File Number 7010-01
Form A22-6726

IBM Systems Reference Library

IBM 7010 Principles of Operation

This manual presents the principles of operation of the IBM 7010 Data Processing System. Its purpose is to provide (1) a reference and guide for those familiar with the system and (2) an instruction aid for the development and training of programmers and operators. The material is presented in a direct manner; it is assumed that the reader is familiar with the information in *IBM 7010 System Summary,* Form A22-6724.

The manual describes the 7010 operating principles, the 7114 Processing Unit, the complete instruction set, and the 1415-2 Console with the I/O Console Printer. Details about I/O device operation are in publications listed in the *IBM 7010 Bibliography,* Form A22-6720.

The information in this manual is machine oriented; that is, the explanations of operations and illustrations of instructions and instruction use are made primarily in machine language. Publications providing detailed information on programming systems and available programs for the 7010 system are listed in the *IBM 7010 Bibliography,* Form A22-6720.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the content of this publication to:
IBM Corporation, Customer Manuals, Dept. B98, PO Box 390, Poughkeepsie, N. Y.

Contents

IBM 7010 Data Processing System

System Organization

The IBM 7010 Data Processing System handles problems and data in volumes and magnitudes that characterize the intermediate to large-scale data processing area. Outstanding capabilities and features, combined with a wide choice of system configurations, make the IBM 7010 System a powerful general-purpose data processing system.

The 7010 system is composed of a 7114 Processing Unit (which includes core storage and one or two data channels), a 1415 Model 2 Console, and optional input/output (I/O) devices and features. The possible combinations of I/O devices and features are shown in the *IBM 7010 Configuration*, Form A22-6723.

The 7114 Processing Unit contains the arithmetic and control circuitry of the system and the address registers, logic, and checking circuits associated with the flow of data in the 7010 system. The four models of the 7114 differ in the capacity of core storage housed within the unit:

Model 1 — 40,000 positions
Model 2 — 60,000 positions
Model 3 — 80,000 positions
Model 4 — 100,000 positions

The basic system has the 40,000 character position core storage unit. The 60,000, 80,000 or 100,000 character position storage units are optional features. Each storage position is individually addressable with a unique five-character address. The basic read-write cycle time of core storage is 2.4 microseconds. Each storage access allows two adjacent characters (in parallel) to be placed in storage or removed from storage for processing.

The 7010 is a stored program, variable word length system. Instructions of the stored program are assigned sequential locations in storage. The method of reading and executing these instructions is also sequential, unless the sequence is altered by an instruction.

Data and instructions are stored and processed in groups of characters called words or fields. Word or field length is defined by the word mark, a special data bit associated with a character. Characters within the system are coded in an odd parity, eight-bit, binary coded decimal form.

Valid instructions are one to twelve characters long, and the form and content of the instruction indicate the operation to be performed. The instruction set is augmented by the indexing and chaining functions.

Information is transferred between the input/output devices and core storage through the I/O data channels. Two data channels with buffered input/output operation, including high-speed printing, card reading, and punching, are available.

Processing overlap and priority processing are standard. Processing overlap allows computing to continue during most of the time taken by the transfer of data to or from input/output devices and core storage. Priority processing is an interrupt system that provides an automatic program interrupt when certain conditions of the data channels or I/O devices occur.

The 7010 system is provided with a complete library of programs and programming systems including Autocoder, Input/Output Control Systems (IOCS), Sort-Merge, and a variety of utility programs to simplify program testing and systems operation. In addition, there is complete instruction compatibility with the IBM 1410 Data Processing System. The 7010 is designed so that programs produced by IBM programming systems for the 1410 operate unaltered when input/output configurations are identical, with no distinction either in input or results.

Character Code

Alphameric characters stored and moved within the system are coded in eight-bit, binary coded decimal form (Figure 1.). Six bits (B, A, 8, 4, 2, 1) are information or data bits, which generate the standard set of 64 characters (Figure 2). The eighth bit, a word mark (WM), identifies a character used to define the starting or ending character positions of data fields and instruction fields. Word marks can be set and cleared by program instructions. The word-mark bit is symbolized, in printed form, by an inverted circumflex (v) over the character containing a word-mark bit ($\overset{v}{A}$).

The seventh bit (C), the check bit, is used to maintain odd parity. Each character must contain an odd number of bits, called parity, for that character. If

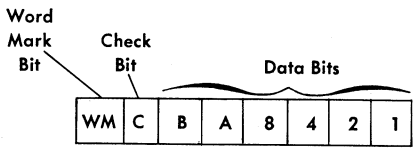


Figure 1. Character Code Bits

	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	
	ǂ		0-7-8			A	8	4	2	1
(2) –	ƀ		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	

	CHARACTER		CARD CODE	BCD CODE (Core Storage)						
	Commerce (Report)	Science (Program)								
	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

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 bit, the word mark bit is counted in the character configuration before the bit status is determined.

Each character is checked at various locations in the system to be sure that the total number of bits, including the check bit and word mark bit, is odd. The following examples show the the bit configuration of characters with and without word marks.

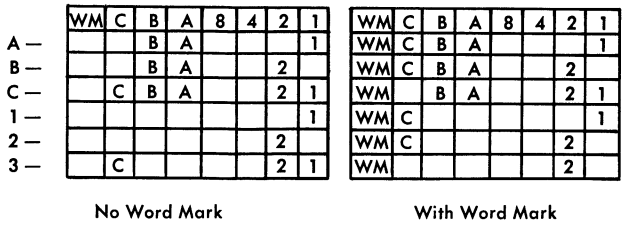


Figure 2 shows the 64 character codes valid in the 7010 and their character or graphic equivalents. The chart is arranged in ascending collating sequence, with a blank having the lowest collating number of 00 and a nine having the highest of 63. Figure 2 also illustrates the IBM Card Code. Figure 3 lists the names of the symbols used in the character set.

The system can read, punch, or type out on the 1415-2 Console I/O Printer, any of the characters symbolizing these 64 character codes. Forty-eight of the the characters will print on an IBM 1403 Printer having an alphameric chain.

Five of the character codes print out either of two 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 A is available for report writing and most commercial uses, and print arrangement H 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 A is being used. The console-I/O printer head type in use corresponds with the print arrangement installed in the 1403 printer.

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

Figure 3. Symbol Names

Core Storage

Core storage is located in the 7114 Processing Unit. Storage capacity of 40,000, 60,000, 80,000, or 100,000 character positions is available. Each storage position is individually addressable with a unique five-character address. Valid addresses range from 00000 to 39000, 59999, 79999, or 99999, depending on the size of the storage unit. During execution of an instruction, no storage address may be decremented below 00000 or incremented past the highest valid address for the storage unit. This programming error causes the system to stop, and an address check is indicated.

Only five-character decimal addresses are valid. Numeric bit configurations greater than 9 (for example: 8-3, 8-4, 8-5, 8-6 and 8-7) are invalid. An address with zone bits in the units, thousands, or the ten-thousands position is invalid. This programming error causes a system stop, and an address check is indicated.

The basic read-write cycle time of core storage is 2.4 microseconds. Each storage access allows two adjacent characters (in parallel) to be read from storage for processing and two characters to be written in storage. The read-write cycle time may be automatically interrupted after the read portion for an indefinite time before the write portion is initiated, allowing variable machine cycle lengths. All control and alignment functions peculiar to the two-character parallel storage are automatically performed.

No particular areas of storage are reserved exclusively for the program; the location of instructions, constants, or data to be processed is entirely at the programmer's discretion.

Instruction Form

Valid instructions are one to twelve characters long. The form and content of the instruction indicate the operation to be performed.

The basic instruction form has four parts: operation code, A-address or x-control field, B-address, and d-character modifier to the operation code (Figure 4).

Part 1	Part 2	Part 3	Part 4
Op Code	A-Address or x-Control Field	B-Address	d-Character
O	aaaaa or xxx	bbbbb	d

Figure 4. 7010 Instruction Form

The Operation Code is a single character that specifies the basic machine operation to be performed.

The A-Address is a five-character core storage address. In some operations, the A-address is identified as an I- or C-address.

The x-Control Field is three characters long and specifies the channel and I/O unit involved in an input/output operation.

The B-Address is a five-character core storage address.

The d-Character specifies a particular operation within the control of the operation code of the instruction.

Instruction forms consist of a single-character operation code or the operation code with one or more of the other three parts. Valid instruction lengths and forms are:

O	Oaaaaad
Od	Oxxxbbbbbd
Oxxxd	Oaaaaabbbbb
Oaaaaa	Oaaaaabbbbbd

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

Each instruction to be executed 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 an operation code of the next sequential instruction (Figure 5).

Instructions are arranged in core storage in sequential order from left to right. Successive instructions are placed in higher numbered storage positions (Figure 5). The address of an instruction for execution is the storage location of the operation code.

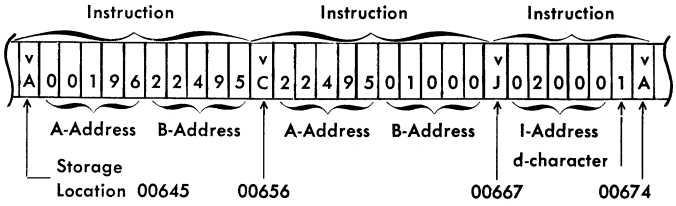


Figure 5. Instruction Word Marks

The 7010 instructions are a double-address type; that is, a single instruction can address two data fields or two different characters. In arithmetic operations, for example, one instruction addresses both factors, such as the addend and augend or the multiplier and multiplicand (Figure 5). Double addressing is also characteristic of data moving and most other operations, but instructions with single addresses are also used — in some cases, optionally; in others, necessarily.

The 7010 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.

Data Flow and Address Control

Figure 6 is a simplified schematic of data flow and storage address controls of the 7114 Processing Unit. The direction of data flow and the action of the registers, lines, and channels are automatically controlled by the instruction to be executed.

B-Data Register

The B-data register holds the two characters read from storage during the read portion of the storage read-write cycle. From the B-data register, the pair of characters may be routed into the processor and/or regenerated directly back into storage. The two core storage positions from which data are removed and into which data are placed are controlled by a five-digit address in the memory address register (MAR). Control circuitry in the processor determines if only one or both characters of a pair is to be processed during any given cycle and correctly sequences the character or characters to the B-channel.

Operation, Operation Modifier, and I/O Channel Select Registers

During instruction read-out, the operation code, d-modifier and x-control field characters are placed in their respective registers, via the B-channel, for control of the operation. Any A- or B-addresses are routed to the A-address and B-address registers respectively.

A-Data Register

The A-data register is a three-character-position register used to store and sequence A-field characters of an operation while the corresponding B-field characters are being read from storage. The A-registers are set only from the B-channel. The A-register characters are

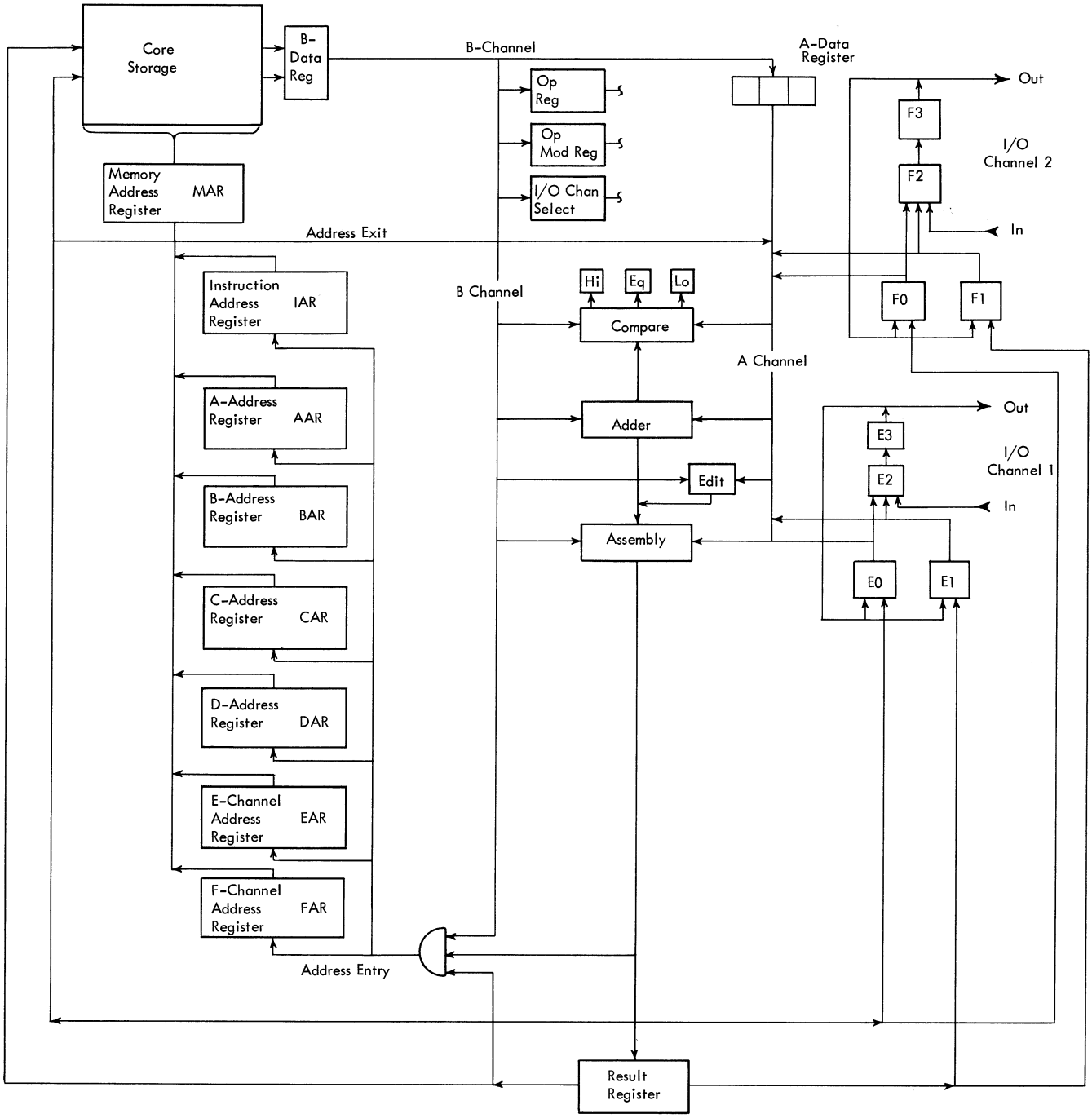


Figure 6. 7114 Logic Schematic — Data Flow and Address Control

routed in correct sequence onto a single-character data channel called the A-channel.

Adder and Compare Units

The adder is a serial decimal adder receiving data from the A-channel and B-channel through independent adder entry lines. True and complement controls are provided at both entries so that either A- or B-data may be complement added with no lost time. Adder output proceeds through the assembly register to the result register and may be switched to storage from either.

The compare unit also has independent entries from the A- and B-channels. In certain types of comparisons, the compare unit utilizes information from the adder.

Assembly and Result Registers

The assembly register is used to combine the proper portions of data received from the A-channel, B-channel, adder output, and edit output into a single character of correct parity. Some nonarithmetic data manipulation functions are also performed in the assembly register. Assembly register output may be switched either to storage, the result register, address entry, or an I/O channel register.

The single-character result register is used to temporarily store one of two successive characters involved in processing. The output of the result register may be switched to storage, address entry, or an I/O register.

Input/Output Channel Registers

Four input/output channel registers, E_0, E_1, E_2, and E_3 are used in the input/output operations on channel 1. Input characters are serially entered into E_2, set into E_3, and then placed into either E_0 or E_1 depending upon the sequence. From E_0 and E_1, the two characters are sequenced into A-channel to the assembly register. From the assembly register, the characters are sent directly to storage or to the result register and then to storage; the specific path taken depends on the sequencing controls.

Output characters are sequentially placed in E_0 and E_1 from either the assembly or result register. From E_0 and E_1, the characters are serially moved through E_2 to E_3 and out on I/O channel 1.

The input/output registers for channel 2 are F_0, F_1, F_2, and F_3. The data path and function of these registers for I/O operations on channel 2 are identical to those of channel 1.

Memory Address Register

The memory address register (MAR) is used exclusively to address storage; that is, it determines where data are to be read from or written into storage. The program has no access to or control of MAR. Address data are automatically set into MAR from one of the address registers.

Instruction Address Register

The instruction address register (IAR) is, in effect, an instruction counter; it keeps track, character by character, of the current address of the stored program. Like MAR, it is automatically controlled and is not directly accessible to the program. The setting of IAR is changed by one of four methods:

1. An address may be loaded into the IAR from the console (for instance, the starting address of a program).
2. On execution of a programmed branch, the branch-to address automatically replaces the current program address of the IAR.
3. Program reset or computer reset automatically loads 00001 into the IAR.
4. During normal programmed operation, the IAR is modified during instruction read-out cycles to the address of the next character of the stored program.

A- and B-Address Registers

The five-character A-address registers (AAR) and B-address registers (BAR) are loaded during instruction read-out with the addresses of the A and B operands for internal instructions. The AAR can also contain the branch-to address during branch instructions; the BAR is loaded with the last previous contents of the IAR upon execution of a successful branch. The BAR is also used for the address of the input or output field in unoverlapped I/O operations. The contents of AAR and BAR are accessible to the programmer by use of the store address register instruction, and both may be loaded from the console.

C- and D-Address Registers

System controls employ these two registers for auxiliary address storage and manipulation. The C-address register is used in store address register instructions to identify the storage locations into which the specified address is stored. The contents of the C- and D-address registers are not otherwise accessible to the program.

