a single-character record.	The 1410 is interlocked during the operation.
	Rewind	RWD	R	Tape unit rewinds, loads its tape, and positions itself at load point.	At the completion of the operation, the tape unit is in a ready status at load point. (Always low speed on 7330.)
	Rewind and unload	RWU	U	Tape unit rewinds and unloads its tape.	At the completion of the operation, the tape unit is effectively disconnected. (High speed on 7330 requires manual reloading.)
v M	Read tape	RT or RTB	R	A record is transferred from magnetic tape to core storage.	Reads to first v ≢ or IRG.
	Write tape	WT or WTB	W	A record is transferred from core storage to magnetic tape.	Writes to first v ≢.
v L	Read tape with word marks	RTW or RTBW	R	A record with word marks is transferred from magnetic tape to core storage.	Reads to first v ≢ or IRG.
	Write tape with word marks	WTW or WTBW	W	A record with word marks is transferred from core storage to magnetic tape.	Writes to first v ≢.
v M	Read tape	RTG or RTBG	$	Read from magnetic tape to core storage. Group mark—word-marks in core storage have no effect on operation.	Stop transfer when IRG is sensed or last core-storage position is encountered.
v L	Read tape with word marks	RTGW or RTBGW	$		
v M	Write tape	WTE or WTBE	X	Contents of core storage are written on tape. Group-mark — word-marks in core storage have no effect on operation.	Stop transfer when last core-storage position is encountered.
v L	Write tape with word marks	WTEW or WTBEW	X		

Note: B in mnemonic specifies odd-parity mode (binary).

● Figure 96. Magnetic Tape Control Instructions

inches; then the tape writing restarts. The following sequence of instructions must be used:

INSTRUCTION	ACTUAL	MNEMONIC
Backspace tape	v U $x^5x^2x^8$ B	BSP
Skip and blank tape	v U $x^5x^2x^8$ E	SKP
Write tape . . .	v M or v L $x^1x^2x^8$ bbbbb W (or X)	WT or WTW (or WTEW, WTBO*, etc.)

x^1 = %, @, □, *
x^2 = U or B (even parity or odd parity)
x^8 = Designated number of tape unit
x^5 = % or □ (Ch 1 or 2 non-overlap)
* Write a record on tape in the move mode, using odd parity and the overlap special feature.

Word Marks: Word marks are not affected.

Timing: T = .0045 (L + 1) + Tm ms.
(See "Tape Timing Considerations.")

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	Bp

Read or Write Tape

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
RT or WT	v M	%Ux^8	bbbbb	R or W
RTB or WTB	v M	%Bx^8	bbbbb	R or W
RTG or WTE	v M	%Ux^8	bbbbb	$ or X
RTBG or WTBE	v M	%Bx^8	bbbbb	$ or X

Function of RT or WT Instructions: The x[3] character in the units position of the x-control field specifies the tape unit that performs the operation. The B-address specifies the high-order position of the tape record core storage area. If the d-character is R, the tape record is read into core storage from the tape. If the d-character is W, the tape record in core storage is written on the tape. Either an inter-record gap in the tape record or a group-mark – word-mark in core storage (whichever is sensed first) stops a read tape operation. If a group-mark – word-mark is sensed first, the data transfer stops, but the tape movement continues to the next inter-record gap. A group-mark – word-mark in core storage stops the write operation and causes an inter-record gap on the tape.

Word Marks: Word marks in the tape record do not affect either operation.

Timing: T = .0045 (L + 1) + Tm ms.
(See "Tape Timing Considerations.")

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

Note: If the x-control field contains %Bx[3], the magnetic-tape operation is performed in an odd-parity mode. A tape mark is always even parity. If a tape mark is encountered during an odd-parity operation, a data check and an end-of-file indication result. Depending on the position of the asterisk insert switch and other console controls, the tape mark ($\vee$) is stored as either 8421 (no C), an unchanged and invalid character, or it is converted to and stored as an asterisk. (See "Asterisk Insert Switch" under "IBM 1415 Console CE Test Panel.") In even-parity mode (%Ux[3]), the tape mark enters storage as C8421 and no data check accompanies the end-of-file indication.

Function of RTG or WTE Instructions: These instructions differ from the RT and WT instructions only as follows. Group-mark – word-marks in core storage have no effect on either operation. With a d-character of $, characters are read from tape into core storage until either an inter-record gap on the tape is sensed or the last position in core storage is filled. The d-character of X causes the contents of core storage to be written on tape, starting at the core-storage location specified in the B-address and continuing until the last core-storage position is encountered. Instructions using the $ or X d-character cannot be overlapped.

Read or Write Tape with Word Marks

Instruction Form:

Mnemonic	Op Code	X-control Field	B-address	d-character
RTW or WTW	$\vee$ L	%Ux[3]	bbbbb	R or W
RTBW or WTBW	$\vee$ L	%Bx[3]	bbbbb	R or W
RTGW or WTEW	$\vee$ L	%Ux[3]	bbbbb	$ or X
RTBGW or WTBEW	$\vee$ L	%Bx[3]	bbbbb	$ or X

Function: These two sets of instructions have the same functions as the read or write tape instructions, except that word marks are written on tape as word separators, and word separators are read into storage as word marks. In a write operation, a word separator is written on tape, one position ahead of the associated character (Figure 97). In a tape read operation, word marks are associated with the next character read from tape (Figure 98).

Word Marks: Each word separator requires one tape character position.

Timing: T = .0045 (L + 1) + Tm ms.
(See "Tape Timing Considerations.")

Note: This instruction is used whenever word marks must be indicated in the tape record. If a tape record is written with word marks, it must be read with word marks when the data are required for a subsequent

1410 core-storage location	A	B	C	
1410 core-storage code	C82	41W	4	
1410 meaning	0	$\vee$ 5	4	
Tape positions	A	B	C	D
Tape code	82	A841	41	C4

Figure 97. Word Mark Translation for Write Tape with Word Marks

Tape positions	A	B	C	D
Tape code	82	A841	41	C4
1410 meaning	0	$\vee$ 5	4	
1410 core-storage location	A	B	C	
1410 core-storage code	C82	41W	4	

Figure 98. Word Mark Translation for Read Tape with Word Marks

operation. This ensures proper translation between the tape and core storage.

These instructions can be done in an odd-parity mode. The x-control field and mnemonics are handled as described in read or write tape.

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	Ap	B + LB + 1

Tape Unit Operation Status Indicators

A branch if any I/O channel status indicator on instruction must be given before the next I/O unit operation on the channel being used to be sure that the data were correctly transferred.

(A specific R̆- or X̆ (I) d instruction may be substituted for the R̆- or X̆ (I) ≢ instruction. This will not cause a system interlock as long as a branch to the I-address specified in the R̆- or X̆ (I) d instruction occurs.) Figure 99 shows the I/O channel-status indicators set during tape operations.

INDICATOR READ(R),WRITE(W), CONTROL UNIT(U)	d-CHARACTER BIT	CONDITION
Not Ready (R-W-U)	1	Tape unit not ready No such tape unit selected Tape adapter unit not on line Tape adapter unit power off
Busy (R-W-U)	2	Tape unit rewinding Tape adapter unit busy (backspace or 7330 read-write not finished)
Data Check (R) (W) (U)	4	Processing unit received wrong parity character Tape adapter unit sent wrong parity character Tape mark read in odd parity mode Tape adapter unit received wrong parity character Tape adapter unit detects rbc parity error Set if write tape mark in odd parity
Condition (R) (W) (U)	8	1st character of record was tape mark Foil strip detected Never set (unless tape mark read)
Wrong Length Record (R) (W-U)	B	Wrong length record (usually set when d-character is $) Never set (unless record is of zero length and first character written is ǂ)
No Transfer (R-W-U)	A	Never set

● Figure 99. I/O Channel Status Indicators Set During Tape Operations

Tape Timing Considerations

All tape units on a given channel in a 1410 system are under control of a tape adapter unit (TAU). TAU can control the operations of only one tape unit at a time. If one tape unit is busy, no other tape unit can be used until all operations (except rewinding) on the busy one have been completed. The execute times of IBM 1410 tape instructions vary according to the type and model of tape units used in the system (Figure 100). In the following formulas:

C is the character rate in milliseconds (ms) based on the setting of the tape density switch.

N is the number of characters in the record.

CN is record time (number of characters in the record, times the character rate).

Start time is the time necessary for the tape unit to accelerate to operating speed.

Stop Time is the time necessary for the tape unit to decelerate and stop.

Tm — Tape movement can be determined from the following:
N = Number of characters in the record
C = Character rate in milliseconds

729 II at 200 cpi = 0.067 ms
at 556 cpi = 0.024 ms
729 IV at 200 cpi = 0.044 ms
at 556 cpi = 0.016 ms
729 V at 200 cpi = 0.067 ms
at 556 cpi = 0.024 ms
at 800 cpi = 0.017 ms
729 VI at 200 cpi = 0.044 ms
at 556 cpi = 0.016 ms
at 800 cpi = 0.011 ms
7330 at 200 cpi = 0.139 ms
at 556 cpi = 0.050 ms

729 Models II, V: READ, 10.7 + CN ms with TAU and processing interlocked
WRITE, 11.7 + CN ms with TAU and processing interlocked

729 Models IV, VI: READ, 7.1 + CN ms with TAU and processing interlocked
WRITE, 7.8 + CN ms with TAU and processing interlocked

7330: READ, 20.5 + CN ms with TAU interlocked
10.8 + CN ms with processing interlocked
WRITE, 20.3 + CN ms with TAU interlocked
13.8 + CN ms with processing interlocked

Rewind
729 Models II, V = 1.2 minutes per reel
729 Models IV, VI = 0.9 minute per reel
7330 = 2.2 minutes per reel at high-speed

Skip and Blank Tape (add to subsequent write time)
729 Models II, V = 40.5 ms
729 Models IV, VI = 27 ms
7330 = 104 ms

Backspace (after READ)
729 Models II, V = 46 + CN ms
729 Models IV, VI = 33 + CN ms
7330 = 428 + CN ms

Backspace (after WRITE)
729 Models II, V = 54 + CN ms
729 Models IV, VI = 39 + CN ms
7330 = 435 + CN ms

Figure 100. Tape Movement Specifications

Record Check Time is the time necessary 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.

