

Reference Manual

CONTROL DATA[®]
160-A COMPUTER

CONTROL DATA
CORPORATION

CONTENTS

CHAPTER 1 – DESCRIPTION

160-A Characteristics	1 - 1
Physical Description	1 - 2
Registers	1 - 2
Addressable Registers	1 - 2
Non-Addressable Registers	1 - 4

CHAPTER 2 – DESCRIPTION OF INSTRUCTIONS

Arithmetic	2 - 1
Notation	2 - 2
Addressing Modes	2 - 3
Word Format	2 - 4
Storage Control	2 - 9

CHAPTER 3 – OPERATION

160-A Computer Console	3 - 1
Register Display	3 - 1
Starting the 160-A	3 - 6
Loading a Program or Data in Paper Tape Load Format	3 - 6
Entering Data from the Console	3 - 7
Examining the Contents of Storage at the Console	3 - 7
Clearing an Entire Storage Bank	3 - 7
Operating the 350 Reader	3 - 8
Tape Loading	3 - 8
Splicing Paper Tape	3 - 8
Operating the Paper Tape Punch	3 - 9
Loading a New Roll of Tape	3 - 9

CHAPTER 3 – OPERATION (Cont'd)

160-A Instructions	3 - 11
Stop Instructions	3 - 11
Data Transmission Instructions	3 - 11
Arithmetic Instructions	3 - 16
Shift Instructions	3 - 19
Logical Instructions	3 - 20
Storage Bank Control Instructions	3 - 21
Jump Instructions	3 - 23
Input/Output Instructions	3 - 25
Selective Stop and Jump Instructions	3 - 29
Using the Instructions	3 - 30
Interrupt	3 - 30
Input/Output	3 - 32
Programming Examples	3 - 37
Internal Programming	3 - 37
Input/Output Programming	3 - 39

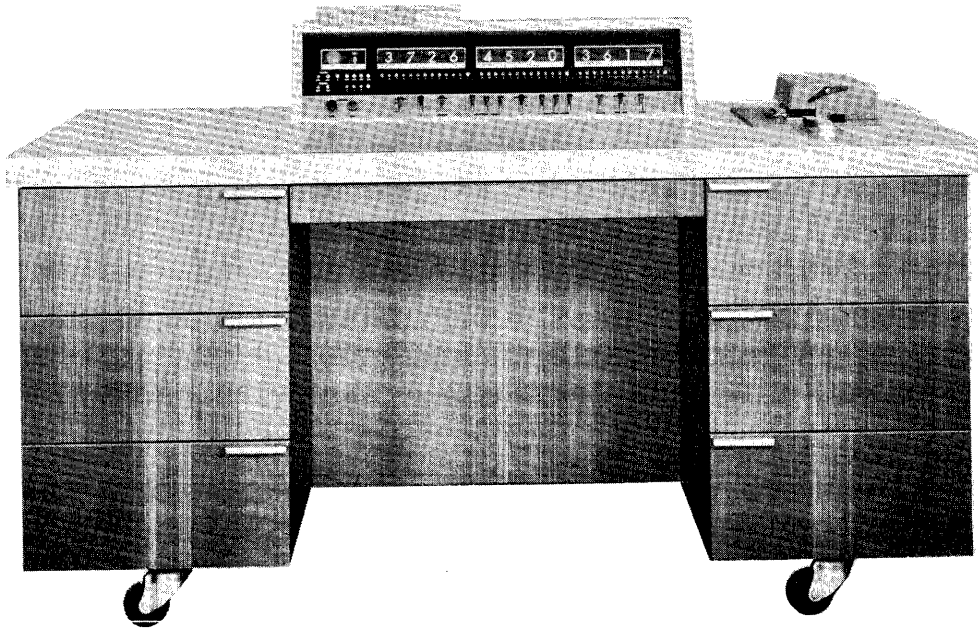
GLOSSARY OF PROGRAMMING TERMS

APPENDIXES

1 Table of Instructions	A - 1
2 Table of External Function Codes and Status Responses	A - 5
3 Table of Powers of 2	A - 12
4 Octal - Decimal Integer Conversion Table	A - 13
5 Octal - Decimal Fraction Conversion Table	A - 17
6 Input/Output Typewriter Codes	A - 20
7 1612 Printer Codes	A - 21
8 166 Printer Codes	A - 23
9 Flexowriter Codes	A - 27
10 Magnetic Tape BCD Codes	A - 28
11 Punched Card Codes	A - 29