E- and F-Channel Address Registers

The five-character E-channel address register (EAR) is used for overlapped I/O data transfers on I/O channel 1. It is loaded during instruction read-out with the high-order (leftmost) location of the I/O field and is automatically modified during I/O cycles. The EAR may be loaded from the console and is accessible to the programmer by use of the store address register instruction. The five-character F-channel address register (FAR) is identical in operation to EAR, except it is related to I/O channel 2 instead of I/O channel 1.

Chaining

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 automatically used to specify the A- and B-fields. Connecting instructions together in this manner is called chaining.

Instruction chaining saves computer time because the address registers do not have to be reloaded dur-

ing 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. After the operation: the A-address register contains 05980 (is decreased five positions), and the B-address register contains 06979 (is decreased six positions). Therefore 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. 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.

Basic considerations for chaining are:

1. Only storage addresses can be chained.
2. All no-address instructions (op code only) use the contents of the A- and B-address registers after the prior operation.
3. Branch instructions have the following effects: If the branch occurs, the B-address register is loaded with, and retains, the address of the next instruction in normal sequence, and the address of the next program step is taken from the A-address register instead of from the I-address register. If no branch occurs, the next instruction address is taken as usual from the I-address register. The instructions – branch if character equal, branch if bit equal, and branch on word mark or zone equal—have B-addresses. If these instructions do not branch, the B-address register contains the B-address minus one. This allows the next position to be given the same test, without specifying a B-address until either the register 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. The A-address register contains the given A-address.

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 10 shows the symbols that represent the contents of the address registers.

Indexing

The 7010 system has 15 five-digit indexing fields (registers) in core storage (Figure 7). Any address of any instruction, except store address register, G(C)d, and x-control fields may be indexed.

The index factor of the selected index register 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. Thus, the instruction in storage 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.

To modify addresses, the index register containing the index factor must be selected. To do this, the A-address, B-address, or both 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 8).

During an indexing operation, the tagged instruction address is considered to be positive. The sign position of the index factor is considered in one of two ways. If it has a plus sign, it is added to the tagged address. If it has a minus sign, it 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 40K, 60K, 80K and 100K 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

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 7. Index Register Location

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 8. Zone Bits Used to Tag Index Registers

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 9. Timing Formula Symbols

because they do not affect the indexing operation. 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

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 8). 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

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
Effective instruction:	A	00459	01534

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

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

Both the A- and B-addresses are tagged by an A-bit over the hundreds position (V = A41 and W = A42).

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 10. Address Register Symbol Chart

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 V	A-address	B-address
Effective instruction:	S	00651	00750

Timing for indexing operations:

T = 9.6 microseconds for each single address indexed.

Instruction Descriptions

Instructions are described in this manual 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 Title: The name and mnemonic of the operation. Mnemonics used for input/output instructions will be for data channel 1 only.

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

Function: The description of the operation performed.

Word Marks: The information the programmer must have to determine the effect of word marks on the operation.

Timing: The formula used to calculate the time required for executing the instruction (in microseconds). See Figure 9 for symbols used in the formulas.

Notes: Special notations or additional information that pertain to the operation.

Address Registers after Operation: The contents of the address registers, represented by the abbreviations shown in Figure 10.

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 7010 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 11).

When the sign of a field is changed as the result of an arithmetic operation, the machine method of signing a field is B- and A-bits (12 zone) for plus and B-bit (11 zone) for minus. Figure 12 shows the sign of the result for add and subtract operations.

Digit Coding

In all arithmetic operations, the codes: 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.

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. A test and branch instruction, $\overset{v}{J}$(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 can be tested by a test and branch instruction – $\overset{v}{J}$(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.

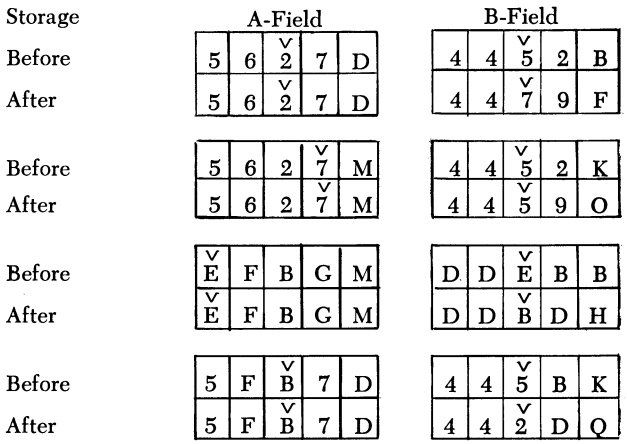
Arithmetic Operation Codes

Add (Two Fields) A

Instruction Form: A(A)(B)

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). A-field zone bits (except for the sign position) are ignored.

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.



$$\textit{Timing: } T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4E + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 3.2\left(\frac{RB+1}{2}\right)$$

Address Registers after Operation:

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

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 11. Bit Equivalents for Signs

Add (One Field) A

Instruction Form: A(A)

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. The sign bit configuration of the result is always the same as the original sign of the A-field.

Storage	A-Field							
Before	5	8	1	2	3	4	5	F
After	5	8	2	4	6	9	1	B

Before	E	H	A	B	C	D	E	O
After	E	H	B	D	F	I	A	K

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

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right) + 5.6\left(\frac{A+1}{2}\right)$

Address Registers after Operation:

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

Subtract (Two Fields) S

Instruction Form: S(A)(B)

Function: The numeric data in the A-field is 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 also 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.

Storage	A-Field					B-Field				
Before	5	6	2	7	4	4	4	5	2	2
After	5	6	2	7	8	4	4	2	4	8
Before	5	6	2	7	M	D	D	F	B	K
After	5	6	2	7	M	D	D	C	D	Q
Before	5	6	2	7	M	D	D	F	B	B
After	5	6	2	7	M	D	D	F	I	F
Before	5	6	2	7	4	4	4	5	2	K
After	5	6	2	7	4	4	0	7	9	O

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4E + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 3.2\left(\frac{RB+1}{2}\right)$

Address Registers after Operation:

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

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

Figure 12. Sign of Result for Add and Subtract Operations

Subtract (One Field) S

Instruction Form: S(A)

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.

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

Storage	A-Field							
Before	B	1	2	3	4	5	6	7
After	B	0	0	0	0	0	0	0
Before	B	1	2	C	D	5	6	P
After	B	0	0	?	?	0	0	!

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right) + 5.6\left(\frac{A+1}{2}\right)$

Address Registers after Operation:

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

Zero and Add (Two Fields) ZA

Instruction Form: ?(A)(B)

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. The final B-field sign will be in standard form. All other zone positions in the B-field are set to no-zone (no A- and no B-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, high-order positions of the A-field that exceed the limits imposed by the B-field word mark are not processed.

Storage	A-Field					B-Field				
Before	5	8	A	B	6	6	6	8	5	3
After	5	8	A	B	6	6	6	1	2	F
Before	4	9	6	5	D	8	5	A	B	C
After	4	9	6	5	D	0	0	6	5	D

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4E + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 3.2\left(\frac{RB+1}{2}\right)$

Address Registers after Operation:

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

Zero and Add (One Field) ZA

Instruction Form: ?(A)

Function: This form of the zero and add instruction is used to strip the A-field of all zones, except in the units (sign) position, and to change non-numeric codes (blanks, 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; however, the bit configuration of the plus sign may change. If the A-field plus sign bit configuration is not an A- and B-bit, it is changed to the A- and B-bit configuration.

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

Storage	A-Field							
Before	A	B	C	D	E	F	G	H
After	1	2	3	4	5	6	7	H
Before	$	*	[	‡	S	T	5	3
After	3	4	5	0	2	3	5	C

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right) + 5.6\left(\frac{A+1}{2}\right)$

Address Registers after Operation:

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

Zero and Subtract (Two Fields) ZS

Instruction Form: !(A)(B)

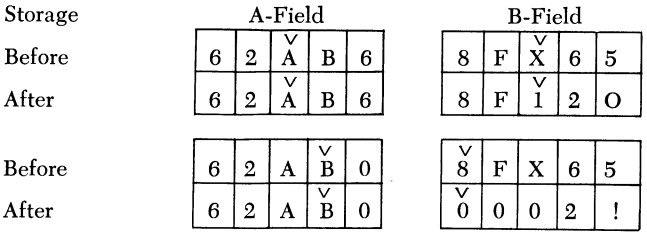
Function: This operation causes the numeric data in the A-field to be moved and stored in the B-field with the opposite sign (Figure 13). The standard machine method of signing a field is used. All other zone positions in the B-field are set to no-zone (no A- and no B-bits). If the A-field is shorter than the B-field, extra high-order B-field positions are set to zero.

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 13. Sign Changes for Zero and Subtract (Two Fields)

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 case, the system inserts zeros in the extra high-order posi-

tions 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 are not processed.



$$\textit{Timing: } T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4E + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 3.2\left(\frac{RB+1}{2}\right)$$

Address Registers after Operation:

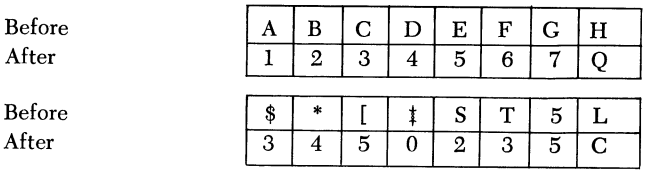
I-address Reg	A-address Reg	B-address Reg
NSI	A-LW	B-LB

Zero and Subtract (One Field) ZS

Instruction Form: !(A)

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 after the operation, and vice versa. The standard machine method of signing a field is used. Non-numeric codes are replaced by their corresponding numeric equivalents. See "Zero and Add."

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



$$\textit{Timing: } T \approx 2.4\left(\frac{L+2}{2}\right) + 5.6\left(\frac{A+1}{2}\right)$$

Address Registers after Operation:

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

Multiply M

Instruction Form: @(A)(B)

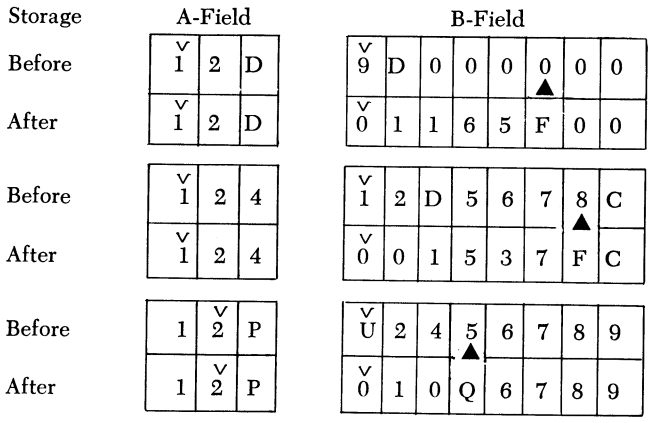
Function: The multiply instruction causes the numeric data in the A-field (multiplicand) to be multiplied by the numeric data (multiplier) in the high-order positions of the B-field. The product is developed in the B-field, eliminating the multiplier. If the multiplier is required for later use in the program, it must be retained in another storage area.

The units position of the multiplicand (A-field) is indicated by the A-address. The units position of the (multiplier-product) B-field is indicated by the B-address. The high-order position of the B-field and, consequently, the high-order position of the multiplier is located to the left of the units position the number of positions equal to 1 plus the number of digits in the multiplicand and the multiplier.

Zones that appear in the multiplicand are not disturbed by the multiply operation. At the beginning of the operation, the B-field is cleared to zeros up to the units position of the multiplier. Zones that appear in the multiplier are eliminated during product development.

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 indicated in the units position of the B-field.

Word Marks: Word marks must be set to identify the high-order positions of both the multiplier and multiplicand fields.



$$\textit{Timing: } T \approx 2.4\left(\frac{L+2}{2}\right) + (2.5M+1)\left[5.6\left(\frac{A+2}{2}\right) + 6.4\right]$$

Address Registers after Operation:

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

Divide D

Instruction Form: %(A)(B)

Function: During a divide operation the dividend located in the dividend field (low-order portion of the B-field) is divided by the divisor located in the A-field. The quotient is developed in the quotient field (high-order portion of the B-field). The remainder is placed in the dividend field, eliminating the original dividend (Figure 14).

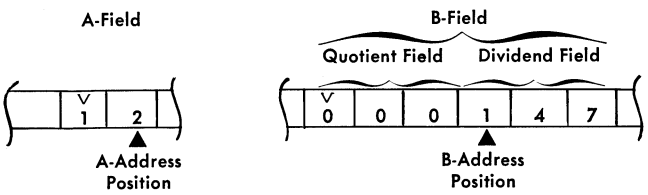


Figure 14. Divide — A and B Fields

The A-address specifies the units position of the divisor. The B-address specifies the high-order position of the dividend. 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:

12) 147 3-digit dividend
2-digit divisor
+1
—
6 positions must be allowed
in the B-field — 000DDD ±

The divisor located in the A-field can be signed or unsigned. If there are no bits in the units position of the divisor factor, the system assumes that the divisor factor is positive.

The dividend factor located in the B-field must have a sign in the units position of the factor. The sign bit configuration must be an A- and B-bit for plus, or a B-bit for minus.

The quotient field should always contain zeros when the divide operation begins. Any numeric data in these positions will be retained and added to the quotient produced. A carry generated in this manner will cause a divide overflow.

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.

Storage	A-Field	B-Field
Before	1 1 B	0 0 0 1 4 G 8
After	1 1 B	0 1 B 0 0 C 8
Before	1 1 B	0 0 0 0 1 4 G
After	1 1 B	0 0 0 A 0 3 E
Before	1 1 K	0 0 1 4 G 5 6
After	1 1 K	7 L 0 0 A 5 6

If the dividend is moved into the B-field by a zero and add instruction, it insures the proper signing of the B-field and the presence of zeros in the quotient positions of the B-field.

At the completion of the operation:

1. 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 15), 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 15).

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

Figure 15. Divide Sign Control

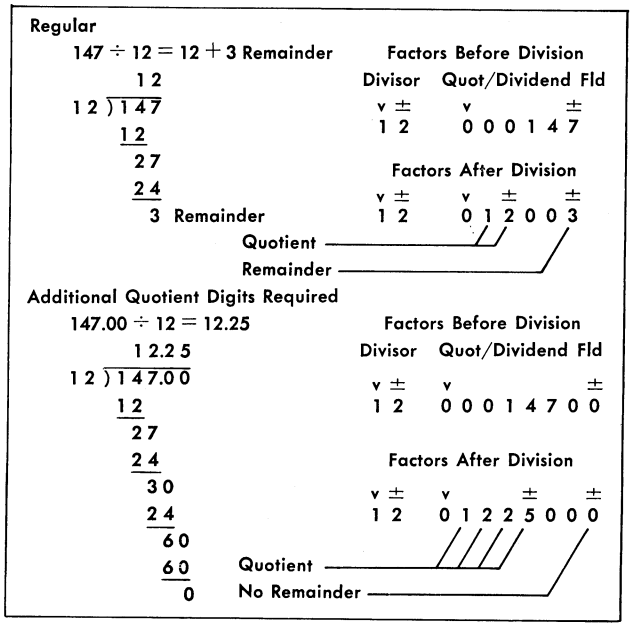


Figure 16. Additional Quotient Digits

Notes:

1. It is important to address the high-order position of the dividend (B-address of the divide instruction). 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 and the adjacent field may be changed. 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 will result in a divide overflow condition. Too small a quotient field is a programming error. The divide overflow condition can be tested by a test and branch instruction $-\overset{v}{J}(I)W$.

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 16.

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right)+6.5Q\left[5.6\left(\frac{A+2}{2}\right)+6.4\right]$

Address Registers after Operation:

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

These operations, used to manipulate data within core storage during processing, 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 $\overset{v}{D}$. The bit structure of the d-character used with the move instruction determines the type of operation that will be performed. (In Figure 17, all 64 characters composed of the bits shown 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 (Figure 20)

Instruction Form: D(A)(B)d

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 17).

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.

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

Word Marks: See Figure 17.

Timing: $T \approx 2.4\left(\frac{L+2}{2}\right)+2.4\left(\frac{A+1}{2}\right)+3.2\left(\frac{B+1}{2}\right)$

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.

Figure 17. d-Character Control Bits 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 18. Address Registers after Move Operations

Address Registers after Operation: See Figure 18. An instruction length of 6 chains the B-address and uses the last previous operation modifier.

Mnemonics

Because each mnemonic character has a special meaning (Figure 19), it is possible to construct the entire mnemonic for any of the 64 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 19. Mnemonic Characters for Move Instructions

EXAMPLE OF SCAN

Instruction: $\overset{\vee}{D}$ 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.

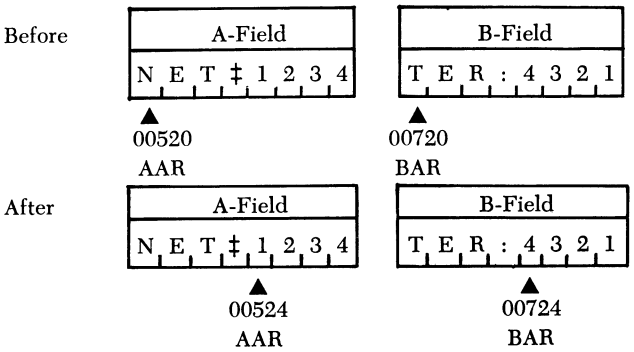


Figure 20 is a complete chart of the data move d-characters and mnemonics.

Move Characters and Suppress Zeros MCS

Instruction Form: Z(A)(B)

Function: This instruction causes the data in the A-field to be moved to the B-field. The A-field remains 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. See Figure 21 for an example of move characters and suppress zeros.

Figure 22 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.