IBM 729 II and V Tape Timings

READ

During a read operation, the tape adapter unit and the processing unit are interlocked for 10.7 + CN ms (Figure 101). This includes:

10.5 ms – start time
.2 ms – record check time for 556-cpi tape
(.6 ms for 200-cpi tape; .15 ms for 800-cpi tape)
CN ms – record time

Therefore, in a tape read operation, processing can take place during 1.9 ms of stop time for 556-cpi tape, 1.5 ms for 200-cpi tape, or 1.95 ms for 800-cpi tape.

WRITE

During a write operation, the tape adapter unit and the processing unit are interlocked for 11.7 + CN ms (Figure 101). This includes:

7.5 ms – start time
4.2 ms – record check time for 556-cpi tape
(4.6 ms for 200-cpi tape; 4.15 ms for 800-cpi tape)
CN ms – record time

Therefore, in a tape write operation, processing can take place during .9 ms of stop time for 556-cpi tape, .5 ms for 200-cpi tape, or .95 ms for 800-cpi tape.

NOMINAL FORMULA

For job timing estimates of tape read and write operations, the nominal formula 10.8 + CN ms can be used.

IBM 729 IV and VI Tape Timings

READ

During a read operation, the tape adapter unit and the processing unit are interlocked for 7.14 + CN ms (Figure 102). This includes:

7.0 ms – start time
.14 ms – record check time for 556-cpi tape
(.4 ms for 200-cpi tape; .1 ms for 800-cpi tape)
CN ms – record time

Therefore, in a tape read operation, processing can take place during 1.96 ms of stop time for 556-cpi tape, 1.7 ms for 200-cpi tape, or 2.0 ms for 800-cpi tape.

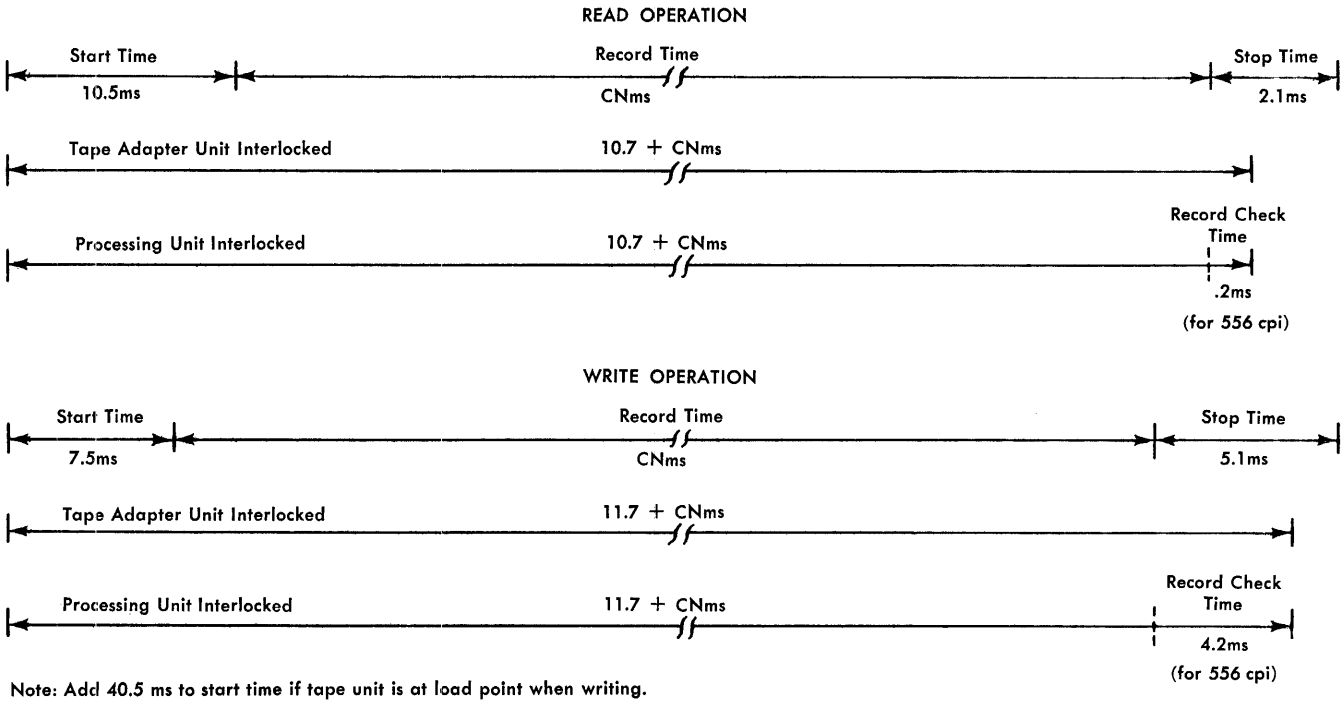


Figure 101. IBM 729 II and V Read-Write Tape Timing

WRITE

During a write operation, the tape adapter unit and the processing unit are interlocked for 7.8 + CN ms (Figure 102). This includes:

5.0 ms – start time
2.8 ms – record check time for 556-cpi and 800-cpi tape (3.0 ms for 200-cpi tape)
CN ms – record time

Therefore, in a tape write operation, processing can take place during 1.3 ms of stop time for 556-cpi or 800-cpi tape, and 1.1 ms for 200-cpi tape.

NOMINAL FORMULA

For job timing estimates of tape read and write operations the nominal formula 7.3 + CN ms can be used.

IBM 7330 Tape Timings

READ

During a 7330 read operation, the tape adapter unit is interlocked for 20.5 + CN ms (Figure 103). This includes:

10.3 ms – start time
9.8 ms – stop time
.4 ms – record check time for high-density tape (1.0 ms for low-density tape)
CN ms – record time

During the same read operation, the processing unit is interlocked for 10.8 + CN ms. This includes:

10.3 ms – start time
.1 ms – stop time
.4 ms – record check time for high-density tape (1.0 ms for low-density tape)
CN ms – record time

Therefore, in a tape read operation, processing can take place during 9.7 ms of stop time for 556-cpi tape or 9.1 ms for 200-cpi tape.

WRITE

During a 7330 write operation, the tape adapter unit is interlocked for 20.3 + CN ms (Figure 103). This includes:

5.0 ms – start time
6.6 ms – stop time
8.7 ms – record check time for high-density tape (9.3 ms for low-density tape)
CN ms – record time

During the same write operation, the processing unit is interlocked for 13.8 + CN ms. This includes:

5.0 ms – start time
.1 ms – stop time
8.7 ms – record check time for high-density tape (9.3 ms for low-density tape)
CN ms – record time

Therefore, in a tape write operation, processing can take place during 6.5 ms of stop time for 556-cpi tape or 5.9 ms for 200-cpi tape.

NOMINAL FORMULA

For job time estimates of tape read and write operations, the nominal formula 20.7 + CN ms can be used.