FIGURES

1	Block Diagram of Registers	1 - 3
2	160-A Console	3 - 1
3	Paper Tape Reader	3 - 8
4	Tape Splices	3 - 9
5	Paper Tape Punch	3 - 10
6	Punched Paper Tape Levels	3 - 40



160-A COMPUTER

Chapter I

DESCRIPTION

The CONTROL DATA* 160-A Computer is a small (desk size), highly flexible, stored program computer with many features normally found only in large scale computers. Using a high speed data transfer channel, the 160-A may communicate with other Control Data computers, allowing multi-computer processing and computer control of computers. Two computers may share input/output equipment such as magnetic tape stations to assure high computing performance at a minimum cost.

160-A CHARACTERISTICS

Buffered input/output	A 350 character/second paper tape reader
Internal and external interrupt	A 110 character/second paper tape punch
An expandable magnetic core memory (8,192 to 32,768 words)	Transistor-diode logic
A 160-A system may be expanded to include the following external equipment:	
Up to 40 magnetic tape stations	Analog-to-digital and digital-to-analog equipment
Input/output typewriters	Two or more Control Data computers operating together in a satellite system
Punched card readers, punches, and low speed line printers	Two CONTROL DATA 160-A Computers operating independently but sharing the same magnetic core memory
High speed (1000 lines/minute) printers	
Plotting and digital display equipment	

*Registered trademark of Control Data Corporation

PHYSICAL DESCRIPTION

Physical characteristics and power requirements are:

Height - 29 inches

Width - 30 inches

Length - 61 $\frac{5}{8}$ inches

Weight - 810 pounds

Power - 26 amps, 110v, 60 cycle, single phase

The 160-A is a parallel, single address, electronic data processor. An internally stored program located in sequential addresses controls operation. The basic memory of the 160-A consists of two banks of magnetic core storage, each with a capacity of 4096 12-bit binary words and a storage cycle time of 6.4 usec. This basic memory may be expanded in modules of 8,192 words up to a maximum of 32,768 words. The bank approach permits increasing the storage capacity of the computer without increasing the size of the 12-bit storage address, and is implemented by executing storage bank control commands at the beginning of a program. Bank selections may be changed at any time during the program. Instructions are executed in one to four storage cycles; the time varies between 6.4 and 25.6 usec. An instruction is a 12-bit word comprising a 6-bit function or operation code (F) and a 6-bit code extension and/or execution address (E). The average program execution time for the 130 instructions is approximately 15 usec per instruction.

A CONTROL DATA 350 Paper Tape Reader is located in the desk top of the computer to the right of the display console, and a Teletype BRPE-11 paper tape punch is housed in the left pedestal of the computer cabinet.

REGISTERS

The registers are the storage units for all data transmission and computation in the 160-A. Contents of all but one of the registers are displayed on the console. Those registers available to the programmer by means of computer instructions are called addressable registers; the others are called non-addressable (figure 1). Unless otherwise specified, all registers are 12 bits long.

ADDRESSABLE REGISTERS

- | | |
|------------|---|
| A Register | The A register receives the results of all arithmetic, logical, and shifting operations. |
| P Register | The P register contains the storage address of the current instruction. At the completion of an instruction which does not require program control to jump out of sequence, the P register is advanced by 1 or 2 to select the address of the next instruction. If control is to be transferred out of sequence the new control address is sent to the P register |