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: $T \approx 2.4\left(\frac{L+2}{2}\right)+2.4\left(\frac{A+1}{2}\right)+3.2\left(\frac{B+1}{2}\right)$

Address Registers after Operation:

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

Comparing

The IBM 7010 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 C bits or word marks are not. The result of the compare operation is determined by the collating sequence of 7010 characters. See Figure 2. B can be equal to, unequal to, higher than, or lower than A.

Compare C

Instruction Form: C(A)(B)

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: $T \approx 2.4\left(\frac{L+2}{2}\right)+2.4\left(\frac{A+1}{2}\right)+3.2\left(\frac{B+1}{2}\right)$

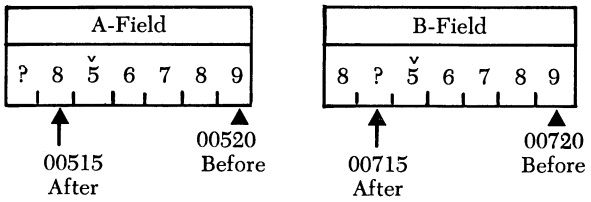
Address Registers after Operation:

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

EXAMPLES

Instruction: $\overset{\vee}{C}$ 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.



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	ǂ MRZR	, MRCR	% MRWR	~ MRNWR	\ MRZWR	ǂ 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	ǂ MRCWM	B, A, 8
	d-Ch BCD Coding (421 Bits)	NONE	1	2	2, 1	4	4, 1	4, 2	4, 2, 1	

Figure 20. Data Move d-Characters and Mnemonics

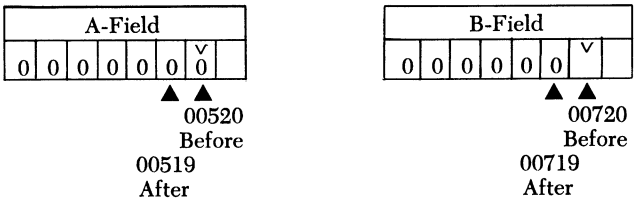
Example	Op Code	A–address	B–address
Move Char. and Suppress Zeros	v Z	xxxxx	xxxxx
Storage before		A–field (data) v ± 001206	B–field (data) v v ± bbbbbb
Storage after		v ± 001206	bb1206

Figure 21. Move Characters and Suppress Zeros Example

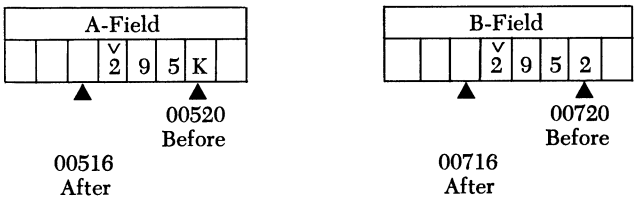
Example	Op Code	A–address	B–address
Move Char. and Suppress Zeros	v Z	xxxxx	xxxxx
Storage before		A–field (data) v ± 0010b@00.25	B–field (data) v v v ± bbbbbbbbbbb
Storage after		v ± 0010b@00.25	bb10b@bb.25

Figure 22. Move Characters and Suppress Zeros Example, Multiple Field

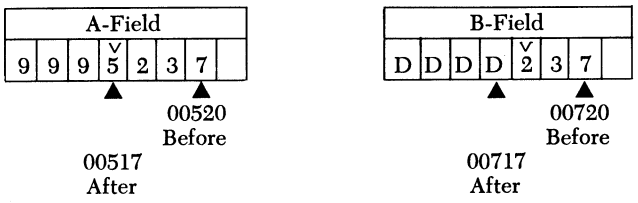
2. Result: B-field is low because of collating sequence.



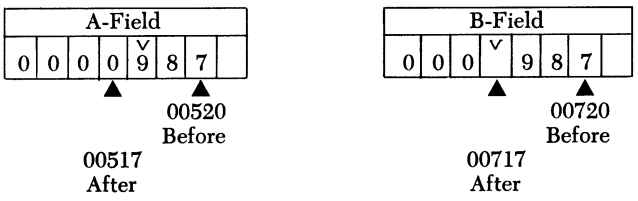
3. Result: B-field is high.



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.

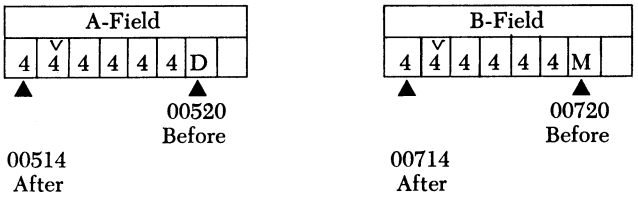


Table Lookup

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 7010 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: T(A)(B)d

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 23 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: $T \approx 2.4\left(\frac{L+2}{2}\right)+3.2\left(\frac{B+1}{2}\right)+2.4N\left(\frac{A+1}{2}\right)$

Address Registers after Operation:

I-address Reg	A-address Reg	B-address Reg
NSI	A-LA	Address of the function 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 23. Valid d-Characters for Table Lookup Instructions

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 24). 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.

Search Argument	TABLE	
	Function (Desired Factor)	Table Argument
1002	598	1000
	698	1002
	297	1003
	197	1004
	198	1005
	298	1006
	395	1241
	495	1242
	695	1243

Figure 24. Table Lookup Operation

Editing

The 7010 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 25). A step-by-step editing process of this example is shown in Figure 30. Thus, editing in the IBM 7010 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 transfers 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.

Move Characters and Edit MCE

Instruction Form: E(A)(B)

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.

$$\textit{Timing: } T \approx 2.4\left(\frac{L+2}{2}\right)+2.4A+3.2B+3.2\left(\frac{Z+1}{2}\right)+3.2\left(\frac{D+1}{2}\right)$$

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.

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 25. Editing

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 26. Zero Suppression

CONTROL CHARACTER	FUNCTION
CR (credit)	Body portion: undisturbed in the position where written. Status portion: if sign of the data field is plus, these two positions are 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 26). A special 0 is placed (in the body of the control word) in the rightmost limit of 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 27). 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).

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 28). 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.

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 27. Asterisk Protection

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 28. Floating Dollar Sign

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 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.

Sign Control Left

CR or – symbols can be placed at the left of a negative field (Figure 29). 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 – are 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 30). The control word is written with a point in the body to the left of the zero-suppression code (0). Two scans are enough to complete this editing operation unless the field contains no significant digits; in that case, three scans are required.

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.

EXAMPLE:

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

Figure 29. Sign Control Left

EXAMPLES:

1.	A–field	00000
	Control word (B–field)	bbb.b0
	First forward scan	000.00
	Reverse scan	bbb.00
	Second forward scan	bbb
	Results of edit	(Blank Field)
2.	A–field	29437
	Control word (B–field)	bbb.b0
	First forward scan	294.37
	Reverse scan	294.37
	Result of edit	294.37
3.	A–field	00001
	Control word (B–field)	bbb.b0
	First forward scan	000.01
	Reverse scan	bbb.01
	Results of edit	.01

Figure 30. Decimal Control

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 31 is a step-by-step editing process of the example shown in Figure 25.

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 31. Step-by-Step Editing Process (continued next page)

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 31. (Continued)

The 7010 program can examine conditions that arise during processing, and transfer the program to a predetermined set of instructions or subroutines as a result of specific tests. 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:

An Unconditional Branch occurs as a direct result of the execution of the instruction itself. Thus, no special condition (other than the execution of the program step) is needed to transfer the program out of its normal sequential execution.

A Conditional Branch occurs as a result of a particular condition, such as an arithmetic overflow, zero balance, etc. If the condition is present at the time a conditional branch instruction is executed, sequential execution of program steps is bypassed, and 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 takes the instruction that appears at the immediate right of the conditional branch instruction (next sequential instruction).

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

Operation Codes

Branch Unconditional B

Instruction Form: J (I) 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: Word marks are not affected.

Timing: $T \approx 2.4 \left(\frac{L + 2}{2}\right)$

Address Registers after Operation:

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

Test and Branch (Figure 32)

Instruction Form: J(I)d

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. Figure 32 shows the indicators and the d-characters used to test them.

The compare indicators are turned off by the next compare, table lookup, or test character and branch instruction and are set to the result of that operation. A computer reset operation turns off the compare indicators and turns on the compare low indicator. The overflow indicators are turned off by the test and branch instruction or a computer reset operation. A computer reset operation also turns off the zero result indicator.

Note: The d-characters of this instruction that are related to input/output operations are discussed in that area.

Word Marks: Word marks are not affected.

Timing: $T \approx 2.4 \left(\frac{L + 2}{2}\right)$

Address Registers after Operation:

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

DESCRIPTION	MNEMONIC	d-CHARACTER
Branch Unconditional	B	Blank
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

Figure 32. Test and Branch Instructions

INSTRUCTION	ACTUAL OP CODE, ADDRESSES, AND ACTUAL d-CHARACTER	MNEMONIC OP CODE, ADDRESSES, AND MNEMONIC d-CHARACTER	TRANSLATION
Branch If Word Mark	V(I)(B)1	BW(I)(B)	Branch to I-address if B-address has a WM bit.
Branch If Zone Equal	V(I)(B)2	BZN(I)(B)	Branch to I-address if B-address has no B nor A bits.
	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).
	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).
	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 or Zone Equal	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.
	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.
	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.
	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 33. Branch If Word Mark or Zone Equal Instructions

Branch if Character Equal BCE

Instruction Form: B(I)(B)d

Function: This instruction causes the bit configuration (BA 8 4 2 1 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: $T \approx 2.4 \left(\frac{L+4}{2}\right)$

Address Registers after Operation:

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

Branch if Bit Equal BBE

Instruction Form: W(I)(B)d

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 (A 8 1 bits) and the d-character contains a 9 (8 1 bits), the program branches.

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

Timing: $T \approx 2.4 \left(\frac{L+4}{2}\right)$

Address Registers after Operation:

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

Branch If Word Mark or Zone Equal

Instruction Form: V(I)(B)d

Function: This instruction (Figure 33) 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. Figure 33 shows both actual and mnemonic d-characters, and the conditions they test. 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: $T \approx 2.4 \left(\frac{L + 4}{2}\right)$

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	NSI
No branch	NSI	BI	B-1

Store Address Register

Instruction Form: G(C)d (See Figure 34.)

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.

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.

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 34. Store Address Register Mnemonics and d-Characters

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.

Timing: $T \approx 2.4 \left(\frac{L + 2}{2}\right) + 9.6$

Note: This instruction cannot be indexed.

Address Registers after Operation:

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

Store and Restore Status

Instruction Form: $(B)d

Function: This instruction stores or restores the contents of the internal machine status indicators and the channel status indicators under control of the d-modifier. The B-address specifies the storage position where information is to be stored or removed.

If the d-modifier is S, the internal machine status indicators are stored as one character, with the bit structure of the character corresponding to the indicators as follows:

BIT	INDICATOR
1	Equal
2	High
4	Low
8	Zero Balance
A	Arithmetic Overflow
B	Divide Overflow

If the d-modifier is E, the contents of channel 1 status indicators are stored as one character with the bit structure of the character corresponding to the indicators as follows:

BIT	INDICATOR
1	Not Ready
2	Busy
4	Data Check
8	External Condition
A	No Transfer
B	Wrong Length Record
WM	Channel Interlock

If the d-modifier is F, the contents of the second channel status indicators will be stored in the same manner as channel 1. Correct parity is automatically inserted.

If the d-modifier is R, the internal machine indicators are restored. The d-modifiers of 1 and 2 restore the

contents of the first and second channel status indicators respectively.

The internal machine status indicators may be stored or restored at any time. This instruction does not reset the indicators when storing. Word marks in the B-field will be eliminated when machine status is stored.

The following restrictions are placed on the use of this instruction:

1. If an attempt is made to store channel status while the channel is in process, the computer will interlock until the channel is free. Storing channel status resets the channel interlock.

2. If an attempt is made to restore channel status while the channel interlock is on, or if the channel is in process, an instruction check will occur.

Word Marks: Word marks are retained in all restore operations and are eliminated when machine status is stored.

Timing: $T \approx 2.4 \left(\frac{L+2}{2}\right) + 9.6$

Address Registers after Operation:

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

Set Word Mark SW

Instruction Form: , (A)(B)

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

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

Timing: $T \approx 2.4 \left(\frac{L+2}{2}\right) + 6.4$

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 CW

Instruction Form: □(A)(B)

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

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

Timing: $T \approx 2.4 \left(\frac{L+2}{2}\right) + 6.4$

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 CS

Instruction Form: /(B)

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 35) is 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.)

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 put there previously. 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: $T \approx 2.4 \left(\frac{L + 2}{2}\right) + 3.2\left(\frac{B + 1}{2}\right)$

Address Registers after Operation:

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

Clear Storage and Branch CS

Instruction Form: /(I)(B)

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: $T \approx 2.4 \left(\frac{L + 2}{2}\right) + 3.2\left(\frac{B + 1}{2}\right)$

INSTRUCTION ADDRESS	OP	INSTRUCTION A/I FIELD d	A/I FIELD x-ctrl fld	A/I FIELD d	B-FIELD d	d
00132	,	0	0 5	0	0	
00139	/				3 6 1 9	9
00135	G	0	0 1	3	4 B	
00142	V	0	0 1	2	9 0	0 5 0 0 1

Figure 35. One Method for Clearing Core Storage

Address Registers after Operation:

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

Halt H

Instruction Form:

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: $T = 2.4 \left(\frac{L + 2}{2}\right) = 3.6$

Address Registers after Operation:

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

Halt and Branch H

Instruction Form: (I)

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: $T \approx 2.4 \left(\frac{L + 2}{2}\right) \approx 9.6$

Address Registers after Operation:

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

No Operation NOP

Instruction Form: 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: $T \approx 2.4 \left(\frac{L + 2}{2}\right)$

Address Registers after Operation:

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

I/O Status Indicators

Each data channel has six input/output status indicators: not ready, busy, data check, condition, no transfer, and wrong length record. These indicators are automatically set during I/O operations to reflect the status of the addressed I/O unit. At the end of an I/O instruction read-out, before the channel is set in process, any I/O status indications from the addressed I/O unit are set in the I/O status indicators. If any indicator is set on, the I/O operation is terminated at this time, and the program continues on to the next instruction. If no indicator is set on, the channel is set in process and the I/O operation is executed. At the end of I/O instruction execution, the I/O status indications are again set into the I/O status indicators to record any change of status that occurred during execution.

The I/O status indicators are reset automatically during I/O instruction read-out or by computer reset (Figure 36).

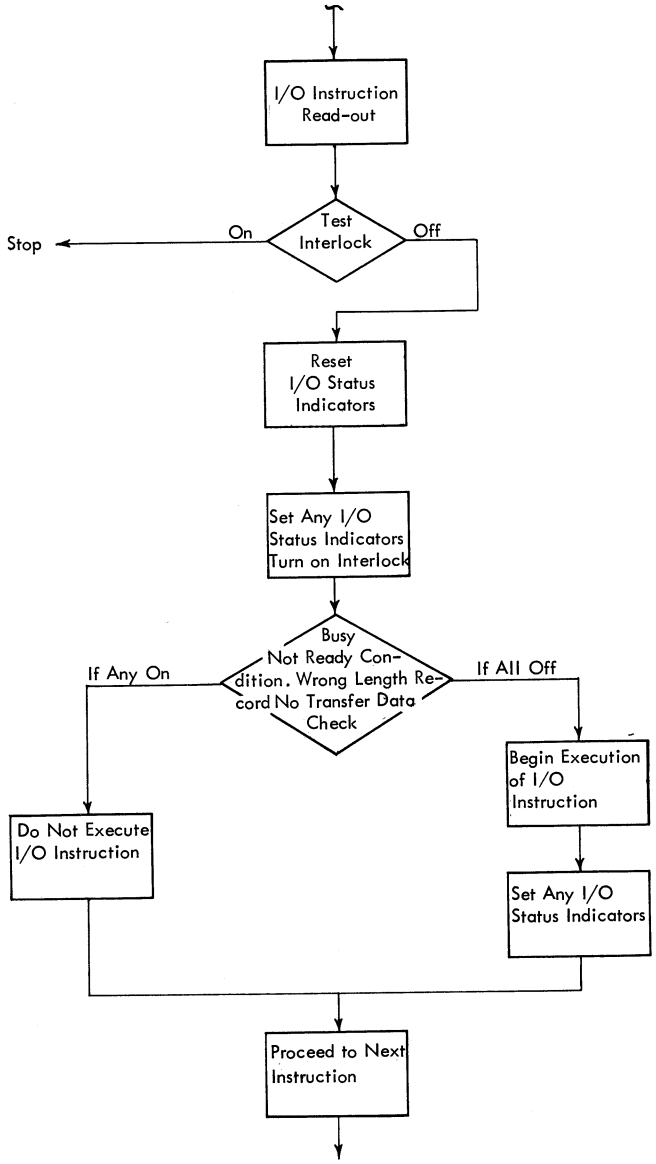


Figure 36. Automatic Interlock and I/O Status

The meaning of these indicators in general is:

Not Ready: I/O device requires manual intervention before it can accept an operation (power is off, out of cards, etc.).

Busy: I/O device cannot accept the operation because it is busy performing some normal function.

Data Check: Parity error is detected by central processor or I/O device during information transfer.

Condition: I/O device detected some error during the interval since it was last addressed.

No Transfer: Data transfer did not take place.

Wrong Length Record: Data field that transferred was different in length from either the field in storage or the field in the I/O device.

The significance of each status indicator varies with the kind of I/O device and is described with that device.