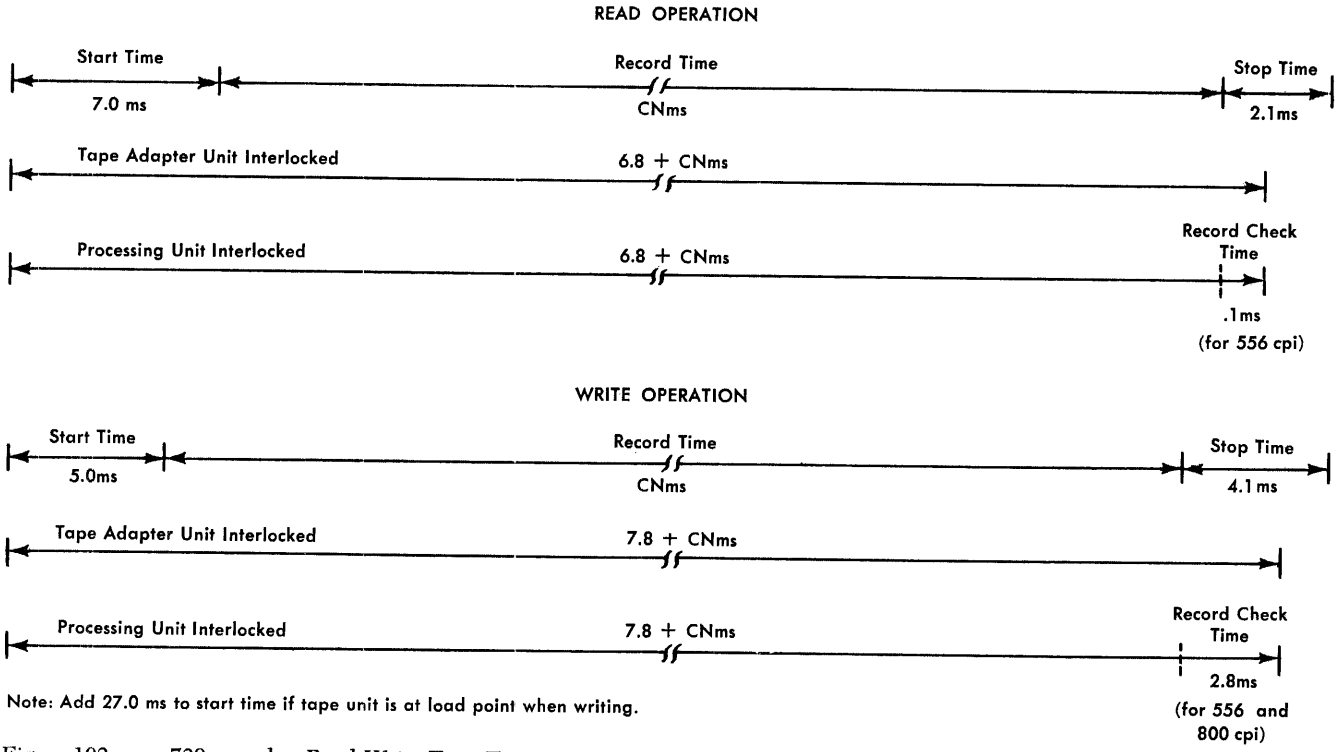


Figure 102. IBM 729 IV and VI Read-Write Tape Timing

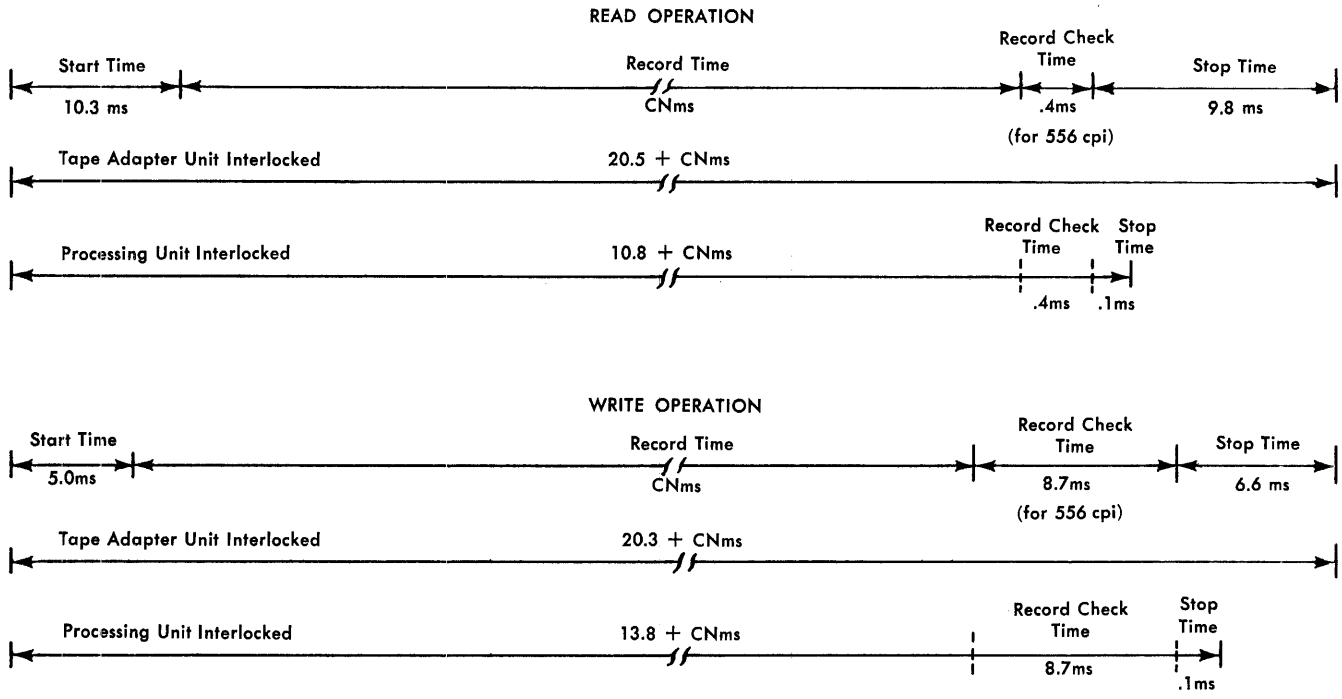


Figure 103. IBM 7330 Read-Write Tape Timing

Data Channel 2

The IBM 1410 Data Processing System has one standard data channel (channel 1) for transmitting or receiving information from I/O devices. An additional channel (channel 2) is a special feature that enables a 1410 system to serve many more I/O or auxiliary storage devices with increased efficiency. All units served by either or both channels are shown in the *IBM 1410 Data Processing System Configuration Bulletin*, A22-6688.

Processing Overlap

The basic IBM 1410 Data Processing System is interlocked whenever an input-output operation is being performed. No processing can occur because the core storage area is being used during the operation. However, an IBM 1410 having the processing overlap feature allows computing to occur in the system while part of the input-output operation is being performed. While the I/O unit is preparing to send or receive data, the system continues computing. The computing is interrupted only as each individual character is stored in, or sent from, core storage.

Feature Components

The overlap feature necessitates the addition of transmission and controlling circuitry for each affected data channel. Each channel has two single-character registers associated with it. These registers are used as an intermediate storage area during the character-by-character data transfers between the system and the input-output units. The channel 1 registers are called the E1 and E2 registers. The channel 2 registers are called the F1 and F2 registers (Figure 5).

Each channel also has a five-character address register associated with it. This address register specifies the core storage location of the character being transferred. The channel 1 register is called the E-address register, and the channel 2 register is called the F-address register.

The six I/O channel-status indicators (discussed under "Checking Execution of I/O Instructions") show conditions resulting from overlapped input-output operations, as well as non-overlapped operations. The six indicators on channel 1 are tested with an $\check{R}$ (I) d instruction, and the six indicators on channel 2 are tested with an $\check{X}$ (I) d instruction.

Each channel has its own overlap-in-process indicator, which can be tested with a test and branch instruction – $\check{J}$ (I) 1 for overlap in process on channel 1, or $\check{J}$ (I) 2 for overlap in process on channel 2. The indicator is turned on at the beginning of an overlapped operation and is automatically turned off when the operation is completed. When the light is on, it indicates that the presently specified I/O unit is not busy completing a previous operation, and has started performing the current operation. The $\check{J}$ (I) 1 or 2 instruction is normally executed immediately following the overlapped I/O unit instruction.

Input-Output Operation

The operation of any I/O unit is initiated by a specific input-output instruction, consisting of an operation code, an x-control field, a B-address, and a d-modifier character. For read or write operations, the operation code is either $\check{M}$ (move) or $\check{L}$ (load). This specifies a data transfer with or without word-mark transfer.

The hundreds position of the x-control field specifies which channel will be used and whether the operation will be executed in an overlapped mode. The symbols and the channel and operation they indicate are:

SYMBOL	CHANNEL	OPERATION
%	Channel 1	non-overlap mode
@	Channel 1	overlap mode
□	Channel 2	non-overlap mode
*	Channel 2	overlap mode

The two remaining positions of the x-control field, the B-address, and the d-modifier character have the same meaning, regardless of the operation mode.

NON-OVERLAP OPERATION

A non-overlapped operation is initiated by using the specified symbol in the hundreds position of the x-control field (% for channel 1; □ for channel 2) associated with the input-output instruction.

If a non-overlapped tape read operation was being executed on channel 1, the instruction would be M or L %U1 01500 R (as one example) and would be executed as shown in Figure 104.

Executing the instruction stops processing completely. Reading and checking a 100-character record from an IBM 729 II at 556 cpi takes 13.1 milliseconds (10.5 milliseconds tape start time + 2.4 milliseconds to read 100 characters +0.2 millisecond record check time).

If the processing of the record takes 15 milliseconds, then the tape is inactive until the record processing is complete. Only one operation can take place at any one time.

OVERLAP OPERATION

An overlap operation is initiated by using the specified symbol in the hundreds position of the x-control field (@ for channel 1; * for channel 2) associated with the input-output instruction.

Once the execution of the overlapped I/O instruction is begun, the program is immediately restarted and the program advances to the next instruction in sequence. No other I/O unit can be addressed on that channel until the operating I/O unit has completed its operation. This is necessary because there is only one set of I/O channel status indicators for each channel, and they must be tested before the next I/O operation begins. This status test must be completed before the next I/O unit operation on that channel is encountered and is ensured by the system's circuitry. The status test is considered complete if either:

1. A branch if any I/O channel status indicator on instruction is executed before encountering the next I/O unit instruction on the same channel; or

2. A specific X- or R (I) d instruction (see Figure 19) is executed and results in a branch before encountering the next I/O unit instruction on the same channel.

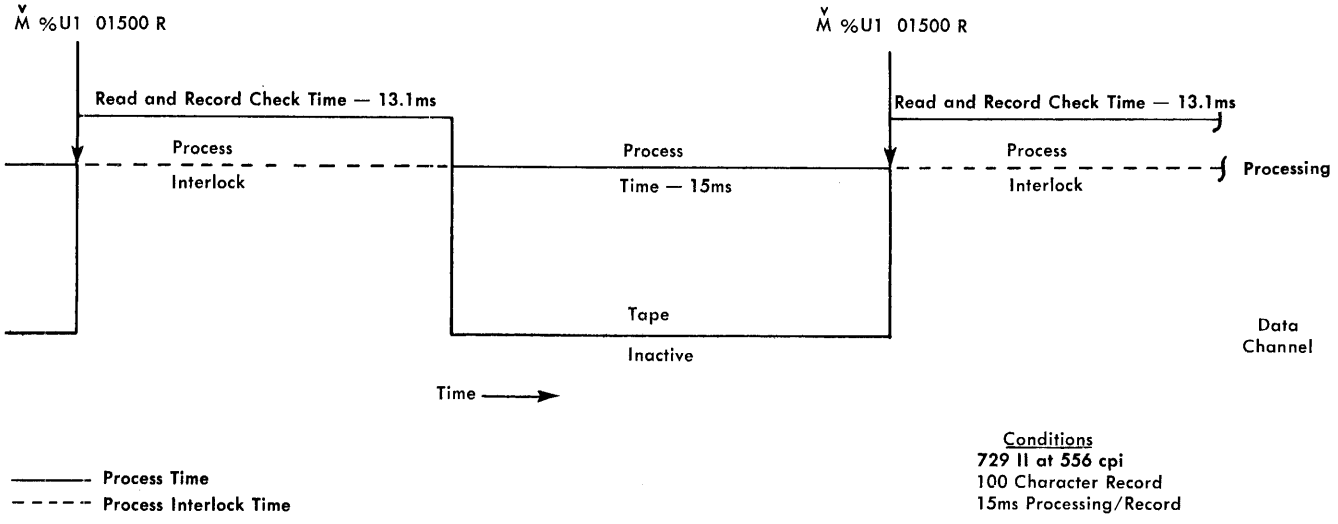
Tape Read Operation: If an overlapped tape-read operation was being executed on channel 1, the instruction would be M or L @U1 01500 R (example), and operation (Figure 105) would take place as follows:

1. The B-address of 01500 is read into the E-address register and the tape movement starts. (The A- and B-address registers continue being used in the processing that overlaps the tape-read operation.) Processing continues during tape start time.