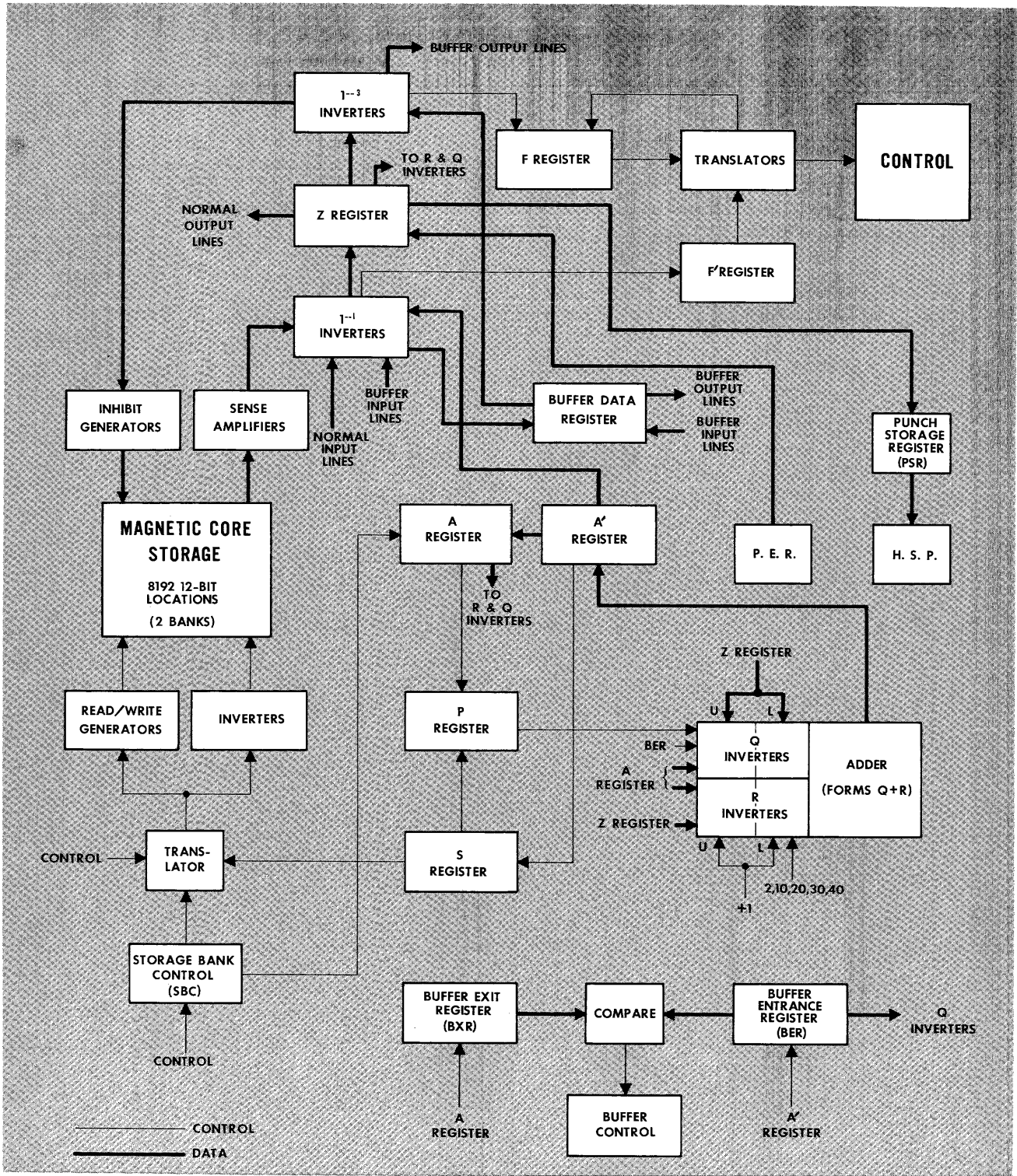


Figure 1. Block Diagram of Registers

and the P register is not advanced. If the P register contains 7776 and is advanced by 1 it will contain 0000; if P contains 7777, advancing it by 1 will change the number to 0001.

Buffer Entrance Register (BER) During buffered input/output operations, BER holds the address to or from which information is being transferred. The contents of this register may be transferred to storage and to the A register.