I/O Channel Interlock

When an I/O instruction is being read out, the interlock for the addressed channel is automatically turned on. If the channel interlock is on when the next I/O instruction for that channel is read out, the system interlocks and stops.

Each channel is interlocked to prevent a second operation on the same channel until the status of the prior I/O operation has been tested. In effect, the system interlock forces the program to interrogate the I/O status indicators at some time before the execution of a subsequent I/O operation on that channel. See Figure 36. The interlock for either channel is removed by the use of the branch if I/O channel status indicator on instruction or store status instruction.

Branch if I/O Channel Status Indicator On

Instruction Form: R or X(I)d

Function: This instruction (Figure 37) is a conditional branch instruction. Branching to the specified I-address occurs if I/O channel status indicators (specified by the bits of the d-character) are on when tested. An R operation code signifies channel 1. An X operation code specifies channel 2. The bit configuration of the d-character determines which indicators are to be tested. For example an RI3 (1 and 2 bits) tests the not ready and busy indicators; an RI7 (1, 2, 4 bits) tests the not ready, busy, and data check indicators. The RI≢ (1, 2, 4, 8, A, B bits) tests all six indicators.

This instruction must be given after an I/O instruction has been executed on a channel (before the next I/O operation on that channel) or the system will interlock. If all channel status indicators are tested by using an (RI≢) or (XI≢) instruction, a branch is not required to remove the interlock. If all indicators are not tested (RId or XId), a branch is required to remove the interlock.

Word Marks: Word marks are not affected.

Timing: $T \approx 2.4 \left(\frac{L+2}{2}\right)$

Address Registers after Operation:

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

Processing Overlap

Processing overlap (channel in process overlapped) allows computing to continue during most of the time taken by the transfer of data to or from input/output units and core storage. Input/output data moves to and from core storage via a two-character channel buffer (see Figure 6). From the channel buffer, data are exchanged with the I/O devices attached to the system. The time required to transfer two characters of data between core storage and the channel buffer is substantially less than the time required to transfer

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 37. Branch if I/O Status Indicator On Instruction

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 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 38. Test and Branch Instructions

two characters of data between the channel buffers and the I/O device. In the 7010 system, processing is interrupted only for the time required to transfer two characters to or from core storage via the channel buffer. In other words, processing can continue while the two characters of data are exchanged between the channel buffer and the I/O device. Also, the time an I/O device is starting, stopping, or otherwise preparing to send or receive data can be overlapped with processing. If a 7010 system has two data channels, reading and writing can be overlapped with processing to utilize the full power and capabilities of the 7010 system.

Once the execution of the overlapped I/O unit instruction is begun (channel in process overlapped), 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 completes its operation. Also, the I/O status indicators must be tested before the next I/O operation on that channel is encountered.

If an I/O status test or store I/O status instruction is read out that applies to a channel that is in process and overlapped, computing will be terminated until the channel has terminated operation. The system will then continue in normal fashion.

If an unoverlapped I/O instruction is read out for a channel, while the other channel is in process and overlapped, computing will terminate. The I/O operation will continue in a serial, unoverlapped mode.

Each channel has its own overlap-in-process indicator. The indicator is turned on at the beginning of an overlapped operation and is automatically turned off when the operation is completed. The test and branch instruction J(I)1 or 2 is used to determine the status of the overlap-in-process indicator.

Test and Branch (Conditional)

Instruction Form: J(I)d

Function: This conditional branch instruction (Figure 38) 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. Figure 38 lists the indicators and d-characters related to input/output operations. The basic description of this instruction is in "Branch and Miscellaneous Operations."

Priority Processing

Priority processing is an interrupt system that provides an automatic branch to a fixed storage location (00101) when certain conditions of the I/O channels or devices occur. These conditions are stored in priority request indicators, which may be tested to determine the specific cause of the interrupt. Each channel has six priority request indicators: channel overlap, seek complete, I/O unit, inquiry, outquiry, and attention.

Two modes of system operation, priority alert mode and normal mode, enable the programmer to control whether a given sequence of instructions may be interrupted or not. One or the other of these modes is in effect at all times. The mode in effect is determined by the priority alert mode indicator. If the priority alert mode indicator is on, the system is in the priority alert mode and is capable of accepting an interrupt. If the priority alert mode indicator is off, the system is in a normal mode and cannot accept an interrupt.

The priority alert mode indicator is turned on by the Y(I)E branch and set priority alert mode instruction and is turned off by:

1. A priority interrupt: a priority interrupt indicator turns on, and the system branches to storage location 00101, turning off the priority alert mode indicator.
2. Priority test and branch instruction Y(I)X, and reset priority alert mode.
3. Computer reset.

Priority Test and Branch

Instruction Form: Y(I)d

Function: This instruction is used to test conditions causing an interrupt and to set and reset interrupt modes. In testing, if the condition indicated by the d-modifier is satisfied, a branch to the I-address will occur; if the condition is not satisfied, instruction execution continues in normal sequence.

The modifiers used to set and reset interrupt modes always cause an unconditional branch to the I-address. When a branch occurs, the address of the instruction immediately following the branch instruction is automatically transferred into the B-address register. The table that follows lists permissible d-modifiers and their effect.

Word Marks: Word marks are not affected.

D-MODIFIER		BRANCH CONDITION
A	BXPR1	Channel 1 Attention
B	BXPR2	Channel 2 Attention
E	BEPA	Branch and Set Priority Alert Mode
F	BUPR2	Channel 2 Selected I/O Unit
N	BQPR1	Channel 1 Outquiry
Q	BIPR1	Channel 1 Inquiry
S	BSPR1	Channel 1 Seek Complete
T	BSPR2	Channel 2 Seek Complete
U	BUPR1	Channel 1 Selected I/O Unit
X	BXPA	Branch and Reset Priority Alert Mode
1	BOPR1	Channel 1 Overlap Complete
2	BOPR2	Channel 2 Overlap Complete
≠	BQPR2	Channel 2 Outquiry
*	BIPR2	Channel 2 Inquiry

Timing: $T \approx 2.4 \left(\frac{L + 2}{2}\right)$

Address Registers after Operation:

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

The interruption of the program takes place during the time the instruction is being read but before the actual execution of the instruction. Some operation codes can be interrupted, while others cannot. A number of operation codes of one length can be interrupted, while the same code of another length cannot. Only unchained operations can be interrupted. Figure 39 shows operation codes that can be interrupted.

When an interrupt branch has occurred, the priority routine must make provision for returning to the interrupted instruction of the main routine. This is accomplished by storing and decrementing by six the address in the B-address register (BAR), following the interrupt. The decremented B-address is the address specified in the branch unconditionally and enter instruction, Y(I)E, located at the end of the priority routine.

If the status of arithmetic and logic indicators will be changed by operations in the priority routine, their status must be saved. If the status is saved, it must be restored before branching back to the main routine program.

The following list of priority request indicators and Figure 40 describe conditions that cause the indicators to be turned on and off.

Channel Overlap Priority Request Indicator is turned on at completion of any overlapped data transfer on the channel (including overlapped file address transfers on seek operations). Status of the indicator is determined by the priority test and branch instruction (Y(I)d). The indicator is turned off by any I/O status condition test operation to that channel (R(I)d).

Seek Complete Priority Request Indicator is turned on when any access mechanism (1301) for that channel is in a seek complete status. The indicator is tested by the priority test and branch instruction (Y(I)d) and is turned off automatically when the seek complete status of all access mechanisms on a channel is reset off. Seek complete status is reset off by directing a read, write, or I/O NOP instruction to the unit or units in a seek complete status.

OPERATION	OP CODE	INTERRUPTIBLE LENGTH	NON-INTERRUPTIBLE LENGTH
Zero and Add	?	11	1,6
Zero and Subtract	!	11	1,6
Add	A	11	1,6
Subtract	S	11	1,6
Multiply	@	11	1,6
Divide	%	11	1,6
Move Characters and Edit	E	11	1,6
Move Characters and Suppress Zeros	Z	11	1,6
Compare	C	11	1,6
Clear Storage and Branch	/	11	1,6
Set Word Mark	'	11	1,6
Clear Word Mark	□	11	1,6
Branch if Bit Equal	W	12	1,6
Branch on Word Mark or Zone Equal	V	12	1,6
Move Data	D	12	1,6
Branch if Character Equal	B	12	1,6
Table Lookup	T	12	1,6
Test and Branch	J	7	1
Branch if I/O Channel Status Indicator On (Ch 1)	R	7	—
Branch if I/O Channel Status Indicator On (Ch 2)	X	7	—
Store and Restore Status	$	7	—
Priority Test Branch	Y	7	1
No Operation	N	—	Any
Move (I/O Operation)	M	—	10
Load (I/O Operation)	L	—	10
Store Address Register	G	—	7
Halt	.	—	1,6
Control Unit	U	—	2
Control Carriage	F,2	—	2
Stacker Select Feed	K,4	—	2

Figure 39. Interruptible and Noninterruptible Operation Codes

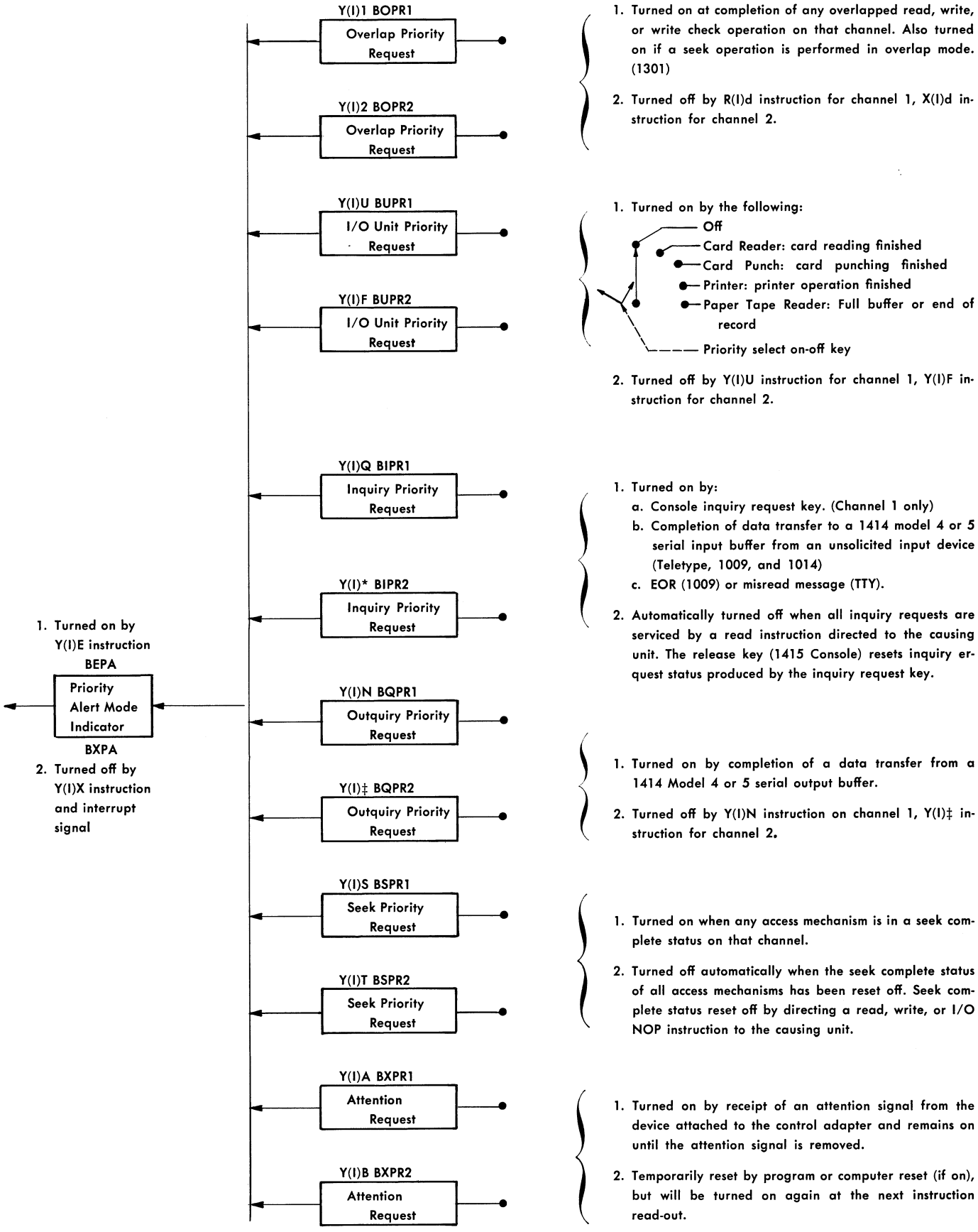


Figure 40. Priority Request Indicators

I/O Unit Priority Request Indicator is operated under control of a rotary priority select switch (1415-2 Console) and one of the I/O devices indicated by the switch setting. Each switch has five positions: OFF, CARD READER, CARD PUNCH, PRINTER, and PAPER TAPE READER. Associated with each priority select switch is a priority select on-off key (1415). This double action key, when on, allows the I/O device designated by its related priority request switch to turn on the I/O unit priority request indicator. The I/O unit priority request indicator is turned on when the I/O device selected by the priority select switch for a channel finishes an operation:

1. Card reader—card reading finished.
2. Card punch—card punching finished.
3. Printer—line of print printed.
4. Paper tape reader—full buffer or end of record.

The I/O unit priority request indicator is also turned on by turning the priority select on-off switch from OFF to ON. The indicator is reset by testing it with the interrupt test and branch instruction.

Inquiry Priority Request Indicator is turned on by completion of a data transfer from an external I/O device to any serial input buffer in the 1414-4 or -5, an end of record (EOR) for the 1009, a misread telegraph message, or a console inquiry request (on channel 1 only). The indicator is reset by performing a read operation with the input buffer requesting service, or by the release or cancel key if the console caused the inquiry request. If more than one interrupt is requesting service, the indicator will remain on until all such buffers have been read.

Outquiry Priority Indicator is turned on by completion of data transfer from a 1414-4 or -5 serial output buffer to an output device. It is turned off by testing the indicator with an appropriate interrupt test and branch (Y(I)d) instruction.

Attention Priority Request Indicator is turned on by receipt of an attention signal from the device attached to the channel control adapter and remains on until the attention signal is removed.

The channel overlap, I/O unit, and outquiry priority request indicators and the inquiry priority request (when turned on by console inquiry request) are reset by program or computer reset in addition to the resets previously specified. The seek complete and inquiry priority request indicators (when turned on by a buffer inquiry request) are temporarily turned off by either program or computer reset, if they are on, but will be turned on again at the next instruction read out. The attention priority request indicator will likewise be temporarily reset; whether it comes on again depends upon the characteristics of the device attached to the control adapter.

Interrupt requests occur on a real-time basis. For this reason, it is possible for a request to occur too late during the instruction read-out time of an interruptible instruction to effect an interrupt. However, the priority request indicator related to the type of interrupt will be turned on. In these instances, the interrupt will occur during instruction read-out time of the next interruptible instruction, with the following exception: when an overlap priority request occurs too late to interrupt and the next instruction is a branch if I/O status indicator on, the priority request indicator previously turned on is reset and no interrupt occurs; the resetting of the overlap priority request indicator under these circumstances is a normal function of the branch if I/O status indicator on instruction.

Input/Output Instructions

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. The 7010 input/output instructions have the form M/L(xxx)(B)d.

M or L Operation Code

For read or write operations, the operation code is either M (move) or L (load) mode. The only difference is in the processing of word marks and word-separator characters.

If the operation code is an M (move), I/O data transmission is in seven-bit mode (six data bits plus parity bit). Input word-separator characters of incoming data are stored unchanged as word-separator characters in core storage. Word-separator characters in core storage are written unchanged as word-separator characters on the output medium. No word marks are transferred to the output medium.

When the operation code is an L (load), word marks are considered in one to two ways: seven-bit mode and eight-bit mode. Load mode (seven-bit) data transmission is under word separator control. In this mode, a single word separator in incoming data is stored in core storage as a word mark over the next incoming character. Two consecutive word-separator characters enter core storage as a single word-separator character. Thus, the input record length is shortened one position in either of these operations: storing one word mark or storing one word-separator character. (Figure 41).

When core storage word marks are transferred to output devices, they are converted to word-separator characters. A word mark over a character in storage is written as a word separator immediately preceding the character on the output medium. A word separator in

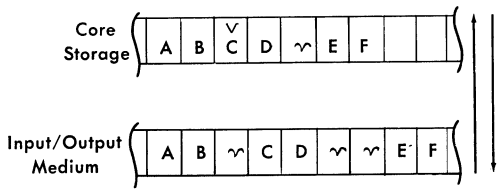


Figure 41. Load Mode, Seven-Bit

core storage is converted to two word separator characters in the output.

In the load (eight-bit) mode, word marks are read or written as part of normal data transmission. (This is possible only with I/O devices capable of handling eight-bit characters.

X-Control Field

The x-control field consists of three characters. The first character (hundreds position) selects the channel and defines whether subsequent channel operation on that channel is to be in overlap or nonoverlap mode. The second character (tens position) defines which unit on the channel is to be addressed. The third character (units positions) performs different functions, depending on the I/O unit addressed. See Figure 42.

B-Address

The B-address defines the high-order (leftmost) position of the data field to be read or written. Normal definition of input or output fields is accomplished by a group-mark-with-word-mark placed immediately to the right of the rightmost position of the data field. Normal I/O data transmission is terminated by sensing the WM-with-GM or by an external end-of-transfer signal from the specified I/O device.

d-Character

There are eight legal d-modifiers for M or L operation codes:

R or W denote normal read or write, respectively. I/O data transmission is terminated by sensing a group-mark-with-word-mark or by an external end-of-transfer signal.