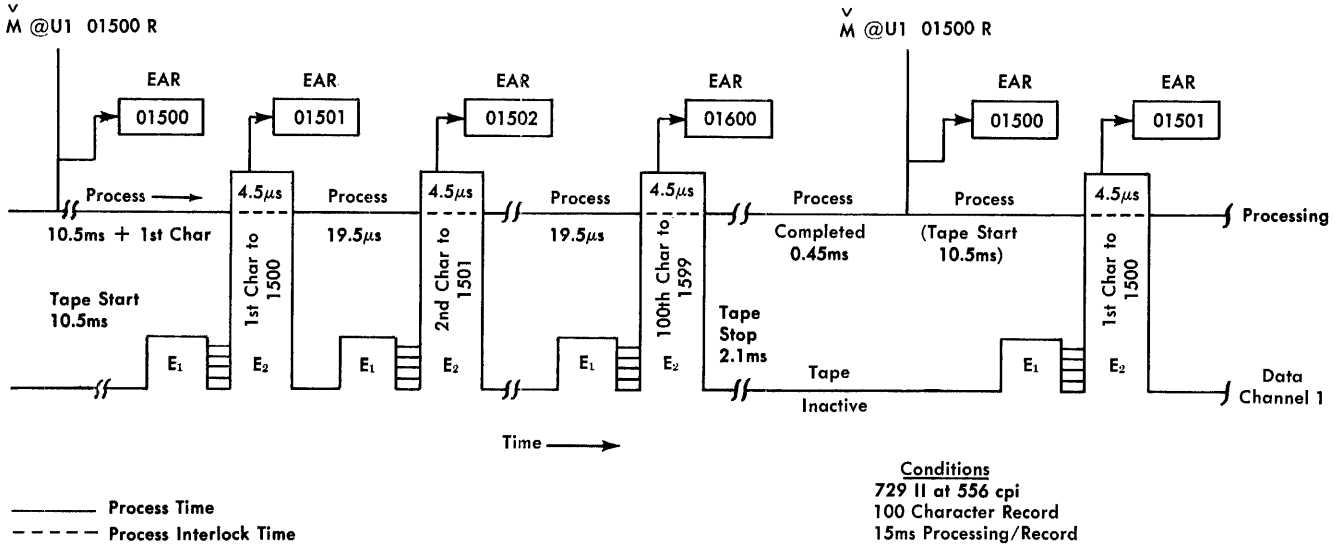
2. As soon as the E1 register receives a character from tape, the character is transferred to the E2 register.

3. At the end of the next processing cycle, processing is suspended for 4.5 microseconds, while the character is transferred from the E2 register to the core-storage position specified by the E-address register. (With the 24-microsecond character rate of the IBM 729 II at 556 cpi there is an average of 19.5 microseconds available for processing before each character is transferred.)

4. The E-address register is increased by one during the 4.5-microsecond suspension of processing. After the 4.5-microsecond suspension, processing continues.



● Figure 104. Non-Overlap Tape Read Operation



● Figure 105. Overlap Tape Read Operation

During the reading of this tape record, it is possible to process a previously-read record. If 15 milliseconds of processing time is required, 10.5 milliseconds of processing time can be utilized during the tape start time. The remaining 4.5 milliseconds of required processing time is overlapped by 4.5 microseconds for tape-character transfer each time a character is read into core storage. Therefore, the available processing time during an overlap operation is *reduced* by N × 4.5 microseconds, where N is the number of tape characters transferred.

The total interruption of processing time for 100 character transfers is 450 microseconds, while the total processing time during the 100 intervals (19.5 microseconds each) before the tape-character transfers is 1950 microseconds. (If the system has the 1410 Accelerator feature, read-in time is reduced to 4.0 microseconds; accordingly, the average interval before a character transfer increases to 20 microseconds.)

At the end of the tape-record transfer, tape stop requires 2.1 milliseconds. Processing continues during all of tape stop time, including the record-check portion. At the end of tape stop time, 0.45 millisecond is still needed to finish processing the previous record.

At the completion of the overlap operation, the E-address register contains the address that is two positions to the right of the last core-storage position read into.

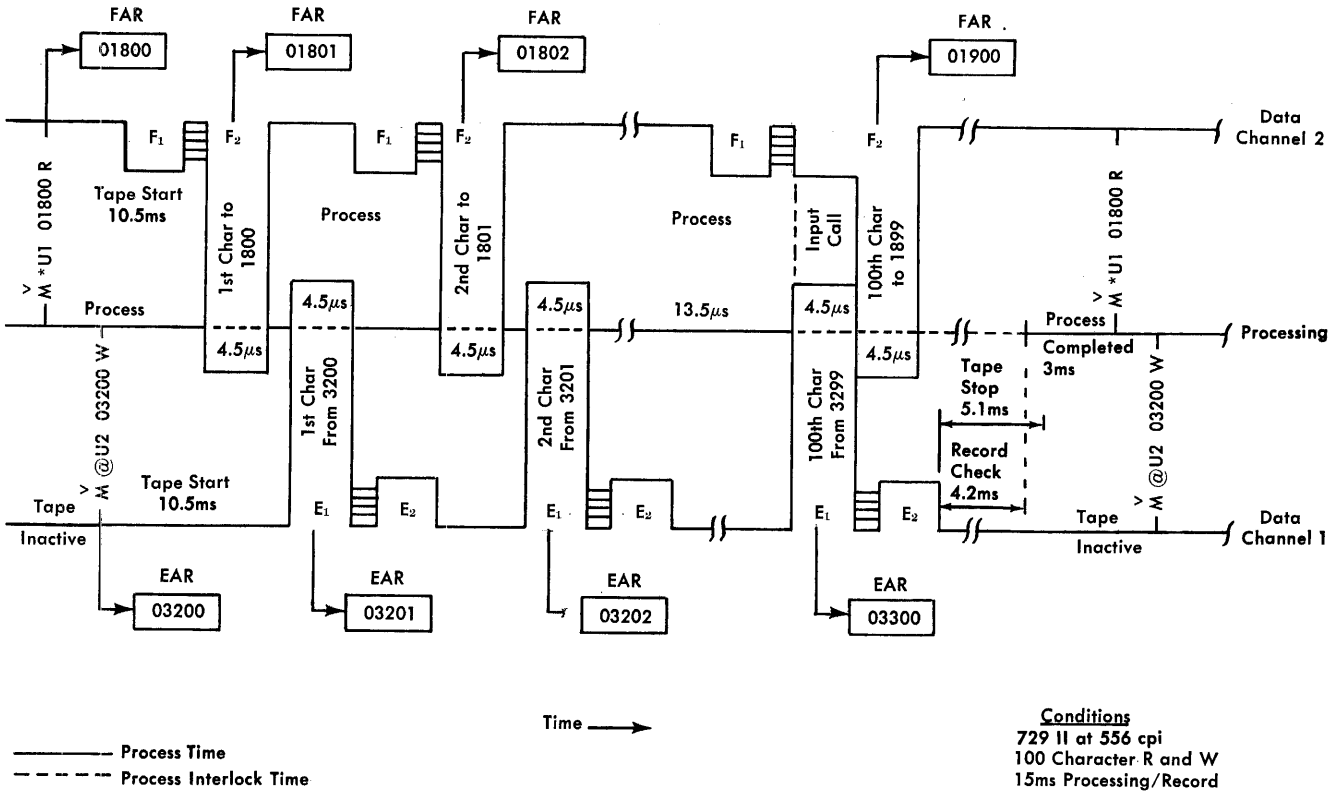
Tape Write Operation: Overlapping a tape-write operation is similar to a tape-read operation. Processing is suspended 4.5 microseconds every time a character is transferred from core storage to the E register. (The E-address register contains the core-storage location of the character transferred.) The character is transferred from the E1 register to the E2 register and then to tape during processing.

Each character read out of core storage required a 4.5-microsecond cycle. Therefore, the available processing time during an overlap operation is *reduced* by N × 4.5 microseconds, where N is the number of characters transferred. Available processing time is further reduced by the suspension of processing (for tape-write only) during record-check time. If the system has the 1410 Accelerator feature, processing is *not* suspended during record-check time; also, read-out time is reduced to 4.0 microseconds per character.

At the end of the overlap operation, the E-address register contains the address that is two positions to the right of the last core-storage position read out of.

Tape Read and Write Operation: A combination tape-read and write operation (Figure 106) makes use of both channels, and permits the overlapping of reading, writing, and processing. Each channel operates independently, and operation of either channel requires suspension of processing as soon as the processing cycle has ended. Maximum processing delay before yielding control is 11.25 microseconds. Channel control alternates between the two channels. If both channels seek control at the same time, channel 1 automatically assumes priority.

NOTE: The available processing time between each character transfer should be considered as utilized 100%, even though the time is not an exact multiple of a core storage cycle. For example, an overlapped channel 1 tape-read operation (Figure 105) had a process time between character transfers that averaged 19.5 microseconds. Because the average characters for IBM 729 II high-density tapes vary from 21-27 microseconds, the available overlapped process time then varies from 16.5–22.5 microseconds. A basic



● Figure 106. Overlap Tape Read and Write Operation

process cycle point is .75 microseconds, and a process cycle (4.5 microseconds in length – 6 cycle points × .75 microseconds) can begin immediately after a character transfer is completed. If the available process time is 15 microseconds, three complete process cycles are completed in 13.5 microseconds. Because the system has no indication that the next tape character will be available in 1.5 microseconds, another process cycle is executed. This delays the next tape character transfer by 3 microseconds; however, because each tape character passes serially through the E1 and E2 registers, it gives the channel the facility to temporarily get a character behind during the data transfer. This 3-microsecond delay is regained by picking up 1.5 microseconds on the next two tape character transfer cycles. In this way, the available process time is fully utilized.