Buffer Exit Register (BXR) During buffered input/output operations, BXR holds the last word address + 1 to or from which information is being transferred.

NON-ADDRESSABLE REGISTERS

Z Register Z, a transient register, holds information during transfers between storage and peripheral equipment on the normal input/output channel. The Z register also restores information read out of storage.

S Register The S register contains the storage address currently being referenced either for an instruction or for an operand.

F Register The F register contains the decoded instruction being executed. The upper 6 bits contain the effective function code and, in general, the lower 6 bits the effective E portion of the instruction.

Buffer Data Register (BFR) During a buffer operation, BFR holds the word of information being transferred to or from storage.

A' Register The A' register acts as an auxiliary transmission register for information moving between the other registers. This is the output register of the borrow pyramid (adder).

Punch Storage Register (PSR) During output operations which use the paper tape punch, the 8-bit PSR releases the Z register, thereby permitting high-speed computation while the output characters are punched at a comparatively slow rate. The contents of PSR cannot be indicated on the display panel.

Chapter II

DESCRIPTION OF INSTRUCTIONS

ARITHMETIC

A 160-A computer word contains 12 binary digits. The bits within a computer word are numbered from 00 (least significant) to 11, starting on the right. All arithmetic is binary, one's complement notation. Any number may be represented as combinations of the two binary digits, 0 and 1. Although the computer operates in the binary system, the octal representation of a binary number is more convenient. The 160-A word can be considered as four octal digits. The following diagram shows the bit position numbering, the binary representation of a 12-bit quantity, and the octal equivalent of that quantity.

11	10	09	08	07	06	05	04	03	02	01	00
0	1	1	1	1	0	0	0	1	0	1	1
3			6			1			3		

In one's complement notation, positive numbers are represented by their binary equivalent; negative numbers are represented by the one's complement of the equivalent positive number. To form the one's complement, reverse each bit of the word.

Example: +5 is represented as: 000 000 000 101
 -5 is represented as: 111 111 111 010

The internal arithmetic of the computer is based on subtraction. Addition is performed by subtracting the complement of the addend from the augend. In subtraction no complementing is necessary.

Example:

The computer is programmed to add +6 to +5. This operation is performed as follows:

$$\begin{array}{r}
 +5 = 000\ 000\ 000\ 101 \\
 +6 = 000\ 000\ 000\ 110 \\
 \text{Complementing } +6 \text{ produces} \\
 \text{and subtraction takes place,} \\
 \quad 111\ 111\ 111\ 001 \\
 \quad +5 = 000\ 000\ 000\ 101 \\
 - \text{ } (+6) = \underline{111\ 111\ 111\ 001} \\
 \text{Borrow (1) } 000\ 000\ 001\ 100 \\
 \\
 \text{Subtract the borrow} \quad - \quad \underline{\quad \quad \quad 1} \\
 \text{produces} \quad 000\ 000\ 001\ 011 = 13_8, \text{ the} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{corrected} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{answer}
 \end{array}$$

During the subtraction process, the borrow from the high order end was carried around and subtracted from the low order end of the word to provide the corrected result. This "end-around-borrow" is the feature which makes the arithmetic of the 160-A modulus $2^{12}-1$.

The value zero can be represented in one's complement notation in either of two separate expressions:

$$\begin{array}{l}
 000\ 000\ 000\ 000 \text{ (plus zero)} \\
 111\ 111\ 111\ 111 \text{ (minus zero)}
 \end{array}$$

Both plus and minus zero are acceptable as arithmetic operands. There are only two cases in which a zero arithmetic result will be minus zero; in all other cases, the result will be plus zero. These two cases are:

$$-0 + (-0) \text{ and } -0 - (+0)$$

All positive numbers must have a "0" in bit 11; all negative numbers must have a "1" in bit 11. As is implicit in the illustrations above, the sign bit is extended to the most significant bit of the number.