$ or X denote read or write to the end of storage. I/O data transmission is terminated by sensing the last valid storage address or by an external end-of-transfer signal. (NOTE: Instructions using the $ or X d-character cannot be overlapped.)

Q or V are substituted for R or W respectively if it is desired to test the status of an I/O device without performing any data transfer. This mode of operation is identical to normal read or write except that the channel is not put in process and no data transfer occurs (except for file addresses in file status tests).

S or C are substituted for R or W to execute a sense or control command, respectively, when addressing the control adapter. The data transfer may be a normal data record or some type of control information. Internally, the operation is identical to a read or write command.

1402 Operation Codes

Read a Card R or RW

Instruction Form: M/L(xxx)(B)d

Function: This instruction causes the transfer of 80 characters in the card-read synchronizer to core storage. If the units position in the x-control fields 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 NR, 1, or 8/2 pocket. During the feed cycle the card-read synchronizer 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 synchronizer 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-with-word-mark sensed in core storage. See Figure 43.

Word Marks: A group-mark-with-word-mark must appear in the core-storage position to the immediate right of the data record. If the $\overset{v}{L}$ op code is used, word-separator characters 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.

Timing: $T \approx 2.4\ (L + 1) + I/O$

(See "IBM 1402 Model 2 Timing Considerations")

Address Register after Operation:

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

Figure 44 shows I/O channel status indicators set during 1402 card read operations.

Select Stacker and Feed SSF

Instruction Form: Kd

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 the NR, 1, or 2 pocket, depend-

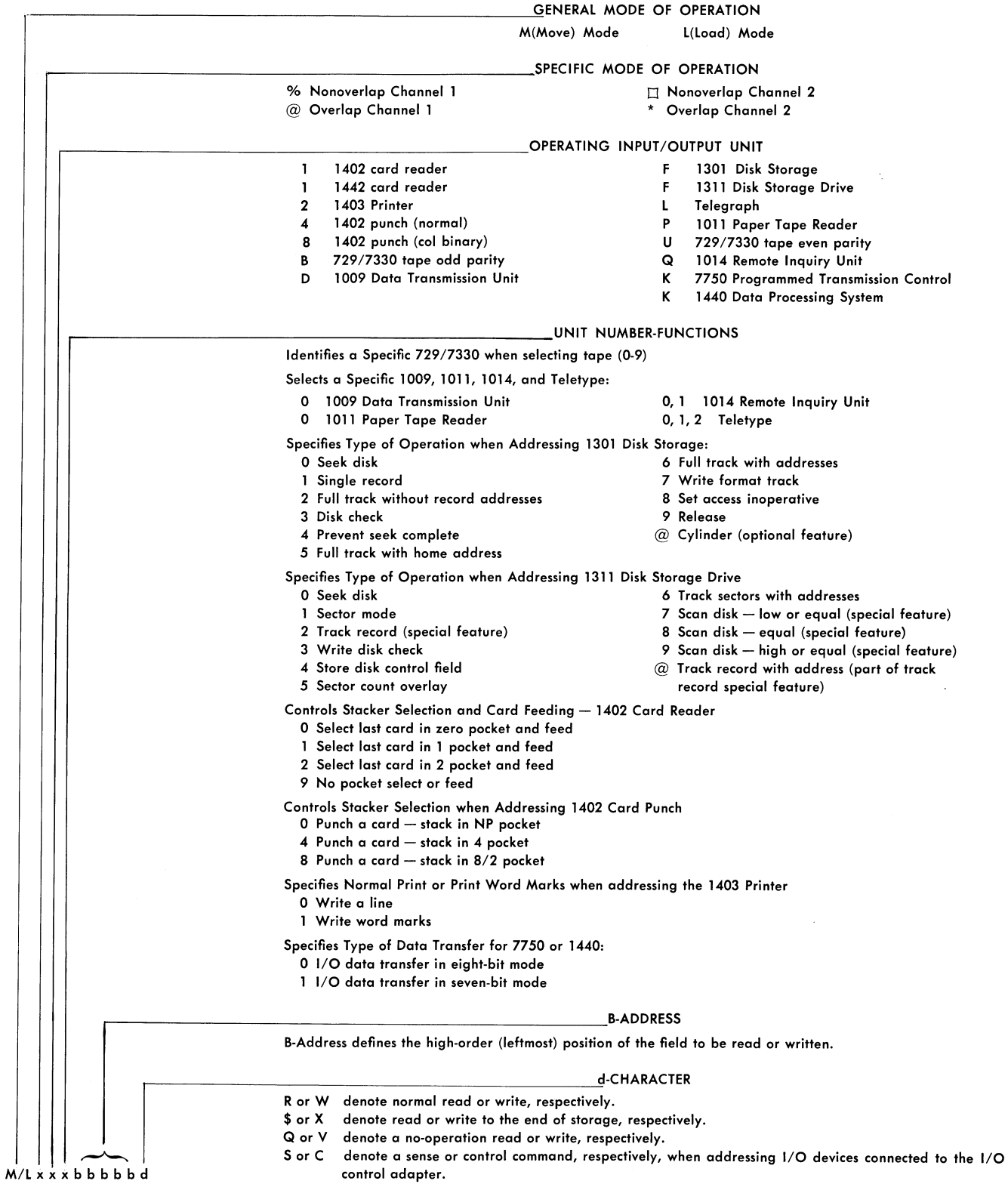


Figure 42. I/O Instruction Form

I/O UNIT	OP CODE	X-CTRL FIELD	DESCRIPTION	MNE-MONIC	d-CHARACTER		OPERATION	NOTES
					CHAR-ACTER	CONTROL		
CARD READER	v M or v L	%10 %11 %12 %19	Read a card	R or R1 (Ch 1) 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. Same as above, except card is stacked in pocket 1. Same as above, except card is stacked in pocket 2. Transfer 80 characters from read buffer to core storage. THERE IS NO CARD FEED AND STACKER SELECT OPERATION.	If v L op code is used, word-separator characters are read into storage as word marks placed over the following data character.
CARD READER	v K (Ch 1) or v 4 (Ch 2)	None	Select stacker and feed	SSF or SSF1 (Ch 1), SSF2 (Ch 2)	0 1 2	Select 0 stacker Select read stacker 1 Select read 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.

Figure 43. 1402 Card Read Operation Codes

ing on the d-character of 0, 1, or 2. A card feed is initiated that refills the card-read synchronizer and positions this card at the select station. See Figure 43.

Word Marks: Word marks are not affected.

Timing: $T \approx 2.4\ (L + 1) + I/O$

(See "IBM 1402 Model 2 Timing Considerations")

Address Registers after Operation:

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

Punch a Card P or PW

Instruction Form: M/L(xxx)(B)d

Function: This instruction causes the transfer of eighty characters in core storage to the punch synchronizer. The units position of the x-control field (n) can contain 0, 4, or 8, which direct the card to the NP, 4, or 8 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-with-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. See Figure 45.

Word Marks: A group-mark-with-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 marks are translated to word-separator characters for punching. Use of the load mode causes the

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*
No Transfer	~~1~~ (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 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.
Wrong Length Record	— (B-bit)	Wrong length record *Never set on Select Stacker and Feed Instruction*

Figure 44. I/O Channel Status Indicators Set during Card Reader Operations

I/O UNIT	OP CODE	X-CTRL FIELD	DESCRIPTION	MNE-MONIC	d-CHARACTER		OPERATION	NOTES
					CHAR-ACTER	CONTROL		
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.	v If 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 45. IBM 1402 Card Punch Operation Codes

word-separator character to be punched ahead of its associated character and affects the result field length.

Timing: T $\approx$ 2.4 (L + 1) + I/O

Address Registers after Operation:

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

Refer to Figure 46 for the I/O channel status indicators set during 1402 card punch operations.

1402 Model 2 Timing Considerations

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

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

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

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	Hole count check detected during the punch cycle of the following card. Error card is directed to 0 stacker; if it is stacked, a blank card (second card behind error card) is fed. Parity error detected during punching.
No Transfer	$\bar{8}$ (A-bit)	Never set
Wrong Length Record	— (B-bit)	Wrong length record (this card not punched)

Figure 46. I/O Channel Status Indicators Set during Card Punch Operations

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

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

3. Select stacker and feed instruction.

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

4. Punch a card instruction.

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

1403 Operation Codes, Timing, and Features

Write a Line W1 or W1W

Instruction Form: M or L(xxx)(B)d

Function: This instruction (Figure 47) transfers 100 (or 132) characters in core storage to the print synchronizer. 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-with-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-separator characters during the transfer to the print synchronizer. There is no character on the chain for a word-separator character, and a blank space results during the printing. When operation is in the load mode, a blank space (corresponding to the position that contained the word-separator character) appears ahead of its associated character. This increases the result field length in the print synchronizer and may cause the loss of some of the field.

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 (Ch2)		Transfer word marks from storage to print buffer and print as 1's.	Positions without word marks transfer and print as blanks. (With 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 49 for the list of d-characters and operations.		
	V 2 (Ch 2)			CC2			

Figure 47. IBM 1403 Printer Operation Codes

Word Marks: A group-mark-with-word-mark must appear in the core storage position to the immediate right of the data record.

Timing: $T \approx 2.4\,(L + 1) + I/O$

Address Registers after Operation:

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

Write Word Marks WM1

Instruction Form: M(xxx)(B)d

Function: This instruction transfers all word marks in the data field to the print synchronizer as digit ones. The one in the units position of the x-control field specifies the write word mark operation. Positions without word marks are transferred to the print synchronizer 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 $\overset{V}{L}$ op code is used, the effect on printing is the same as for write a line.

Figure 48 shows the I/O channel status indicators set during 1403 operations.

Word Marks: A group-mark-with-word-mark must appear in the core storage position to the immediate right of the data record.

Timing: $T \approx 2.4\,(L + 1) + I/O$

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	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)
No Transfer	⅋ (A-bit)	Never set
Wrong Length Record	— (B-bit)	Wrong length record (line is not printed)

Figure 48. I/O Channel Status Indicators Set during Write a Line and Write Word Marks Operation

Control Carriage CC1

Instruction Form: Fd

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. Figure 49 shows the d-characters and the operations they initiate. Figure 50 shows channel status indicators.

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.

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

1403 Printer Timing Considerations

Data transfer from the print area of core storage to the print synchronizer 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 ordinarily start until 1,492 microseconds after the initiation of the first transfer.

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 49. d-Character for Control Carriage Instruction

INDICATOR	d-CHARACTER	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	Never set
No Transfer	A	Never set
Wrong Length Record	B	Never set

Figure 50. I/O Channel Status Indicators Set during IBM 1403 Carriage Operation

Numeric Print Feature

The numeric print feature for the IBM 1403 Printer is designed for businesses having certain 7010 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. Other industries can also use this feature for many applications in which reports are or can be numerically coded.

With this feature, the user can switch from alphameric to numeric mode by simply 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 numeric mode, the 1403 can print 1,285 lines per minute — more than twice as fast as in alphameric mode.

To change from one mode to another, an operator, with 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.

Magnetic Tape Instructions (729/7330)

Read or Write Tape (Figure 51)

Instruction Form: M(xxx)(B)d

Function: The n-character in 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-with-word-mark in core storage (whichever is sensed first) stops a read tape operation. If a group-mark-with-word-mark is sensed first, the data transfer stops, but tape movement continues to the next inter-record gap. A group-mark-with-word-mark in core storage stops the write operation and causes an inter-record gap on the tape. See Figure 52.

	OPERATION CODE	MNEMONIC	d-CHARACTER	OPERATION	NOTES
v M	Read tape	RT or RTB	R	A record is transferred from magnetic tape to core storage.	Reads to first v ǂ or IRG.
	Write tape	WT or WTB	W	A record is transferred from core storage to magnetic tape.	Writes to first v ǂ.
v L	Read tape with word marks	RTW or WTBW	R	A record with word marks is transferred from magnetic tape to core storage.	Reads to first v ǂ or IRG.
	Write tape with word marks	WTW or WTBG	W	A record with word marks is transferred from core storage to magnetic tape.	Writes to first v ǂ.
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 WTREW	X		

Figure 51. Magnetic Tape — Read and Write Instructions

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)	4	Processing unit received wrong parity character
(W)		Tape adapter unit sent wrong parity character Tape mark read in odd parity mode Tape adapter unit received wrong parity character
(U)		Tape adapter unit detects rbc parity error Set if write tape mark in odd parity
Condition (R)	8	1st character of record was tape mark
(W)		Foil strip detected
(U)		Never set (unless tape mark read)
No Transfer (R-W-U)	b (A-bit)	Never set
Wrong Length Record (R) (W-U)	— (B-bit)	Wrong length record (usually set when d-character is $) Never set (unless record is of zero length and first character written is v ǂ)

Figure 52. I/O Channel Status Indicators Set during Tape Operations

Word Marks: Word marks in the tape record do not affect either operation.

Timing: $T \approx 2.4\,(L + 1) + I/O$

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 %Bn, 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.

The RTG or WTE 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 (Figure 51)

Instruction Form: L(xxx)(B)d

Function: These two instructions have the same function as the read tape and write tape instructions, except that word marks are written on tape as word-separator characters (A841), and word-separator characters are read into storage as word marks. In a write operation, a word-separator character is written on

tape, one position ahead of the associated character. In a tape read operation, word marks are associated with the next character read from tape.

Word Marks: Each word-separator character requires one tape position.

Timing: T ≈ 2.4 (L + 1) + I/O

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 operation. This assures proper translation between the tape and core storage.

These instructions can be done in an odd-parity mode and with $ or X d-character. 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 + 2

Unit Control (Figure 53)

Instruction Format: U(xxx)d

Function: The tape unit, specified by the units position n-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 record	B
Skip and blank tape	E
Write tape mark	M
Rewind	R
Rewind and unload	U

Word Marks: Word marks are not affected.

Timing: T ≈ 2.4 (L + 1) + TM

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

	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 7010 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 7010 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.)

Figure 53. Magnetic Tape Control Instructions

IBM 1415 Model 2 Console

The 1415 Model 2 Console is a desk-type unit having a control panel, a console-I/O printer, and an indicator-light panel (Figure 54). The console is used to:

1. Control the system manually.
2. Display machine and program status indicators.
3. Display contents of storage and certain registers.
4. Revise or correct data and instructions in storage.
5. Provide an inquiry mode of operation.
6. Provide messages under stored program control.
7. Provide a method of program or machine operation diagnosis.

There are no lights on the indicating panel to display data and instructions in storage.

The 7010 system automatically provides, on the console-I/O printer, an operating log (printed copy) of all major data handling operations performed at the console. In addition, an automatic stop print-out occurs on all manual, programmed, and error stops. The stop print-out consists of an identifying character, the contents of the instruction address register, A-address register, B-address register, operation register, modifier register, the A-channel, B-channel, assembly channel, result register, unit select registers, and I/O unit number registers for channel 1 and 2 (Figure 55).

Console Control Panel

The control panel contains the keys, lights, and switches that control the 7010 (Figure 56).

Power Keys and Computer Reset

POWER ON

Pressing this key initiates a controlled sequence of power application to the system and turns on the illuminated portion of the power-on key. Completion of the sequence results in a power-on reset, and the ready light is turned on. If the preceding operation was depression of the DC off key, depressing the power-on key reapplies DC power to the system.

POWER OFF

Pressing this key removes all electrical power from the system in a normal interlocked sequence. Removal of system power also turns off the ready light and the illuminated portion of the power-on key.

DC OFF

Pressing this key removes DC power from the system in a normal interlocked sequence. Use of this key turns off the ready light.

Figure 54. IBM 1415-2 Console

	PRINTOUT IDENT.	I A R	A A R	B A R	OPERATION CODE OP MODIFIER	A CHANNEL CONTENTS B CHANNEL CONTENTS ASSEMBLY CHANNEL RESULT REGISTER	CH. #1 UNIT SEL REG UNIT NUM REG	CH. #2 UNIT SEL REG UNIT NUM REG
NORMAL STOP (Double Space)	S	XXXXX	XXXXX	XXXXX	X X	X X X X	X X	X X
HALF-CYCLE (Double Space)	C	XXXXX	XXXXX	XXXXX	X X	X X X X	X X	X X
ERROR STOP (Double Space)	E	XXXXX	XXXXX	XXXXX	X X	X X X X	X X	X X
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 55. IBM 1415 Printing Layout

EMERGENCY POWER OFF PULL SWITCH

When this switch is operated, all power is immediately removed from the system. This switch is intended for emergency use only (to prevent injury to an individual or damage to the system). When this key is operated, customer engineering action is needed to restore power to the system.

COMPUTER RESET

Operating this key resets all check circuits, resets the instruction address register to 00001, resets all timing clocks, and resets all machine indicators (overflow, compare, etc). The inquiry indicators (except the console inquiry indicator) and the tape-density indicator are not reset.

Mode Switch

The six modes of machine operation are selected by the mode switch (Figure 56). Changing the mode switch setting produces an automatic stop print-out.

RUN MODE

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

I/E CYCLE MODE

In the I/E cycle mode, an instruction is executed in two parts; instruction read-out and instruction execution. 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, stop, and perform a stop print-out operation. Print-out identification character is the letter C.