Overlap Error or Stop Conditions: The error or stop conditions that can occur during an overlap operation are:

1. An I/O instruction, encountered before completing a previous I/O operation on the same channel, causes an immediate system stop. This condition requires reprogramming.

2. A processing unit error, occurring during an overlapped tape operation, causes a system stop. Normal restart procedures must be used.

3. A programmed stop or a console stop, occurring during an overlapped input-output unit operation, cause processing to stop, but the I/O unit continues until the data transfer is completed. Any restart procedure that is used must include a branch if any I/O channel status indicator on instruction so that the status test is satisfied before the next I/O instruction is encountered. The status test can also be satisfied by an X̌- or Ř (I) d instruction that results in a branch.

Overlap Operational Considerations: Operational considerations during an overlapped operation are:

1. An overlapped I/O instruction executed on an IBM 1410 Data Processing System without the overlap feature causes the system to stop.

2. The B-field of both an overlapped and a non-overlapped I/O instruction can be indexed.

3. On a 1410 system equipped with both channels and the overlap feature, read/write/processing, read/read/processing, or write/write/processing is possible, except for I/O instructions containing d-characters of $

or X, which must always be programmed in the non-overlap mode. (If programmed in the overlap mode, they will be executed as normal read or write instructions and stop at the first ≠.)

4. Ů xxx d instructions have the same handling as Ṁ or Ḷ xxx (B)d. However, the Ů xxx B tape-backspace operations are not overlapped, because no data transfer is involved. The 1410 system is free to continue processing during the entire backspace operation, and the unit involved in the operation is released as soon as the backspace operation is under way.

5. If the console I/O printer is operated in the inquiry mode on channel 1, the operation is the same as in the non-overlapped mode except that the program continues to be overlapped with the inquiry operation. The channel 1 overlap-in-process indicator remains on until the operator presses either the cancel or the release key.

6. As soon as a branch if any I/O channel status indicator on instruction is encountered during an overlap operation, processing is stopped until the I/O operation on that channel is completed. If this processing was not stopped, any I/O channel-status indicator set on after the Ṙ (I) d or Ẋ (I) d instruction was executed would not be detected before it was turned off by the next I/O operation. An exception occurs when the I/O unit signals not-ready or busy. A not-ready or busy signal shows that the I/O operation never began, and there is no need to stop processing.

7. In overlapped tape-write operations, processing is suspended during the record-check time (see "Tape Timing Considerations") *unless* the 1410 Accelerator feature is installed. Processing continues, however, during the record-check time of any overlapped tape-read operations.

1410 Accelerator

The 1410 Accelerator, an optional feature, increases the speed of the IBM 1410 Data Processing System by reducing the memory cycle from 4.5 microseconds to 4.0 microseconds and reducing the cycle length of data operations. Any 1410 system configuration adding this feature becomes significantly faster. The internal speed-up is about 23%, resulting in an increase of 15% to 23% in the data throughput.

The acceleration of internal speed and throughput is accomplished by making changes in the 1411 Processing Unit. The accelerated system is otherwise identical to the unchanged system. All instructions and I/O devices operate the same; however, the feature creates the 39 new timing formulas and improved timings that follow:

ADD (TWO FIELDS)

Timing: $T = 4 (L + 1 + E + A + B + RB)$.
When L = 1, E = 1; when L = 11, E = 0.

ADD (ONE FIELD)

Timing: $T = 4 (L + 1 + 2A)$.
L = 1 or 6.

SUBTRACT (TWO FIELDS)

Timing: $T = 4 (L + 1 + E + A + B + RB)$.
When L = 1, E = 1; when L = 11, E = 0.

SUBTRACT (ONE FIELD)

Timing: $T = 4 (L + 1 + 2A)$.
L = 1 or 6.

ZERO AND ADD (TWO FIELDS)

Timing: $T = 4 (L + 1 + E + A + B)$.
When L = 1, E = 1; when L = 11, E = 0.

ZERO AND ADD (ONE FIELD)

Timing: $T = 4 (L + 1 + 2A)$.
L = 1 or 6.

ZERO AND SUBTRACT (TWO FIELDS)

Timing: $T = 4 (L + 1 + E + A + B)$.
When L = 1, E = 1; when L = 11, E = 0.

ZERO AND SUBTRACT (ONE FIELD)

Timing: $T = 4 (L + 1 + 2A)$.
L = 1 or 6.

MULTIPLY

Timing: $T = 4 [L + 1 + E + 2M + (2.5M + 1) (2A + 2)]$.

DIVIDE

Timing: $T = 4 [L + 1 + E + 6.5Q (2A + 2)]$.

TIMING FOR INDEXING OPERATIONS

Timing: T = 30.67 for each single address indexed.

BRANCH UNCONDITIONAL

Timing: $T = 4 (L + 2)$.
L = 1 or 7.

TEST AND BRANCH (CONDITIONAL)

Timing: $T = 4 (L + 1 + C)$.

BRANCH IF I/O CHANNEL STATUS INDICATOR ON

Timing: $T = 4 (L + 1 + C)$.

BRANCH IF CHARACTER EQUAL

Timing: $T = 4 (L + 2.5 + C)$.
L = 1, 6, or 12.

BRANCH IF BIT EQUAL

Timing: T = 4 (L + 2.5 + C).
L + 1, 6, or 12.

BRANCH ON WORD MARK OR ZONE EQUAL

Timing: T = 4 (L + 2.5 + C).
L = 1, 6, or 12.

MOVE INSTRUCTIONS

Timing: T = 4 (L + 1 + A + B).
L = 1, 6, or 12.

MOVE CHARACTERS AND SUPPRESS ZEROS

Timing: T = 4 (L + 1 + 4A).
L = 1, 6, or 11.

COMPARE

Timing: T = 4 (L + 1 + A + B).
L = 1, 6, or 11.

TABLE LOOKUP

Timing: T = 4 (L + 1 + B +NA).
L = 1, 6, or 12.

MOVE CHARACTERS AND EDIT

Timing: T = 4 (L + 1 + A + 1.5B + 1.5Z + 1.5D).
L = 1, 6, or 11.

STORE ADDRESS REGISTER

Timing: T = 4 (L + 8.5) = 62 microseconds.
Note: This instruction cannot be indexed.

SET WORD MARK

Timing: T = 4 (L + 4).
L = 1, 6, or 11.

CLEAR WORD MARK

Timing: T = 4 (L + 4).
L = 1, 6, or 11.

CLEAR STORAGE

Timing: T = 4 (L + 1 + B).
L = 1 or 6.

CLEAR STORAGE AND BRANCH

Timing: T = 4 (L + 2 + B).
L = 11.

HALT

Timing: T = 4.

HALT AND BRANCH

Timing: T = 32.

NO OPERATION

Timing: T = 4 (L + 1).
L = 1, 2, 3 . . . no limit

READ A CARD*

Timing: T = 44 + I/O.

SELECT STACKER AND FEED*

Timing: T = 12 + I/O.

PUNCH A CARD*

Timing: T = 44 + I/O.

WRITE A LINE†

Timing: T = 44 + I/O.

WRITE WORD MARKS†

Timing: T = 44 + I/O.

CONTROL CARRIAGE

Timing: T = 12.

UNIT CONTROL

Timing: T = .0040 (L + 1) + Tm ms.

READ OR WRITE TAPE

Timing: T = .0040 (L + 1) + Tm ms.

READ OR WRITE TAPE WITH WORD MARKS

Timing: T = .0040 (L + 1) + Tm ms.

*For I/O see "IBM 1402, Model 2, Timing Considerations."

†For I/O see "IBM 1403 Printer, Timing Considerations."

IBM 1401-1410 Compatibility

The IBM 1410 Data Processing System will run many programs originally written for the IBM 1401 Data Processing System. Standard 1401 system units (1401, 1402, 1403, 1405, 729, 7330) and many 1401 system special features are fully utilized by a 1410 system operating in the 1401 mode. These 1401 features are:

1. Additional Storage (10K 1410 system operates as an 8K 1401 system; all other 1410 systems operate as a 16K 1401 system)
2. Multiply-Divide
3. Expanded Print Edit
4. Advanced Programming (includes Indexing, Store Address Register, and Move Record)
5. Print Storage
6. High-Low-Equal Compare
7. Read-Punch Release
8. Sense Switches

Differences in 1401 and 1410 system operation may entail some 1401 program modification before 1401 programs can be run on the 1410 system. The extent of program compatibility can best be defined by noting the differences between the two systems.

Special Features

The 1410 is not compatible with a 1401 program that makes use of the following special features:

1. Column Binary Feature
2. Compressed Tape Feature
3. Punch Feed Read Feature
4. Serial I/O Adapter, including provisions for attaching the following units:
   a. IBM 1009 Data Transmission Unit
   b. IBM 1011 Paper Tape Reader
   c. IBM 1012 Paper Tape Punch
   d. IBM 1412 Magnetic Character Reader
   e. IBM 1418 Optical Character Reader
   f. IBM 1419 Magnetic Character Reader
5. Process Overlap Feature
6. Selective Tape Listing Feature
7. Space Suppression Feature

Processing

INPUT CHARACTERS HAVING INCORRECT PARITY

The 1401 system corrects any input character of incorrect parity by adding or removing the C bit, forcing the character valid. The 1410 system inserts an asterisk (*) in core storage in place of all input characters having incorrect parity.

QUOTIENT BLANKS

On a 1401 system divide operation, if the B field initially contained blanks, and if the resulting quotient is zero, the blanks remain in the quotient.

A 1410 system operating in the 1401 mode converts all blanks of this type to zeros.

Magnetic Tape Operations

LOADING TAPE UNITS

On the 1401, it is possible to load a tape reel by pressing the reset, load rewind, and start keys in rapid succession. The program can then be started immediately (though this is not recommended); if it addresses the tape unit before tape loading has been completed, the program will wait until the tape is fully loaded and then proceed.