NOTATION

The following abbreviations are used throughout the remainder of this manual:

- A A register
- P P register
- Z Z register
- BER Buffer Entrance register
- BXR Buffer Exit register

BFR	Buffer Data register
E	6 low order bits in the first word of a 160-A instruction
F	6 high order bits in the first word of a 160-A instruction
G	12 bits in the second word of all two-word 160-A instructions
X, Y	Any octal digit, from 0 through 7
XXXX	An octal operand or EF code
YYYY	An octal address
()	Contents of whatever location or register is specified within the parentheses. The only exception to this is reference to a specific storage bank control in its numeric value. In this case, reference is to the storage bank number, e.g., (0) refers to storage bank zero. This is the only time a single digit will be enclosed in parentheses.
(d)	Contents of the direct storage bank control
(r)	Contents of the relative storage bank control
(i)	Contents of the indirect storage bank control
(b)	Contents of the buffer storage bank control
→	Function or quantity to the left of the arrow replaces the function or quantity to the right.
FWA	First word address. This term is used when reference is made to any block of data. Also, the core storage address of the first word of such a block.
LWA	Last word address. This term is used when reference is made to any block of data. Also, the core storage address of the last word of such a block.

ADDRESSING MODES

Eight addressing modes provide maximum flexibility of the 6-bit address:

No Address (N)	When the E portion of the instruction is used as a 6-bit operand, it performs arithmetic and logical operations with a 6-bit constant contained in the instruction. This mode eliminates the need for entering many constants into memory.
Direct Address (D)	Refers to a 12-bit operand in one of the first 64 (100 octal) storage locations.
Indirect Address (I)	Provides for operand references and jump addresses. Instructions employing indirect addressing use E to refer to one of 77 of the first 100 octal storage locations starting at location 0001. The contents of this address are used as the address of the operand or as the jump address.

Relative Address Forward (F) Operand or jump address is obtained by adding E to the current contents of P to specify one of the 63 (77 octal) addresses immediately following the address of the current instruction.

Relative Address Backward (B) Operand or jump address is obtained by subtracting E from the current contents of P to specify one of the 63 (77 octal) addresses immediately preceding the address of the current instruction.

Specific Address (S) The effective address is always location 7777 (octal) in storage bank zero. The E portion of the instruction word is always equal to zero in the specific mode.

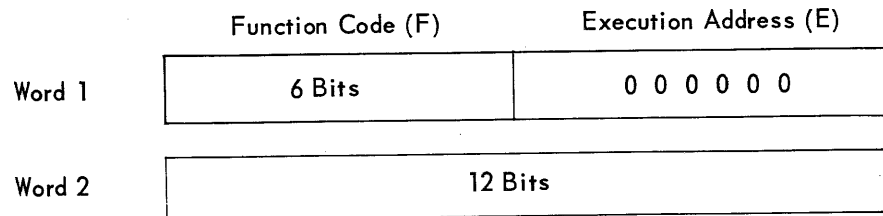
Two modes (C and M) use two sequential storage locations, the second of which is designated G. A prerequisite for these modes is that E must always be equal to zero.

Constant Address (C) The G portion of the 24-bit instruction word contains the operand.

Memory Address (M) The G portion of the 24-bit instruction word contains the address of the operand.

WORD FORMAT

A 160-A instruction word is divided into a 6-bit function code (F) and a 6-bit execution address (E). Most instructions require only one word of storage, but certain expanded instructions are available which occupy two words of storage. The format of two-word instructions is always as shown below; the first word contains a 6-bit function code and a 6-bit execution address. The execution address is always equal to zero. The second word contains a 12-bit address or operand (G), depending on the instruction. Words 1 and 2 must be located in sequential storage addresses of the same bank.



F, and, in some cases, E determines which instruction in the 160-A repertoire will be executed. Because E contains only 6 bits, it is not possible for E to specify a complete storage address in all cases. Therefore, depending on F, the computer selects, for each instruction, one of eight addressing modes.

NO ADDRESS MODE (N)

In the no address mode, E is the lower 6 bits of an implied 12-bit operand, the upper 6 bits of which are always zero. Thus, the E portion of the instruction word becomes the operand.