The second operation of the start key causes the execution of the instruction previously read out. After executing the instruction, the system stops and another stop print-out occurs. Subsequent operation of

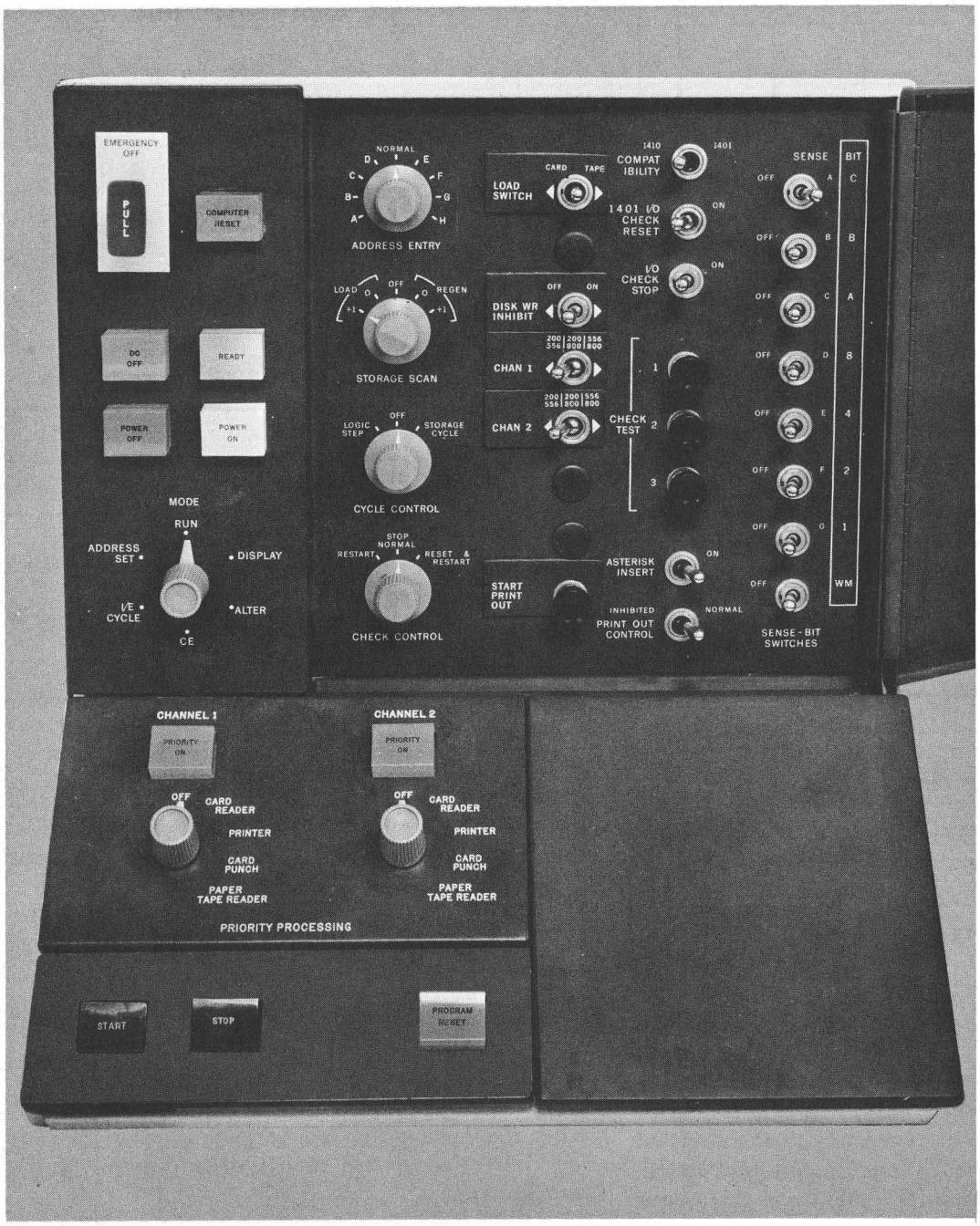


Figure 56. Console Control Panel

the start key results in the alternate instruction and execution phases.

DISPLAY MODE

Any portion of storage may be displayed (printed) on the console-I/O printer 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.

A display operation is accomplished as follows:

1. Turn mode switch to DISPLAY.
2. Depress start key (Figure 56). The letter D is printed, followed by a single carrier space operation.
3. At the keyboard of the console-I/O printer, manually enter the five-digit, high-order address of the field to be displayed. The five-digit address will be printed as the keys are depressed.

After the fifth address character is printed, an automatic carrier return occurs, and the character D is automatically printed. This is followed by a single-space operation and the printing of the contents of storage, starting at the addressed position and continuing until a word mark is sensed. The character and its associated word mark are printed. Any incorrect parity character is underlined.

The adjacent storage field can be displayed if the start key is pressed again. A continuous display results from holding the start key in its operated position. If, during a display operation, the end-of-line condition is detected, the carrier returns and the display continues. The display operation can be ended at any time by pressing the stop key.

If a system error occurs during a display operation, a stop print-out occurs. Identification character is the letter E. Channel errors are ignored; address errors cause a stop. The carrier returns when the stop key is pressed or the error is reset.

NOTE: If an extra key is depressed when keying in the address, the extra character overrides the automatic carrier return and leaves the carrier on the same line. The next print-out takes place on the same line with no error indication or machine stop.

ALTER MODE

By using the ALTER setting of the mode switch, it is possible to alter the information at any storage location. A display operation of the specific storage location must be completed before an alter operation can be performed. This display operation prerequisite insures having a record of the storage content before the alteration takes place. An alter operation is performed as follows:

1. Perform display operation.
2. Turn mode switch to ALTER. The letter A is printed, the carrier is moved one space, and the keyboard of the console-I/O printer is unlocked.
3. Begin manual entry of characters with keyboard of the console-I/O printer.

The alter operation begins at the initial location specified by the preceding display operation. Characters entered by the keyboard replace the previously displayed information. If information is to be replaced, it is duplicated at the keyboard. If characters are to be corrected or changed, new characters are entered at the keyboard. Use of the space bar enters valid blanks in storage. The operation can continue until a group mark or end of line is sensed. The operation can also be terminated by the stop key. Termination of the alter operation initiates a carrier return and index operation. When a character is entered into the last location of storage during an alter routine, the alter operation is terminated and the carrier is returned and an index operation is performed.

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

ADDRESS SET MODE

This setting of the mode switch is used to start a program at a specific place in storage. An address set operation is performed as follows:

1. Turn mode switch to ADDRESS SET (causes normal stop print-out).
2. Depress start key. The console-I/O printer will automatically print a B; the carrier will move one space and wait for a new address.
3. At the keyboard, manually enter the storage address at which the program is to start. This action is followed by an automatic carrier return and line space (index) operation.
4. Turn mode switch to either RUN or I/E CYCLE.
5. Depress start key. The program resumes control at the instruction located at the address entered by the operator.

CE (CUSTOMER ENGINEERING) MODE

When the mode switch is set to CE, customer engineering functions, such as storage scan, are available to the customer engineer.

Control Keys

START KEY

With the mode switch set to RUN, depressing the start key causes the system to switch to run status and begin executing instructions. Also, the start key initiates the operation when the mode switch is set to I/E CYCLE, DISPLAY, or ALTER. Depressing the start key causes error indicators to be reset. It is also active in some customer engineering modes.

STOP KEY

Operation of the stop key while the program is running stops the program after the current instruction is executed; an automatic stop print-out is performed. The print-out identification character is the letter S.

PROGRAM-RESET KEY

Pressing this key forces an error reset. The instruction address register is reset to 00001 and address registers, data registers, operation register, and modifier register are reset to no bits.

Figure 57. Console I/O Printer Keyboard

Read and Write — Console Printer Operations

The console-I/O printer read (inquiry) and write (reply) operations are performed under stored program control. They can occur while the system is operating in either the run or I/E cycle mode. Use of the console-I/O printer keyboard keys is discussed in order of use in performing the read and write operations.

REQUEST KEY (CONSOLE I/O PRINTER READ OPERATION)

A read operation is initiated by pressing the inquiry request key (Figure 57), which turns on the channel 1 inquiry status indicator. The program tests the inquiry status indicator with a test and branch instruction J(I)C. If the indicator is on, the program branches to a subroutine that contains the read console printer instruction M(X)(B)R (Figure 58). Initiation of an

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 58. Console-I/O Printer Control Instructions

inquiry operation causes the character I to be automatically printed by the console printer (Figure 55). The printer then single spaces and the keyboard is unlocked.

Once the keyboard is unlocked, data can be entered manually. Information enters storage one character at a time as it is typed on the keyboard. A space operation causes a blank to be entered into storage. The first character is placed in storage beginning at the location indicated by the B-address of the read console instruction. Subsequent characters are placed in the next higher positions.

Before the console inquiry operation, the program must define the last position of the inquiry field by storing a group-mark-with-word-mark character. Figure 59 shows conditions that set I/O channel status indicators on during an I/O printer read operation.

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
No Transfer	A	No message request — Cancel Key operated before inquiry
Wrong Length Record	B	Wrong length record

Figure 59. I/O Channel Status Indicators Set during IBM 1415 Read Operation.

RELEASE KEY

As the last inquiry character is entered in storage, the group-mark-with-word-mark character is sensed. The operator must press the release key at this time to obtain a correct length record and to insure processing of the inquiry. With a correct length record, pressing the release key:

1. Terminates the operation.
2. Causes a carrier return and line space.
3. Locks the keyboard.
4. Causes the program to continue to the next instruction.

If the number of characters to be entered exceeds the inquiry format, the first extra character entered causes the keyboard to lock and the wrong length record indicator to turn on. The character that caused the keyboard to lock prints but does not enter storage. The release or cancel key must be pressed to continue the program.

Pressing the release key when the number of characters entered into storage is less than the prescribed format for the read operation causes a carrier return and line space; the wrong length record indicator is turned on, and the program goes to the next instruction. If the console-I/O printer is addressed on a load-write operation (L op code), blank characters in storage are printed as b's, and word marks are printed as inverted circumflexes 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 printer space operation generates a blank character in storage.

CANCEL KEY

If a request has been made (request key pressed) but has not been recognized (before any characters are entered), pressing the cancel key or release key will reset the console inquiry request indicator. If an error is recognized during typing (entering of data), pressing the cancel key turns on the condition indicator and the program continues. While a first inquiry is being entered, a second inquiry can be requested by holding down the inquiry request key while pressing the inquiry release key. This causes the inquiry status indicator to remain on.

CONSOLE I/O PRINTER WRITE OPERATION (REPLY ROUTINE)

A reply routine or programmed print-out can occur at any time. The console write instruction, M/L (% T 0) (BBBBB) W causes:

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

Figure 60 shows conditions that set I/O channel status indicators on and that turn on their associated lights during an I/O 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
No Transfer	A	Never set
Wrong Length Record	B	Never set

Figure 60. I/O Channel Status Indicators Set during Console I/O Printer Write Operation.

CPU Processing Error: A CPU processing error during data transfer ends the reply routine and initiates 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-I/O printer, the error character is printed and underlined and the data check I/O channel status indicator is set on. The reply routine continues until a group-mark-with-word-mark is sensed in storage.

Console-I/O Printer Keys and Levers

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. This printer has no type bars or movable carriage; instead, it has an interchangeable sphere-shaped type-head containing all the characters. The type-head moves from left to right across the paper during a printing operation. Characters are printed ten to the inch. Maximum printing rate is 932 characters per minute. 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. An 8.5-inch writing line is standard.

Figure 57 shows the 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.

WORD-MARK KEY

Pressing this key prints a word mark and backspaces the carriage after printing. Pressing a character key 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 when entering a character with a word mark into storage.

SHIFT KEYS

Pressing either one of the two shift keys shifts the printer into upper case. The printer automatically returns to lower-case shift when the shift key is released.

LOCK KEY

Pressing this key activates the shift keys and locks the printer in upper-case shift until released by pressing one of the shift keys.

COPY CONTROL LEVER

Operating the copy control lever at the left end of the carriage positions the carriage 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, and moving the lever to the rear increases this distance.

PAPER RELEASE LEVER

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

MARGIN SET LEVERS

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

INDEX SELECTOR LEVER

When the index selector lever (the outer lever at the right end of the carriage) is set toward the rear, the platen double spaces for each line of printing (three lines per inch). With the index selector lever set toward the front of the machine, the platen single spaces for each line of printing (six lines per inch).

Console Test Panel

The console test panel (Figure 61) is primarily for customer engineers for diagnostic testing and performing preventive maintenance routines. Certain functions of the panel, however, can be used advantageously by customer personnel when checking new program routines.

PRINT-OUT CONTROL SWITCH

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

*Trademark

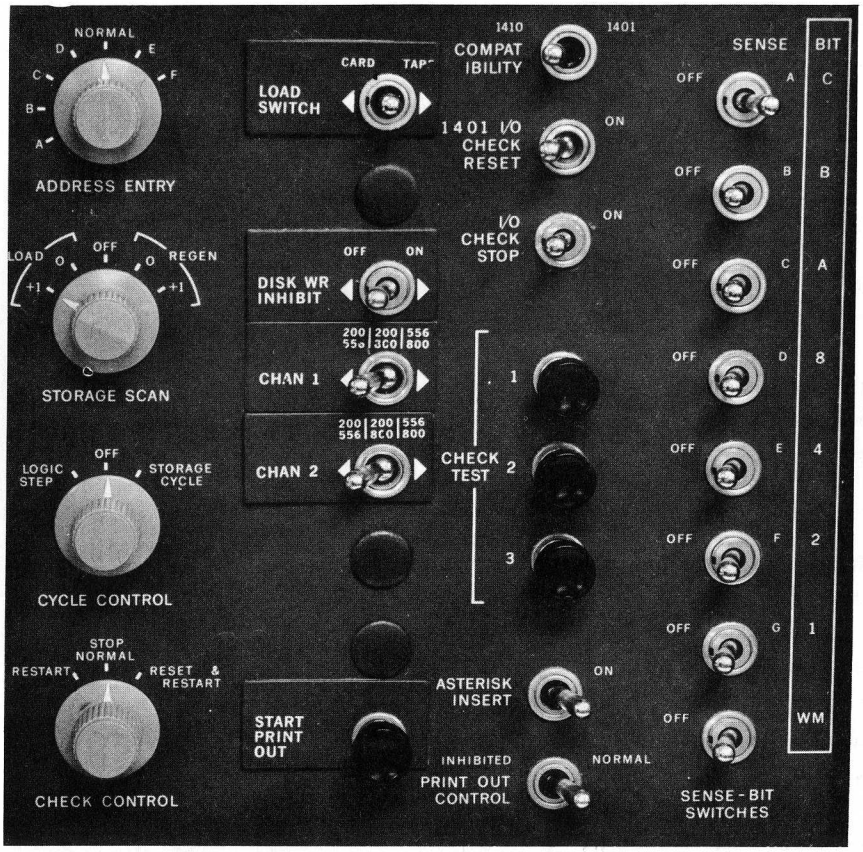


Figure 61. Test Panel

START PRINT-OUT SWITCH

This switch is used when the program routine fails to advance and a stop print-out operation cannot be initiated by pressing the stop key. Operating this switch initiates a stop print-out operation. Printing the contents of various registers aids in determining the cause of failure. This switch can also be used to initiate occasional print-outs while single cycling with print-out control off.

ASTERISK INSERT SWITCH

When this toggle switch is set to ON, any input unit character of incorrect parity is converted to an asterisk and entered 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 check control switch is set to STOP NORMAL, the data transfer stops. If the check control switch is set to RESTART, the full record can be entered into storage and used for diagnostic purposes or for reconstruction of the incorrect record.

COMPATIBILITY SWITCH

When this switch is set to 1401 position (with the 1401 Compatibility Feature), the 7010 will operate only in 1401 mode. When the switch is set to OFF, the system operates in the 7010 mode.

I/O CHECK STOP SWITCH

This switch is operative only when the 7010 is operating in 1401 mode. When the switch is on, the system will stop at the completion of any I/O operation during which an error occurred. 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 stored program controls the system in case of an 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 switch resets any I/O unit error conditions sensed when the I/O check stop switch is set to ON. (The switch is primarily used by customer engineers for diagnostic testing.)

SENSE-BIT SWITCHES

These eight switches allow any bit to be individually entered into storage scan operations (7010 mode). When set on in 1401 mode, the switches can be tested by a branch instruction.

LOAD KEY

The load key is a toggle switch on the console. This switch has a home position and may be pushed right to load from tape or left to load from the card reader. Only devices on channel 1 may be selected.

A load sequence operates as follows:

1. Put source data on the I/O device (card reader or tape unit) and put the unit in ready condition.
2. Press the load switch to either TAPE or CARD. This causes the 7010 to:
   a. Force computer reset.
   b. Set unoverlap, load mode for channel 1, and either select card reader and NR pocket or Select tape unit 0.
   c. Simulate I ring 12-time for status sample.
   d. Turn on channel 1 in process.
   e. Read end of record into storage location 0001.
   f. After completing the record, force program reset and auto start at 0001.

The programmer must check for correct data transfer.

CHECK TEST SWITCHES

These three switches enable the customer engineer to test the checking circuits in the system to determine if they are working properly.

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.

STORAGE SCAN-ROTARY

This switch is enabled when the mode switch is set to the CE position.

Normal: Storage addressing is under program control.

Regen +1: Storage addressing is modified by +1 and the contents of storage are regenerated.

Regen +0: Storage addressing is modified by +0 and the contents of storage are regenerated.

Load +1: Storage addressing is modified by +1 and the settings of the auxiliary console bit switches are loaded into storage.

Load +0: Storage addressing is modified by +0 and the settings of the auxiliary console bit switches are loaded into storage.

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 the 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.

CHECK CONTROL SWITCH

The check control switch is a three-position rotary switch. When it is set to STOP NORMAL and the asterisk insert switch is set OFF, any CPU error or input parity error 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 result in an 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 result in an immediate stop in the same manner as with 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.

Indicator Light Panel

The indicator light panel (Figure 62) contains lights that indicate the status, components in operation, and other conditions in the system.