On the 1410 operating in 1401 mode, the identical procedure, and premature tape addressing, will prevent the tape drive from starting. Operations cannot continue until the console stop key is pressed, an address-set is made on the instruction to that tape unit, and the console start key is pressed. To avoid this condition, simply do not press the tape unit start key until the loading operation has been completed. Also, no problem will occur if the program is halted at a halt instruction and the program is not started, by pressing the console start key, until the tape load operation is completed.

READ AND WRITE TAPE WITH WORD MARKS

When writing on tape in the load mode, if a word separator is encountered in core storage, the 1401 will write one word separator on tape, but the 1410 will write two word separators on tape.

When reading tape in the load mode on the 1401, any number (one or more) of word separators read in succession from tape are eliminated and a word mark is placed over the first non-word separator character that follows the word separators.

When reading tape in the load mode on the 1410, a pair of adjacent word separators on tape are read into core storage as one word separator and no word mark is placed over the next non-word separator character.

Card and Print Operations

B-ADDRESS REGISTER

At the completion of a card or print operation, the setting of the B-address register will be different on a 1401 than on a 1410 operating in the 1401 mode. The following chart shows a comparison of the B-address register contents at the completion of the specified operation:

1401 OP CODE	B-ADDRESS REGISTER OF 1401	B-ADDRESS REGISTER OF 1410 (1401 MODE)
1 (Read)	081	082
2 (Print)	333*	335
3 (Read and Print)	081	082
4 (Punch)	181	183
5 (Read and Punch)	181	183
6 (Print and Punch)	181	183
7 (Read, Print, and Punch)	181	183

*335 for an unbuffered printer.

CARD READ-PUNCH CHARACTER SET

The 1410 system punches on A bit in core storage (ƀ, formerly ¢) as an 8-2 combination in a card column and reads an 8-2 combination in a card column as an A bit.

The 1401 system punches an A bit in core storage as a zero in a card column and reads an 8-2 combination in a card column as an invalid character.

A no-charge RPQ* (898148) makes the 1401 operate as described for the 1410.

CARD READ-PUNCH OPERATIONS

The 1401 puts a "set-up" character in core storage after a read or punch instruction. A punch instruction leaves a zero in core storage position 100 and any read instruction leaves an ampersand (&) in core storage position 000. A 1410 system operating in the 1401 mode does not do this.

Cards punched with an MLP (multiple line printing) code can be read by the 1401 system but cause a

*Request for Price Quotation

validity check in the 1402 Card Read Punch used with the 1410 system. The effect depends on the mode: in the 1401 mode, the validity check is recognized as a reader check by the 1411; in the 1410 mode, the validity check sets the data check channel status indicator.

CARD POSITION WHEN SYSTEM STOPS

Because the 1410 makes use of a buffer on data transfers between core storage and the card reader or punch, the 1401 read release and punch release instructions do not cause actual card-feed motion on the 1410 system operating in the 1401 mode. Following a system stop operation, the position of the cards in the hoppers and stackers of an IBM 1402 may not be the same in a 1410 system as they are in a 1401 system.

CARD STACKER SELECTION

In a 1401 system, about 10 milliseconds of computer time are available at the end of a card read for starting a stacker select operation. For a 1410 operating in the 1401 mode, the amount of time available to start a stacker select operation may vary from 8 to 82 milliseconds. No incompatibilities arise unless:

1. The program between the read and stack instructions takes longer (up to 2 milliseconds possible) than the time available on the 1410 system. This results in a stack instruction that is effective on a 1401 system but is too late to be effective on a 1410 system. Because of the faster internal processing of a 1410, this result is unlikely.
2. A stack instruction in a 1401 program is effective on a 1410 system but is too late to be effective on a 1401 system. Because the 1410 has faster internal processing and usually has a longer available time to start a stack instruction, this problem may result if 1401 programs are tested and debugged on a 1410 system in 1401 mode.

51-COLUMN CARDS

When the 51-column read feed feature is installed on the 1402, and 51-column cards are being read in either the move or load mode, the card data are always read into core storage locations 15 through 65. The 1401 leaves the storage locations 1 through 14 and 66 through 80 undisturbed. In the 1401 mode of the 1410 system, however, the positions 1 through 14 and 66 through 80 in the read buffer are filled with valid blanks, and these blanks are transferred* as blanks to the same core storage positions. Consequently, the program must not place data in those locations, or must move it elsewhere in core storage before a card is read.

PRINTER CARRIAGE CONTROLS

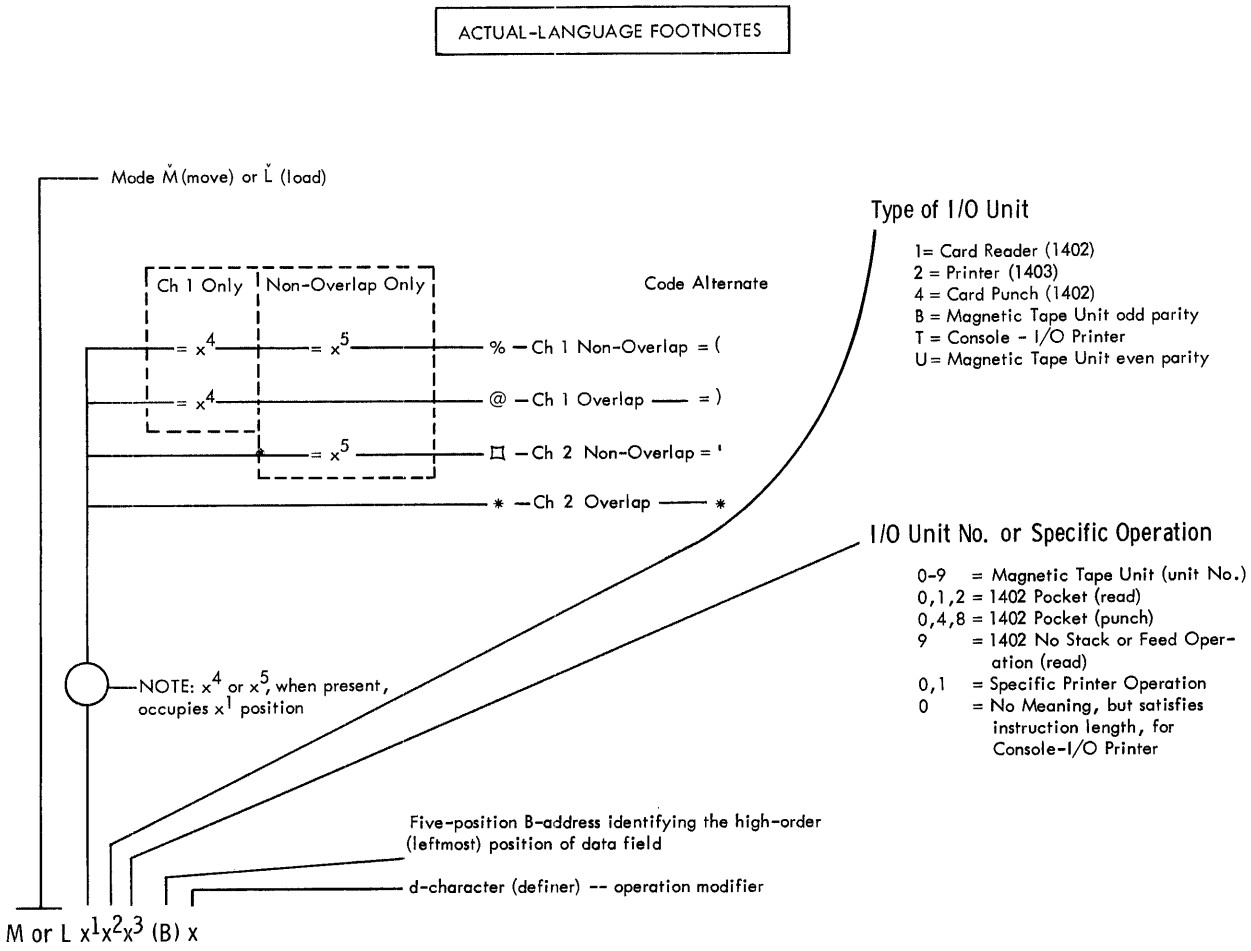
A 1401 system will not execute an immediate skip to channel X when the carriage is already at that channel, but a 1410 operating in the 1401 mode will execute the instruction by moving to the next punch of that same channel number.

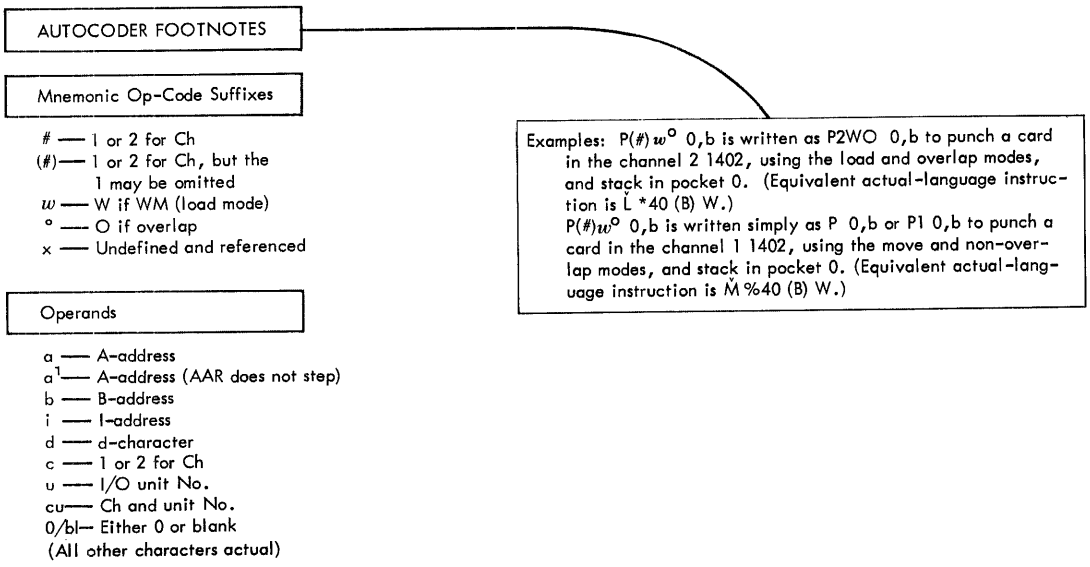
SINGLE CHARACTER EDIT

If the specified A-field in an edit operation contains a word mark in the units position (a single-character field), the 1401 system will not transfer this single-character field to the B-field.