Central Processing Unit Lights

I-RING

These lights indicate the 13 steps of the instruction ring (OP, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12).

CHARACTER GATE

Character gate 1 indicates that the first of the two sequential characters selected by the storage address is being gated to the CPU. Character gate 2 indicates that the second of the two sequential characters selected by the storage address is being gated to the CPU.

A-RING

These lights indicate the five steps of the A-ring (2, 3, 4, 5, and 6).

PROC (PROCESS)

This light indicates that the portion of a compute cycle normally devoted to processing a second character, is not disallowed.

CLOCK

These lights indicate the eight steps of the main clock (A, B, C, D, E, F, G, and H).

SCAN

These lights indicate what type of adress modification is taking place.

N: This light indicates that the CPU is readdressing a storage location (operating in a + 0 modification cycle). The address register is not being modified.

1: This light indicates that the CPU is in the (−1) address-modification cycle.

2: This light indicates that the CPU is in the (+1) address-modification cycle.

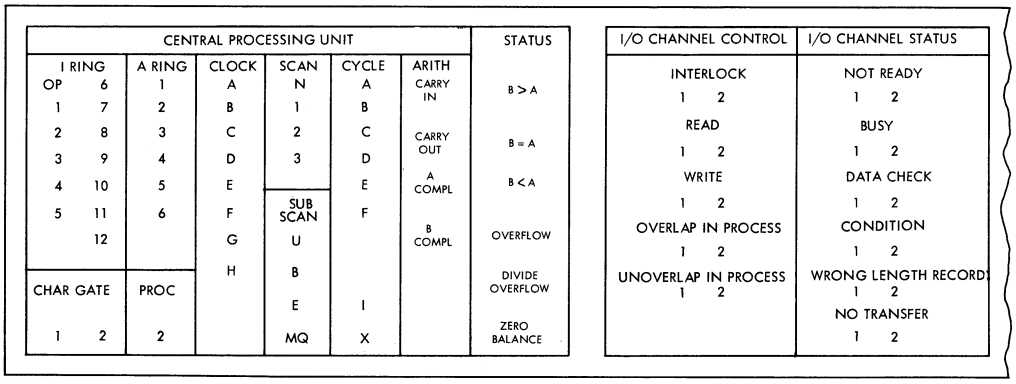
3: This light indicates that storage is being re-addressed and the CPU is operating in a −1 address-modification cycle.

SUB SCAN

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

U (Units): This light indicates that the units position of the field is being addressed during an arithmetic operation.

B (Body): This light indicates that the body of the field is being addressed during an arithmetic operation (excluding the units position of the field).



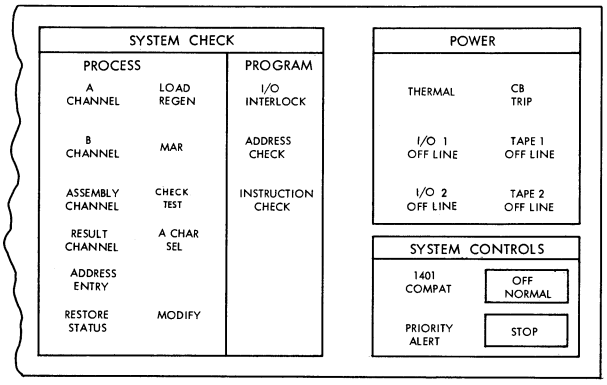


Figure 62. Sections of Indicator Light Panel

E (Extension): This light indicates that the extension portion of the field is being addressed.

MQ (Multiplier-Quotient): This light indicates that the multiplier or quotient is being addressed during a multiply or divide operation.

CYCLE

These lights indicate in which of the ten types of cycles the CPU is operating (A, B, C, D, E, F, G, H, I, or X).

ARITHMETIC

Carry In: This light indicates that the carry-in indicator has been set on.

Carry Out: This light indicates that the carry-out incator has been set on.

A Compl (A-Complement): This light indicates that the A-channel data are being complemented during the arithmetic operation.

B Compl (B-Complement): This light indicates that the B-channel data are being complemented during the arithmetic operation.

STATUS

B>A (High): This light indicates that the B-field is greater that the A-field. The light remains on until the condition is reset by a stored program operation, a computer reset, or a power-on reset.

B = A (Equal): This light indicates 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, or a power-on reset.

B < A (Low): This light indicates 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 stored program operation, a computer reset, or a power-on reset.

Overflow: This light indicates that an arithmetic overflow condition has occurred during an add or subtract operation. It cannot be turned on during a zero and add, zero and subtract, multiply, or divide operation. The light remains on until the condition is reset off by a stored program test operation, a computer reset, or a power-on reset.

Divide Overflow: This light indicates the occurrence of a divide overflow condition. The light remains on until the condition is reset off by a stored program test operation, a computer reset, or a power-on reset.

Zero Balance: This light, when on, indicates the occurrence of a zero-balance condition. It is set by any add, subtract, zero and add, zero and subtract, or multiply operation that terminates with a zero result. It remains on until the condition is reset by computer reset or power-on reset or by any add, subtract, zero and add, zero and subtract, or multiply operation that does not terminate in a zero result.

I/O Channel Control Lights

Lights glowing behind the numerals refer to the I/O channel whose control indicators are set on.

Interlock: This light indicates that either an I/O read or write operation has been called for. The light is turned off when the channel status test is satisfied following a read or write operation. If the status test is not satisfied before the next I/O instruction for that channel is called for, the system is interlocked and the interlock light remains on.

Read: This light indicates that an I/O read operation has been called for. It will remain on until the next I/O operation for that channel.

Write: This light indicates that an I/O write operation has been called for. It will remain on until the next I/O operation for that channel.

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

Unoverlap in Process: This light turns on at the beginning of any I/O operation that is not performed in 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 stops because of an error.

I/O Channel Status Indicator Lights

Lights glowing behind the numerals indicate the I/O status for the respective channels.

Not Ready: This light indicates that one of the units attached to the channel was not capable of taking a cycle when last selected.

Busy: This light indicates that one of the units attached to the channel had not completed a previous operation.

Data Check: This light indicates that one of the units attached to a channel has detected a data parity condition.

Condition: This light indicates that one of the units attached to the channel has encountered an end of file or data transfer control condition.

Wrong Length Record: This light indicates that one of the units attached to the channel has encountered or sent a wrong length record.

No Transfer: This light indicates that an operation of one of the units attached to the channel has resulted in a no-transfer condition.

System Check Indicator Lights

PROCESS

A-Channel: This light indicates that a parity error has been detected on the A-channel.

B-Channel: This light indicates that a parity error has been detected on the B-channel.

Assembly Channel: This light indicates that a parity error has been detected at the assembly output.

Result Channel: This light indicates that a parity error has been detected at the result channel output.

Address Entry: This light indicates that a validity error has been detected on the address entry channel.

Restore Status: This light indicates that a validity error was encountered during a restore status operation.

Load Regen: This light indicates that a compare of the load and regen controls in core storage did not result in a one and only one condition. When this light comes on, the memory cycle is forced to be a regen cycle.

MAR: This light indicates a validity check has been detected on the memory address register.

Check Test: This light indicates that a check circuit error has been detected.

A Char Sel: This light indicates that more than one character at a time has been gated to the A-channel.

Modify: This light indicates that a validity error has been detected on the modifier channel.

PROGRAM

I/O Interlock: This light indicates the program has failed to test the I/O channel status indicator before issuing the next I/O instruction on that channel.

Address Check: This light indicates that an improper core storage address has been given by the program or that an operation goes beyond the capacity of core storage.

Instruction Check: This light indicates that an improper instruction has been given by the program.

Power Indicator Lights

Thermal: If the internal temperature exceeds the allowable limit or if a circuit breaker in a blower trips, the power is turned off and this light is turned on.

CB Trip: If any circuit breaker in any power supply trips, the DC power is turned off and this light is turned on.

I/O 1 Off Line: This light indicates that the power is not being supplied to the buffer unit (1414) on channel 1 or the unit is operating in an off-line operation.

I/O 2 Off Line: This light indicates that the power is not being supplied to the buffer unit (1414) on channel 2 or the unit is operating in an off-line operation.

Tape 1 Off Line: This light indicates that power is not being supplied to the tape I/O synchronizer on channel 1 or the tape I/O synchronizer is operating in an off-line operation.

Tape 2 Off Line: This light indicates that power is not being supplied to the tape I/O synchronizer on channel 2 or the tape I/O synchronizer is operating in an off-line operation.

System Controls Lights

1401 Compat: This light indicates that the system is in the 1401 mode of operation (capable of running 1401 programs). It is turned on when the compatibility switch on the CE panel is in the ON position.

Priority Alert: When this light is on, the system is operating in the priority alter mode and is capable of being interrupted.

Off Normal: This red indicator indicates that certain CE switches on the console are not in proper position for normal operation. It is turned on if the print-out control switch is in INHIBIT position, if the asterisk insert switch is in ON position, if the cycle control switch is not in OFF position, if the check control switch is not in NORMAL position, if the storage scan switch is not in OFF position, or if the address entry switch is not in the NORMAL position.

Stop: This light indicates that the system has stopped. Operator intervention is required to start a new operation.

7010 Instruction Timings

Add, Subtract, Zero and Add, Zero and Subtract (two fields)

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4\,E + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 3.2\left(\frac{RB+1}{2}\right)$$

Add, Subtract, Zero and Add, Zero and Subtract (one field)

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 5.6\left(\frac{A+1}{2}\right)$$

Multiply

$$T \approx 2.4\left(\frac{L+2}{2}\right) + (2.5\,M+1)\left[5.6\left(\frac{A+2}{2}\right) + 6.4\right]$$

Divide

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 6.5\,Q\left[5.6\left(\frac{A+2}{2}\right) + 6.4\right]$$

Indexing

$T \backsimeq 9.6$ per address

Branch Unconditional

$$T \approx 2.4\left(\frac{L+2}{2}\right)$$

Test and Branch (conditional)

$$T \approx 2.4\left(\frac{L+2}{2}\right)$$

Test and Branch if Channel Status Indicator On

$$T \approx 2.4\left(\frac{L+2}{2}\right)$$

Test Character and Branch

$$T \approx 2.4\left(\frac{L+4}{2}\right)$$

Test Bit and Branch

$$T \approx 2.4\left(\frac{L+4}{2}\right)$$

Test Zone or WM and Branch

$$T = 2.4\left(\frac{L+4}{2}\right)$$

Data Move

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right)$$

Move and Zero Suppress

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right)$$

Compare

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4\left(\frac{A+1}{2}\right) + 3.2\left(\frac{B+1}{2}\right)$$

Edit

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 2.4\,A + 3.2\,B + 3.2\left(\frac{Z+1}{2}\right) + 3.2\left(\frac{D+1}{2}\right)$$

Table Lookup

$$T \approx 2.4\left(\frac{L+2}{2}\right) + 3.2\left(\frac{B+1}{2}\right) + 2.4\,N\left(\frac{A+1}{2}\right)$$

Store Address Register

$$T \approx 2.4 \left(\frac{L+2}{2}\right) + 9.6$$

Store and Restore Status

$$T \approx 2.4 \left(\frac{L+2}{2}\right) + 2.4$$

Set WM, Clear WM

$$T \approx 2.4 \left(\frac{L+2}{2}\right) + 6.4$$

Clear

$$T \approx 2.4 \left(\frac{L+2}{2}\right) + 3.2 \left(\frac{B+1}{2}\right)$$

Clear and Branch

$$T \approx 2.4 \left(\frac{L+2}{2}\right) + 3.2 \left(\frac{B+1}{2}\right)$$

Halt

$$T = 4.8$$

Halt and Branch

$$T \approx 2.4 \left(\frac{L+2}{2}\right)$$

No Op

$$T \approx 2.4 \left(\frac{L+2}{2}\right)$$

I/O M, L, F, K, 2, 4, (except file)

$$T \approx 2.4 (L+1) + I/O$$

I/O Unit Control

$$T \approx 2.4 (L+1) + TM$$

I/O M, L (File)

$$T \approx 2.4 (L+1) + \text{Address Transfer} + I/O$$

Timing Formula Symbols

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 re-complement is taken on an add or subtract operation. It is zero otherwise.
TM	tape movement
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.

I/O Channel Status Indicators (Figures 63 through 68)

INDICATOR	TELEGRAPH READ	TELEGRAPH WRITE	1009 READ	1009 WRITE
Not Ready	Power off in 1414 or buffer not on line. (No data transfer)	Power off in 1414, buffer not on line, or local telegraph not ready. (No transfer of data)	Power off in 1414, buffer not on line, 1009 not on line, or power off. No transfer of data	Power off in 1414, buffer not on line, 1009 not on line or power off. No transfer of data
Busy	Buffer filling	Buffer emptying	Buffer filling	Both buffers have data; one is emptying. Or, last segment of message is in 1 buffer
Data Check	Parity error, format check, or character pile-up between telegraph unit and 1414 or parity error between 1414 and 1410. Incorrect data stored as *, if asterisk switch is on	Parity error between 1410 core storage and 1414. No transfer to telegraph	Parity error between 1414 and core storage. Incorrect data stored as * if asterisk switch is on	Parity error between core storage and the 1414. Incorrect data arrived in 1414. (No transfer to local 1009)
Condition	Missed message. Buffer not emptied in time	Preceding message had parity or translate error between 1414 and telegraph	Missed message (buffer not emptied in time) or transmission error. (Indicator comes on only after end-of-message condition is recognized by the 1414)	Current message in error. Transmitted to local 1009, but not successfully to remote 1009
No Transfer	No request. No message in buffer to be read	Preceding message transmitted but received incorrectly or not at all because of invalid format line failure, or excessive delay in getting characters to the output line, or no group mark following the EOM sequence	End of message	End of message. (This comes on only after busy goes off)
Wrong Length Record	Incorrectly placed GM-WM. Data stored only to GM-WM. Remainder lost	Incorrectly placed GM-WM. Data transfer up to GM-WM, but only to the 1414 (not to telegraph)	Incorrectly placed GM-WM. Data stored only to GM-WM. Remainder lost	Incorrectly placed GM-WM. Data transfer up to GM-WM, but only to 1414, not to local 1009

Figure 63. I/O Channel Status Indicators – Telegraph and 1009

INDICATOR	1014 READ	1014 WRITE	1011 READ
Not Ready	Power off in 1414 or buffer not on line. (No data transfer)	Power off in 1414 or buffer not on line. (No data transfer)	Power off in 1414, buffer not on line. 1011 out of tape or tape is broken, or 1011 not attached to 1414 (no data transfer)
Busy	Not applicable	Buffer emptying	Buffer filling
Data Check	79 characters entered or parity error between 1414 and core storage. If parity error, incorrect data stored as * if asterisk switch is on	Parity error between core storage and 1414. No transfer to 1014.	Parity error between 1414 and core storage. Incorrect data stored as an * if asterisk switch is on
Condition	Machine check within 1414	Preceding message in error as received at station. Current message not transmitted	Not applicable
No Transfer	Buffer not full	Preceding message not transmitted; station inoperative (non-existent station, power off, station out of forms, or station didn't acknowledge preceding message)	Not applicable
Wrong Length Record	Incorrectly placed GM-WM. Data stored only to GM-WM. Remainder lost	Incorrectly placed GM-WM. Data transfer up to GM-WM, but only to 1414	Incorrectly placed GM-WM. Data stored only to GM-WM. Remainder lost

Figure 64. I/O Channel Status Indicators – 1014 and 1011

INDICATOR	7631/1301
Not Ready	Access inoperative or 7631 off-line 7631 power off Home address switch check
Busy	Access in motion 7631 not available (model 3)
Data Check	Parity check Check character code check Write disk check Format character check Invalid track number
Condition	Wrong length format No record found Write check without mode setting Disk storage circuit check File control circuit check Invalid operation code
No Transfer	No read or write operation performed (No data or address transferred)
Wrong Length Record	Short or long record

Figure 65. I/O Channel Status Indicators – 1301 Disk Storage

INDICATOR	1311 DISK STORAGE DRIVE
Not Ready	Unsafe condition, 1311 power off, 1311 disconnected, drive motor off (disk pack being changed). Write address key light in wrong position
Busy	Any access in motion (without seek overlap). Addressed access in motion (with seek overlap)
Data Check	Parity check, write disk check
Condition	No record found, absence of sector address match subsequent to sector address match
No Transfer	No read or write operation performed
Wrong Length Record	Short or long length record

Figure 66. I/O Channel Status Indicators – 1311 Disk Storage Drive

INDICATOR	7750 PROGRAMMED TRANSMISSION CONTROL
Not Ready	7750 not ready
Busy	Not used
Data Check	Unusual-end signal from 7750. Input parity error in 7010
Condition	7750 failed to terminate previous operation correctly. 7750 unable to initiate operation
No Transfer	Data transfer incomplete. 7750 became inoperative during data transfer
Wrong Length Record	Data field actually transferred was different in length from either the field in storage or the field in 7750. On input, data are stored to ≑; remainder is lost

Figure 67. I/O Channel Status Indicators — 7750 Programmed Transmission Control

INDICATOR	1440 DATA PROCESSING SYSTEM
Not Ready	1440 not ready
Busy	Not used
Data Check	An unusual end signal from the 1440 Input parity error in 7010
Condition	1440 failed to terminate previous operation correctly after sending an end or unusual end signal to the 7010. 1440 unable to initiate the operation
No Transfer	The 7010 became non-operational during data transfer. Data transfer incomplete
Wrong Length Record	Input: 1. 1440 defined length longer than 7010 defined length. Data stored in 7010 only to ≑. Remainder lost. 7010 has detected ≑, but no ≑ detected by 1440. 2. 1440 defined length shorter than 7010 defined length. End or unusual-end signal from 1440, but no ≑ detected by 7010. Output: 1. 7010 defined length longer than 1440 defined length. End or unusual-end signal from 1440 but no ≑ detected by 7010. Note: The 7010 cannot detect a wrong length during output if the 7010 defined length is shorter than the 1440 defined length

Figure 68. I/O Channel Status Indicators — 1440 Data Processing System

Alphabetic Listing of 7010 Instructions (Indexed)

NOTE: For key to symbols, see footnotes at end of listing.

INSTRUCTION	MNEMONIC	OPERAND	FORM	PAGE
Add (One Field)	A	a	A(A)	15
Add (Two Fields)	A	a b	v A(A)(B)	14
Backspace Tape	BSP	cu	v UxxxB	49
Branch and Set Priority Alert Mode	BEPA	i	v Y(I)E	39
Branch and Reset Priority Alert Mode	BXPA	i	v Y(I)X	39
Branch if Arithmetic Overflow	BAV	i	v J(I)Z	31
Branch if Bit Equal (any bit in b matches a bit in d)	BBE	i b d	v W(I)(B)d	32
Branch if Carriage Busy (Ch 1 or 2)	BPCP#	i	v J(I) [R or L]	38
Branch if Carriage 9 (Ch 1 or 2)	BC9#	i	v J(I) [9 or !]	38
Branch if Carriage Overflow, 12 (Ch 1 or 2)	BCV#	i	v J(I) [@ or □]	38
Branch if Any I/O Channel Status Indicator On (Ch 1 or 2)	BA#	i	v R or v X(I)≠	37
Branch if Overlap in Process (Ch 1 or 2)	BOL#	i	v J(I) [1 or 2]	38
Branch if I/O Unit Not Ready (Ch 1 or 2)	BNR#	i	v R or v X(I)1	37
Branch if I/O Unit Busy (Ch 1 or 2)	BCB#	i	v R or v X(I)2	37
Branch if I/O Unit Data Check (Ch 1 or 2)	BER#	i	v R or v X(I)4	37
Branch if I/O Unit Condition (Ch 1 or 2)	BEF#	i	v R or v X(I)8	37
Branch if I/O Wrong Length Record (Ch 1 or 2)	BWL#	i	v R or v X(I)–	37
Branch if I/O Unit No Transfer (Ch 1 or 2)	BNT#	i	v R or v X(I)⅋	37
Branch if Any On in Plural Indicator Test	BEX#	i d	v R or v X(I)d	37
Branch if Character Equal (b = d)	BCE	i b d	v B(I)(B)d	32
Branch if Compare Equal	BE	i	v J(I)S	31
Branch if Compare High (B greater than A)	BH	i	v J(I)U	31
Branch if Compare Low (B less than A)	BL	i	v J(I)T	31
Branch if Compare Equal	BU	i	v J(I)/	31
Branch if Divide Overflow	BDV	i	v J(I)W	31
Branch if Inquiry Request (Ch 1 or 2)	BNQ#	i	v J(I)[Q or *]	38
Branch if Zero Balance	BZ	i	v J(I)V	31
Branch if WM Present	BW	i b	v V(I)(B)1	32
Branch if WM Present, or Zone Bits Absent	BWZ	i b	v V(I)(B)3	32
Branch if WM Present, or Zone Equal A	BWZ	i b A	v V(I)(B)T	32
Branch if WM Present, or Zone Equal AB	BWZ	i b AB	v V(I)(B)C	32
Branch if WM Present, or Zone Equal B	BWZ	i b B	v V(I)(B)L	32
Branch if Zone Bits Absent	BZN	i b	v V(I)(B)2	32
Branch if Zone Equal A	BZN	i b A	v V(I)(B)S	32
Branch if Zone Equal AB	BZN	i b AB	v V(I)(B)B	32
Branch if Zone Equal B	BZN	i b B	v V(I)(B)K	32
Branch on Attention (Ch 1 or 2)	BXPR#	i	v Y(I)[A or B]	39
Branch on Inquiry (Ch 1 or 2)	BIPR#	i	v Y(I)[Q or *]	39
Branch on Outquiry (Ch 1 or 2)	BQPR#	i	v Y(I)[N or ∔]	39
Branch on Overlap Complete (Ch 1 or 2)	BOPR#	i	v Y(I)[1 or 2]	39
Branch on Seek Complete (Ch 1 or 2)	BSPR#	i	v Y(I)[S or T]	39
Branch on Selected I/O Unit (Ch 1 or 2)	BUPR#	i	v Y(I)[U or F]	39
Branch Unconditionally	B	i	v J(I)blank	31
Carriage Control Immediate Skip to (1-9) (Ch 1 or 2)	CC#	(1 to 9)	v F or v 2 [1 to 9]	46
Carriage Control Immediate Skip to 10 (10-12) (Ch 1 or 2)	CC#	(0 # @)	v F or v 2 [0 #@]	46
Carriage Control Immediate (1 to 3) Spaces (Ch 1 or 2)	CC#	(J K L)	v F or v 2 [J K L]	46

INSTRUCTION	MNEMONIC	OPERAND	FORM	PAGE
Carriage Control Skip after Print to (1-9) (Ch 1 or 2)	CC#	(A to I)	v F or 2 [A to I]	46
Carriage Control Skip after Print to (10, 11, 12) (Ch 1 or 2)	CC#	(?.□)	v F or 2 [?.□]	46
Carriage Control (1, 2, 3) Space After Print (Ch 1 or 2)	CC#	(/ S T)	v F or 2 [/ S T]	46
Clear Storage	CS	b	v /(B)	34
Clear Storage and Branch	CS	i b	v /(I)(B)	35
Clear Word Mark (One Address)	CW	a	v □(A)	34
Clear Word Mark (Two Addresses)	CW	a b	v □(A)(B)	34
Compare (b to a)	C	a b	v C(A)(B)	22
Divide (a into b)	D	a b	v %(A)(B)	18
Erase Forward (Skip and Blank Tape)	SKP	cu	v UxxxE	
Halt	H		v .	35
Halt and Branch	H	i	v .(I)	35
Lookup Equal	LE	a b	v T(A)(B)2	25
Lookup Equal or High	LEH	a b	v T(A)(B)6	25
Lookup High	LH	a b	v T(A)(B)4	25
Lookup Low	LL	a b	v T(A)(B)1	25
Lookup Low or Equal	LLE	a b	v T(A)(B)3	25
Lookup Low or High	LLH	a b	v T(A)(B)5	25
Lookup to Any	----	---	v T(A)(B)7	25
Lookup to End	----	---	v T(A)(B)blank	25
Move Characters and Edit	MCE	a b	v E(A)(B)	26
Move Characters and Suppress Zeros	MCS	a b	v Z(A)(B)	22
Move Left Characters and WM Single Position	MLCWS	a b	v D(A)(B)7	20
Move Left Characters and WM thru 1st A-Field WM	MLCWA	a b	v D(A)(B)X	20
Move Left Characters and WM thru 1st B-Field WM	MLCWB	a b	v D(A)(B)P	20
Move Left Characters and WM thru 1st WM	MLCW	a b	v D(A)(B)G	20
Move Left Characters Single Position	MLCS	a b	v D(A)(B)3	20
Move Left Characters thru 1st A-Field WM	MLCA	a b	v D(A)(B)T	20
Move Left Characters thru 1st B-Field WM	MLCB	a b	v D(A)(B)L	20
Move Left Characters thru 1st WM	MLC	a b	v D(A)(B)C	20
Move Left Numeric and WM Single Position	MLNWS	a b	v D(A)(B)5	20
Move Left Numeric and WM thru 1st A-Field WM	MLNWA	a b	v D(A)(B)V	20
Move Left Numeric and WM thru 1st B-Field WM	MLNWB	a b	v D(A)(B)N	20
Move Left Numeric and WM thru 1st WM	MLNW	a b	v D(A)(B)E	20
Move Left Numeric Single Position	MLNS	a b	v D(A)(B)1	20
Move Left Numeric thru 1st A-Field WM	MLNA	a b	v D(A)(B)/	20
Move Left Numeric thru 1st B-Field WM	MLNB	a b	v D(A)(B)J	20
Move Left Numeric thru 1st WM	MLN	a b	v D(A)(B)A	20
Move Left WM Single Position	MLWS	a b	v D(A)(B)4	20
Move Left WM thru 1st A-Field WM	MLWA	a b	v D(A)(B)U	20
Move Left WM thru 1st B-Field WM	MLWB	a b	v D(A)(B)M	20
Move Left WM thru 1st WM	MLW	a b	v D(A)(B)D	20
Move Left Zones and WM Single Position	MLZWS	a b	v D(A)(B)6	20
Move Left Zones and WM thru 1st A-Field WM	MLZWA	a b	v D(A)(B)W	20
Move Left Zones and WM thru 1st B-Field WM	MLZWB	a b	v D(A)(B)O	20
Move Left Zones and WM thru 1st WM	MLZW	a b	v D(A)(B)F	20
Move Left Zones Single Position	MLZS	a b	v D(A)(B)2	20

INSTRUCTION	MNEMONIC	OPERAND	FORM	PAGE
Move Left Zones thru 1st A-Field WM	MLZA	a b	v D(A)(B)S	20
Move Left Zones thru 1st B-Field WM	MLZB	a b	v D(A)(B)K	20
Move Left Zones thru 1st WM	MLZ	a b	v D(A)(B)B	20
Move Right Characters and WM thru 1st A-Field ǂ	MRCWG	a b	v D(A)(B)△	20
Move Right Characters and WM thru 1st A-Field ǂ	MRCWR	a b	v D(A)(B)+++	20
Move Right Characters and WM thru 1st A-Field ǂ or ǂ	MRCWM	a b	v D(A)(B)ǂ	20
Move Right Characters and WM thru 1st WM	MRCW	a b	v D(A)(B)∨	20
Move Right Characters thru 1st A-Field ǂ	MRCG	a b	v D(A)(B)$	20
Move Right Characters thru 1st A-Field ǂ	MRCR	a b	v D(A)(B),	20
Move Right Characters thru 1st A-Field ǂ or ǂ	MRCM	a b	v D(A)(B).	20
Move Right Characters thru 1st WM	MRC	a b	v D(A)(B)#	20
Move Right Numeric and WM thru 1st A-Field ǂ	MRNWG	a b	v D(A)(B)]	20
Move Right Numeric and WM thru 1st A-Field ǂ	MRNWR	a b	v D(A)(B)ₘ	20
Move Right Numeric and WM thru 1st A-Field ǂ or ǂ	MRNWM	a b	v D(A)(B)[	20
Move Right Numeric and WM thru 1st WM	MRNW	a b	v D(A)(B):	20
Move Right Numeric thru 1st A-Field ǂ	MRNG	a b	v D(A)(B)R	20
Move Right Numeric thru 1st A-Field ǂ	MRNR	a b	v D(A)(B)Z	20
Move Right Numeric thru 1st A-Field ǂ or ǂ	MRNM	a b	v D(A)(B)I	20
Move Right Numeric thru 1st WM	MRN	a b	v D(A)(B)9	20
Move Right WM thru 1st A-Field ǂ	MRWG	a b	v D(A)(B)*	20
Move Right WM thru 1st A-Field ǂ	MRWR	a b	v D(A)(B)%	20
Move Right WM thru 1st A-Field ǂ or ǂ	MRWM	a b	v D(A)(B)□	20
Move Right WM thru 1st WM	MRW	a b	v D(A)(B)@	20
Move Right Zones and WM thru 1st A-Field ǂ	MRZWG	a b	v D(A)(B);	20
Move Right Zones and WM thru 1st A-Field ǂ	MRZWR	a b	v D(A)(B)\	20
Move Right Zones and WM thru 1st A-Field ǂ or ǂ	MRZWM	a b	v D(A)(B)<	20
Move Right Zones and WM thru 1st WM	MRZW	a b	v D(A)(B)>	20
Move Right Zones thru 1st A-Field ǂ	MRZG	a b	v D(A)(B)!	20
Move Right Zones thru 1st A-Field ǂ	MRZR	a b	v D(A)(B)ǂ	20
Move Right Zones thru 1st A-Field ǂ or ǂ	MRZM	a b	v D(A)(B)?	20
Move Right Zones thru 1st WM	MRZ	a b	v D(A)(B)0	20
Multiply	M	a b	v @(A)(B)	17
No Operation	NOP		v N	35
Punch a Card, Stack in Pocket 0	P#WO	0 b	v M or v L x40(B)W	44
Punch a Card, Stack in Pocket 4	P#WO	4 b	v M or v L x44(B)W	44
Punch a Card, Stack in Pocket 8	P#WO	8 b	v M or v L x48(B)W	44
Read a Card, Stack in Pocket 0	R#WO	0 b	v M or v L x10(B)R	43
Read a Card, Stack in Pocket 1	R#WO	1 b	v M or v L x11(B)R	43
Read a Card, Stack in Pocket 2	R#WO	2 b	v M or v L x12(B)R	43
Read a Card, No Stack or Feed Operation	R#WO	9 b	v M or v L x19(B)R	43
Read Console Printer (Ch 1 only)	RCPWO	b	v M or v L xT0(B)R	54
Read Tape	RTWO	cu b	v M or v L xUx(B)R	47, 49
Read Tape to IRG or End of Core	RTGW	cu b	v M or v L xUx(B)$	48
Read Tape Binary	RTBWO	cu b	v M or v L xBx(B)R	48
Read Tape Binary to IRG or End of Core	RTBGW	cu b	v M or v L xBx(B)$	48
Rewind	RWD	cu	v Uxxx(B)R	49
Rewind and Unload	RWU	cu	v Uxxx(B)U	49
Restore Channel Status (Ch 1 or 2)	----	---	v $(B)[1 or 2]	33

INSTRUCTION	MNEMONIC	OPERAND	FORM	PAGE
Restore Machine Status	----	---	v $(B)R	33
Scan Left Single Position	SCNLS	a b	v D(A)(B)blank	20
Scan Left thru 1st A-Field WM	SCNLA	a b	v D(A)(B)⅋	20
Scan Left thru 1st B-Field WM	SCNLB	a b	v D(A)(B)–	20
Scan Left thru 1st WM	SCNL	a b	v D(A)(B)&	20
Scan Right thru 1st A-Field ǂ	SCNRG	a b	v D(A)(B)Q	20
Scan Right thru 1st A-Field ǂ	SCNRR	a b	v D(A)(B)Y	20
Scan Right thru 1st A-Field ǂ or ǂ	SCNRM	a b	v D(A)(B)H	20
Scan Right thru 1st WM	SCNR	a b	v D(A)(B)8	20
Select Stacker 0 and Feed (Ch 1 or 2)	SSF#	(0 b1)	v K or v 4 0	43
Select Stacker 1 and Feed (Ch 1 or 2)	SSF#	1	v K or v 4 1	43
Select Stacker 2 and Feed (Ch 1 or 2)	SSF#	2	v K or v 4 2	43
Set Word Mark (One Address)	SW	a	v ,(A)	34
Set Word Mark (Two Addresses)	SW	a b	v ,(A)(B)	34
Skip and Blank Tape	SKP	cu	v U xxx E	49
Store A-Address Register	SAR	a	v G(C)A	33
Store B-Address Register	SBR	a	v G(C)B	33
Store E-Address Register	SER	a	v G(C)E	33
Store F-Address Register	SFR	a	v G(C)F	33
Store Channel Status (Ch 1 or 2)	----	---	v $(B)E or F	33
Store Machine Status	----	---	v $(B)S	33
Subtract (One Field)	S	a	v S(A)	15
Subtract (Two Fields)	S	a b	v S(A)(B)	15
Table Lookup (see Lookup)				
Test and Branch (see Branch)				31
Unit Control (see Backspace, Skip, Write, Rewind)				
Write a Line	W#WO	b	v M or v L x20(B)W	45
Write Console Printer (Ch 1 only)	WCPWO	b	v M or v L xTO(B)W	55
Write Printer (see Write a Line)				
Write Tape	WTWO	cu b	v M or v L xUx(B)W	47, 49
Write Tape to End of Core	WTEW	cu b	v M or v L xUx(B)X	48
Write Tape Binary	WTBWO	cu b	v M or v L xBx(B)W	48
Write Tape Binary to End of Core	WTBEW	cu b	v M or v L xBx(B)X	48
Write Tape Mark	WTM	cu	v U xxx M	49
Write Word Marks as 1's	WM#O	b	v M x 21(B)W	46
Zero and Add (One Field)	ZA	a	v ?(A)	16
Zero and Add (Two Fields)	ZA	a b	v ?(A)(B)	16
Zero and Subtract (One Field)	ZS	a	v !(A)	17
Zero and Subtract (Two Fields)	ZS	a b	v !(A)(B)	16

Instruction Listing Footnotes:

MNEMONIC OP-CODE SUFFIXES

- # 1 or 2 for channel
- W W if WM (Load Mode)
- O O if Overlap

OPERANDS

- a A-address
- b B-address
- i I-address
- d d-character
- c 1 or 2 for channel
- u I/O Unit Number
- ( ) Parenthesis defines d-character choices
  All other characters actual

INSTRUCTION FORM

- [ ] Brackets define d-character choices
- d Symbolizes d-character
- x Symbolizes x-control field character

Index to Figures

The figures in this list are those most likely to be referred to. Not all figures are listed.

Index

CUT ALONG LINE

COMMENT SHEET

IBM 7010 DATA PROCESSING SYSTEM

PRINCIPLES OF OPERATION, FORM A22-6726

FROM

NAME _____

OFFICE _____

FOLD

FOLD

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

☐ 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: CE MANUALS, DEPARTMENT B98

CUT ALONG LINE

FOLD

FOLD

STAPLE

STAPLE