The 1410 will transfer and edit the single-character field.

*In the 1410 mode, only the 51 active positions are transferred from buffer to storage, and a ǂ is needed in the 52nd position of the read-in area. In the 1401 mode, all 80 positions, including blanks, are transferred, and no ǂ is required.





● Figure 107. Footnotes from Appendix

Appendix

Alphabetic Listing of 1410 Instructions (Indexed)

INSTRUCTION	AUTOCODER	ACTUAL	PAGE
Add (One Field)	A a	Ǎ (A)	14
Add (Two Fields)	A a,b	Ǎ (A) (B)	14
Backspace Tape	BSP cu	Ǔ $x^5x^2x^8$ B	83
Branch if Arithmetic Overflow	BAV i	J̌ (I) Z	21
Branch if Bit Equal (any bit in b matches a bit in d)	BBE i,b,d	W̌ (I) (B) x	23
Branch if Carriage Busy (Ch 1)	BPCB(#) i	J̌ (I) R	21
Branch if Carriage Busy (Ch 2)	BPCB2 i	J̌ (I) L	21
Branch if Carriage 9 (Ch 1)	BC9(#) i	J̌ (I) 9	21
Branch if Carriage 9 (Ch 2)	BC92 i	J̌ (I) !	21
Branch if Carriage Overflow, 12 (Ch 1)	BCV(#) i	J̌ (I) @	21
Branch if Carriage Overflow, 12 (Ch 2)	BCV2 i	J̌ (I) □	21
Branch if Any I/O Channel Status Indicator On (Ch 1)	BA1 i	Ř (I) ⋕	22
Branch if Any I/O Channel Status Indicator On (Ch 2)	BA2 i	X̌ (I) ⋕	22
Branch if I/O Unit Not Ready (Ch 1 or 2)	BNR# i	Ř or X̌ (I) 1	22
Branch if I/O Unit Busy (Ch 1 or 2)	BCB# i	Ř or X̌ (I) 2	22
Branch if I/O Unit Data Check (Ch 1 or 2)	BER# i	Ř or X̌ (I) 4	22
Branch if I/O Unit Condition (Ch 1 or 2)	BEF# i	Ř or X̌ (I) 8	22
Branch if I/O Wrong Length Record (Ch 1 or 2)	BWL# i	Ř or X̌ (I) —	22
Branch if I/O Unit No Transfer (Ch 1 or 2)	BNT# i	Ř or X̌ (I) ƀ	22
Branch if Any On in Plural Indicator Test (by d-char of more than one but less than all bits, such as ⋕)	BEX# i,d	Ř or X̌ (I) x	22
Branch if Character Equal (b = d)	BCE i,b,d	B̌ (I) (B) x	23
Branch if Compare Equal	BE i	J̌ (I) S	21
Branch if Compare High (B greater than A)	BH i	J̌ (I) U	21
Branch if Compare Low (B less than A)	BL i	J̌ (I) T	21
Branch if Compare Unequal	BU i	J̌ (I) /	21
Branch if Divide Overflow	BDV i	J̌ (I) W	21
Branch if Inquiry Request (Ch 1)	BNQ(#) i	J̌ (I) Q	21
Branch if Inquiry Request (Ch 2)	BNQ2 i	J̌ (I) *	21
Branch if Overlap in Process (Ch 1)	BOL1 i	J̌ (I) 1	21
Branch if Overlap in Process (Ch 2)	BOL2 i	J̌ (I) 2	21
Branch if Zero Balance	BZ i	J̌ (I) V	21
Branch if WM Present	BW i,b	V̌ (I) (B) 1	23
Branch if WM Present, or Zone Bits Absent	BWZ i,b	V̌ (I) (B) 3	23
Branch if WM Present, or Zone Equal A	BWZ i,b,A	V̌ (I) (B) T	23
Branch if WM Present, or Zone Equal AB	BWZ i,b,AB	V̌ (I) (B) C	23
Branch if WM Present, or Zone Equal B	BWZ i,b,B	V̌ (I) (B) L	23
Branch if Zone Bits Absent	BZN i,b	V̌ (I) (B) 2	23
Branch if Zone Equal A	BZN i,b,A	V̌ (I) (B) S	23
Branch if Zone Equal AB	BZN i,b,AB	V̌ (I) (B) B	23

INSTRUCTION	AUTOCODER	ACTUAL	PAGE
Branch if Zone Equal B	BZN i,b,B	V (I) B) K	23
Branch Unconditionally	B i	J (I) blank	21
Carriage Control Immediate Skip to 1 (Ch 1 or 2)	CC(#) 1	F or 2 1	79
Carriage Control Immediate Skip to 2 (Ch 1 or 2)	CC(#) 2	F or 2 2	79
Carriage Control Immediate Skip to (3-10) (Ch 1 or 2)	CC(#) (3-0)	F or 2 3-0	79
Carriage Control Immediate Skip to 11 (Ch 1 or 2)	CC(#) #	F or 2 #	79
Carriage Control Immediate Skip to 12 (Ch 1 or 2)	CC(#) @	F or 2 @	79
Carriage Control Immediate 1 Space (Ch 1 or 2)	CC(#) J	F or 2 J	79
Carriage Control Immediate 2 Spaces (Ch 1 or 2)	CC(#) K	F or 2 K	79
Carriage Control Immediate 3 Spaces (Ch 1 or 2)	CC(#) L	F or 2 L	79
Carriage Control Skip After Print to (1-5) (Ch 1 or 2)	CC(#) (A-E)	F or 2 A-E	79
Carriage Control Skip After Print to (6-7) (Ch 1 or 2)	CC(#) (F-G)	F or 2 F-G	79
Carriage Control Skip After Print to (8-9) (Ch 1 or 2)	CC(#) (H-I)	F or 2 H-I	79
Carriage Control Skip After Print to 10 (Ch 1 or 2)	CC(#) ?	F or 2 ?	79
Carriage Control Skip After Print to 11 (Ch 1 or 2)	CC(#) .	F or 2 .	79
Carriage Control Skip After Print to 12 (Ch 1 or 2)	CC(#) □	F or 2 □	79
Carriage Control 1 Space After Print (Ch 1 or 2)	CC(#) /	F or 2 /	79
Carriage Control 2 Spaces After Print (Ch 1 or 2)	CC(#) S	F or 2 S	79
Carriage Control 3 Spaces After Print (Ch 1 or 2)	CC(#) T	F or 2 T	79
Clear Storage	CS b	/ (B)	37
Clear Storage and Branch	CS i,b	/ (I) (B)	37
Clear Word Mark (One Address)	CW a	□ (A)	36
Clear Word Mark (Two Addresses)	CW a,b	□ (A) (B)	36
Compare (b to a)	C a,b	C (A) (B)	28
Data Move (see Move)	Mxxxx a,b	D (A) (B) x	25
Divide (a into b)	D a,b	% (A) (B)	17
Erase Forward (Skip and Blank Tape)	SKP cu	U $x^5x^2x^3$ E	83
Halt	H	.	37
Halt and Branch	H i	. (I)	38
Lookup Equal	LE a,b	T (A) (B) 2	30
Lookup Equal or High	LEH a,b	T (A) (B) 6	30
Lookup High	LH a,b	T (A) (B) 4	30
Lookup Low	LL a,b	T (A) (B) 1	30
Lookup Low or Equal	LLE a,b	T (A) (B) 3	30
Lookup Low or High	LLH a,b	T (A) (B) 5	30
Lookup to Any		T (A) (B) 7	30
Lookup to End		T (A) (B) blank	30
Move Characters and Edit	MCE a,b	E (A) (B)	31
Move Characters and Suppress Zeros	MCS a,b	Z (A) (B)	27
Move Left Characters and WM Single Position	MLCWS a,b	D (A) (B) 7	25
Move Left Characters and WM thru 1st A-Field WM	MLCWA a,b	D (A) (B) X	25
Move Left Characters and WM thru 1st B-Field WM	MLCWB a,b	D (A) (B) P	25
Move Left Characters and WM thru 1st WM	MLCW a,b	D (A) (B) G	25
Move Left Characters Single Position	MLCS a,b	D (A) (B) 3	25
Move Left Characters thru 1st A-Field WM	MLCA a,b	D (A) (B) T	25

INSTRUCTION	AUTOCODER	ACTUAL	PAGE
Move Left Characters thru 1st B-Field WM	MLCB a,b	D̆ (A) (B) L	25
Move Left Characters thru 1st WM	MLC a,b	D̆ (A) (B) C	25
Move Left Numeric and WM Single Position	MLNWS a,b	D̆ (A) (B) 5	25
Move Left Numeric and WM thru 1st A-Field WM	MLNWA a,b	D̆ (A) (B) V	25
Move Left Numeric and WM thru 1st B-Field WM	MLNWB a,b	D̆ (A) (B) N	25
Move Left Numeric and WM thru 1st WM	MLNW a,b	D̆ (A) (B) E	25
Move Left Numeric Single Position	MLNS a,b	D̆ (A) (B) 1	25
Move Left Numeric thru 1st A-Field WM	MLNA a,b	D̆ (A) (B) /	25
Move Left Numeric thru 1st B-Field WM	MLNB a,b	D̆ (A) (B) J	25
Move Left Numeric thru 1st WM	MLN a,b	D̆ (A) (B) A	25
Move Left WM Single Position	MLWS a,b	D̆ (A) (B) 4	25
Move Left WM thru 1st A-Field WM	MLWA a,b	D̆ (A) (B) U	25
Move Left WM thru 1st B-Field WM	MLWB a,b	D̆ (A) (B) M	25
Move Left WM thru 1st WM	MLW a,b	D̆ (A) (B) D	25
Move Left Zones and WM Single Position	MLZWS a,b	D̆ (A) (B) 6	25
Move Left Zones and WM thru 1st A-Field WM	MLZWA a,b	D̆ (A) (B) W	25
Move Left Zones and WM thru 1st B-Field WM	MLZWB a,b	D̆ (A) (B) O	25
Move Left Zones and WM thru 1st WM	MLZW a,b	D̆ (A) (B) F	25
Move Left Zones Single Position	MLZS a,b	D̆ (A) (B) 2	25
Move Left Zones thru 1st A-Field WM	MLZA a,b	D̆ (A) (B) S	25
Move Left Zones thru 1st B-Field WM	MLZB a,b	D̆ (A) (B) K	25
Move Left Zones thru 1st WM	MLZ a,b	D̆ (A) (B) B	25
Move Right Characters and WM thru 1st A-Field ǂ̆	MRCWG a,b	D̆ (A) (B) △	25
Move Right Characters and WM thru 1st A-Field ǂ	MRCWR a,b	D̆ (A) (B) ⧺	25
Move Right Characters and WM thru 1st A-Field ǂ or ǂ̆	MRCWM a,b	D̆ (A) (B) ǂ	25
Move Right Characters and WM thru 1st WM	MRCW a,b	D̆ (A) (B) ∨	25
Move Right Characters thru 1st A-Field ǂ̆	MRCG a,b	D̆ (A) (B) $	25
Move Right Characters thru 1st A-Field ǂ	MRCR a,b	D̆ (A) (B) ,	25
Move Right Characters thru 1st A-Field ǂ or ǂ̆	MRCM a,b	D̆ (A) (B) .	25
Move Right Characters thru 1st WM	MRC a,b	D̆ (A) (B) #	25
Move Right Numeric and WM thru 1st A-Field ǂ̆	MRNWG a,b	D̆ (A) (B) ⎦	25
Move Right Numeric and WM thru 1st A-Field ǂ	MRNWR a,b	D̆ (A) (B) ⁓	25
Move Right Numeric and WM thru 1st A-Field ǂ or ǂ̆	MRNWM a,b	D̆ (A) (B) [	25
Move Right Numeric and WM thru 1st WM	MRNW a,b	D̆ (A) (B) :	25
Move Right Numeric thru 1st A-Field ǂ̆	MRNG a,b	D̆ (A) (B) R	25
Move Right Numeric thru 1st A-Field ǂ	MRNR a,b	D̆ (A) (B) Z	25
Move Right Numeric thru 1st A-Field ǂ or ǂ̆	MRNM a,b	D̆ (A) (B) I	25
Move Right Numeric thru 1st WM	MRN a,b	D̆ (A) (B) 9	25
Move Right WM thru 1st A-Field ǂ̆	MRWG a,b	D̆ (A) (B) *	25
Move Right WM thru 1st A-Field ǂ	MRWR a,b	D̆ (A) (B) %	25
Move Right WM thru 1st A-Field ǂ or ǂ̆	MRWM a,b	D̆ (A) (B) □	25
Move Right WM thru 1st WM	MRW a,b	D̆ (A) (B) @	25
Move Right Zones and WM thru 1st A-Field ǂ̆	MRZWG a,b	D̆ (A) (B) ;	25
Move Right Zones and WM thru 1st A-Field ǂ	MRZWR a,b	D̆ (A) (B) \	25
Move Right Zones and WM thru 1st A-Field ǂ or ǂ̆	MRZWM a,b	D̆ (A) (B) <	25
Move Right Zones and WM thru 1st WM	MRZW a,b	D̆ (A) (B) >	25
Move Right Zones thru 1st A-Field ǂ̆	MRZG a,b	D̆ (A) (B) !	25
Move Right Zones thru 1st A-Field ǂ	MRZR a,b	D̆ (A) (B) ǂ	25
Move Right Zones thru 1st A-Field ǂ or ǂ̆	MRZM a,b	D̆ (A) (B) ?	25

INSTRUCTION	AUTOCODER	ACTUAL	PAGE
Move Right Zones thru 1st WM	MRZ a,b	D (A) (B) 0	25
Multiply	M a,b	@ (A) (B)	15
No Operation	NOP	N	38
Punch a Card, Stack in Pocket 0 (wo/w WM)	P(#)*w*° 0,b	M or L x¹40 (B) W	62
Punch a Card, Stack in Pocket 4 (wo/w WM)	P(#)*w*° 4,b	M or L x¹44 (B) W	62
Punch a Card, Stack in Pocket 8 (wo/w WM)	P(#)*w*° 8,b	M or L x¹48 (B) W	62
Read a Card, Stack in Pocket 0 (wo/w WM)	R(#)*w*° 0,b	M or L x¹10 (B) R	61
Read a Card, Stack in Pocket 1 (wo/w WM)	R(#)*w*° 1,b	M or L x¹11 (B) R	61
Read a Card, Stack in Pocket 2 (wo/w WM)	R(#)*w*° 2,b	M or L x¹12 (B) R	61
Read a Card, No Stack or Feed Operation (wo/w WM)	R(#)*w*° 9,b	M or L x¹19 (B) R	61
Read Console Printer (wo/w WM)	RCP*w*° b	M or L x⁴T0 (B) R	41
Read Tape (wo/w WM)	RT*w*° cu,b	M or L x¹Ux⁸ (B) R	85
Read Tape to IRG or End of Core (wo/w WM)	RTG*w* cu,b	M or L x⁵Ux⁸ (B) $	85
Read Tape Binary (odd parity) wo/w WM	RTB*w*° cu,b	M or L x¹Bx⁸ (B) R	85
Read Tape Binary (odd parity) to IRG or End of Core (wo/w WM)	RTBG*w* cu,b	M or L x⁵Bx⁸ (B) $	85
Rewind	RWD cu	U x⁵x²x⁸ (B) R	83
Rewind and Unload	RWU cu	U x⁵x²x⁸ (B) U	83
Scan Left Single Position	SCNLS a,b	D (A) (B) blank	26
Scan Left thru 1st A-Field WM	SCNLA a,b	D (A) (B) ℏ	26
Scan Left thru 1st B-Field WM	SCNLB a,b	D (A) (B) –	26
Scan Left thru 1st WM	SCNL a,b	D (A) (B) &	26
Scan Right thru 1st A-Field ≢	SCNRG a,b	D (A) (B) Q	26
Scan Right thru 1st A-Field ≠	SCNRR a,b	D (A) (B) Y	26
Scan Right thru 1st A-Field ≠ or ≢	SCNRM a,b	D (A) (B) H	26
Scan Right thru 1st WM	SCNR a,b	D (A) (B) 8	26
Select Stacker 0 and Feed (Ch 1 or 2)	SSF(#) 0/bl	K or 4 0	62
Select Stacker 1 and Feed (Ch 1 or 2)	SSF(#) 1	K or 4 1	62
Select Stacker 2 and Feed (Ch 1 or 2)	SSF(#) 2	K or 4 2	62
Set Word Mark (One Address)	SW a	, (A)	36
Set Word Mark (Two Addresses)	SW a,b	, (A) (B)	36
Skip and Blank Tape	SKP cu	U x⁵x²x⁸ E	83
Store A-address Register	SAR a¹	G (C) A	36
Store B-address Register	SBR a¹	G (C) B	36
Store E-address Register	SER a¹	G (C) E	36
Store F-address Register	SFR a¹	G (C) F	36
Subtract (One Field)	S a	S (A)	14
Subtract (Two Fields)	S a,b	S (A) (B)	14
Table Lookup (see Lookup)	Lxx a,b	T (A) (B) x	30
Test and Branch (see Branch)	Bxx i	J (I) x	21
Unit Control (see Backspace, Skip, Write √, Rewind)	xxx cu	U x⁵x²x⁸ x	83
Write a line	W(#)° b	M x¹20 (B) W	78
Write a line, WM Create Blanks in Printing	W(#)W° b	L x¹20 (B) W	78
Write Console Printer (wo/w WM)	WCP*w*° b	M or L x⁴T0 (B) W	41
Write Printer (see Write a Line)	W(#)*w*° b	M or L x¹20 (B) W	78

INSTRUCTION	AUTOCODER	ACTUAL	PAGE
Write Tape (wo/w WM)	WT*w*° cu,b	M or L x^1Ux^8 (B) W	85
Write Tape to End of Core (wo/w WM)	WTE*w* cu,b	M or L x^5Ux^8 (B) X	85
Write Tape Binary (odd parity) wo/w WM	WTB*w*° cu,b	M or L x^1Bx^8 (B) W	85
Write Tape Binary (odd parity) to End of Core (wo/w WM)	WTBE*w* cu,b	M or L x^5Bx^8 (B) X	85
Write Tape Mark	WTM cu	U $x^5x^2x^8$ M	83
Write Word Marks As 1's	WM(#)° b	M x^121 (B) W	79
Zero and Add (One Field)	ZA a	? (A)	15
Zero and Add (Two Fields)	ZA a,b	? (A) (B)	14
Zero and Subtract (One Field)	ZS a	! (A)	15
Zero and Subtract (Two Fields)	ZS a,b	! (A) (B)	15

Index

CUT ALONG LINE

COMMENT SHEET

IBM 1410 DATA PROCESSING SYSTEM

PRINCIPLES OF OPERATION, FORM A22-0526

FROM

NAME _____

OFFICE _____

FOLD

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

FOLD

☐ SUGGESTED ADDITION (PAGE)

☐ SUGGESTED DELETION (PAGE)

☐ ERROR (PAGE)

EXPLANATION

FOLD

FOLD

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

FOLD ON TWO LINES, STAPLE, AND MAIL

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P. O. BOX 390
POUGHKEEPSIE, N. Y.

ATTN: CUSTOMER MANUALS DEPT. B98

CUT ALONG LINE

FOLD FOLD

STAPLE STAPLE

IBM 1410 Princ. Op. Printed in U.S.A. A22-0526

IBM®

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