Example:	Location	F	E
	(r)0100	LDN	43
	(r)0101	next instruction	

At location (r)0100 is a load no address (LDN) instruction.

A load instruction transmits the operand to the A register. The 12-bit operand for LDN 43 is 0043.* Therefore the number 0043 will be transferred to the A register. At the completion of a no address (N) instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In this case, control will continue at location (r)0101.

DIRECT ADDRESS MODE (D)

In the direct address mode, E selects one of the first 64 (100 octal) locations in the direct storage bank (d) as the operand address.

Example:	Location	F	E
	(d)0076	12	34
	(r)1075	LDD	76
	(r)1076	next instruction	

At location (r)1075 is a load direct (LDD) instruction.

E specifies that the operand address is (d)0076. This address contains the quantity 1234 which will be transferred to the A register. At the completion of a direct address (D) instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In this case, control will continue at (r)1076.

INDIRECT ADDRESS MODE (I)

In the indirect address mode, E references 77 of the first 100 octal locations,

* All numbers are in octal unless stated otherwise.

starting at location 0001, the direct storage bank (d). The location (d)00E is then referenced and the contents of (d)00E become the operand address in the indirect bank (i).

Example:	Location	F	E
	(d)0045	33	65
	(i)3365	46	57
	(r)4121	LDI	45
	(r)4122	next instruction	

At location (r)4121 is a load indirect (LDI) instruction.

E calls for a reference to (d)0045, which contains the address 3365. A final reference is now made to (i)3365, which contains the number 4657. The quantity 4657 will be transferred to the A register. Notice that both the direct (d) and indirect (i) storage bank controls are involved in the indirect address (I) mode. At the completion of an (I) instruction, control always continues at the location in the relative storage bank specified by the contents of P + 1. In the above example, control will continue at (r)4122.

There are two (I) instructions which are exceptions to the above rules, JPI and JFI:

- 1) In JPI, the initial reference is made to (d)00E. A transfer of control then takes place within the relative (r) bank to the location specified by the contents of (d)00E.
- 2) In JFI, the initial reference is relative forward. A transfer of control then takes place within the relative (r) bank to the address specified in the relative forward reference.

RELATIVE FORWARD ADDRESS MODE (F)

In the relative forward address mode, E is added to the contents of the P register. This sum then becomes the effective operand address in the relative storage bank (r).

Example:	Location	F	E
	(r)0233	LDF	22
	(r)0234	next instruction	
	(r)0255	77	03

At location (r)0233 is a load forward (LDF) instruction.

E is added to the P register to yield the address (r)0255. The contents of (r)0255 will be transferred to the A register. At the completion of this instruction, A will contain the quantity 7703. At the completion of an (F) instruction which does not cause control to be transferred, control will continue in the relative storage

bank (r) at the location specified by the contents of P + 1. In the above example, control continues at location (r)0234. Forward Jump instructions cause control to be transferred E locations forward in the relative bank (see instruction repertoire).

RELATIVE BACKWARD ADDRESS MODE (B)

The relative backward address mode functions in a manner analogous to the relative forward (F) mode except that (E) is subtracted from the contents of the P register to form an effective address in the relative storage bank.

SPECIFIC ADDRESS MODE (S)

In the specific address mode, E is always equal to zero. The effective address for an (S) instruction is always storage bank zero, location 7777, (0)7777.

Example:	Location	F	E
	(r)0177	LDS	00
	(r)0200	next instruction	
	(0)7777	43	21

Location (r)0177 contains a load specific (LDS) instruction. The fact that E equals zero is used in address decoding to specify that the address of the operand of this instruction is (0)7777. Thus the quantity 4321 will be transferred to the A register. At the completion of an (S) instruction, control continues in the relative storage bank at the location specified by the contents of P + 1. In the above example, control continues at (r)0200.

CONSTANT ADDRESS MODE (C)

All constant address mode instructions occupy two sequential storage locations

wherein the G portion of the 24-bit instruction word contains the operand. E is always equal to zero.

Example:	Location	F	E	G
	(r)0101	LDC	00	
				7337
	(r)0103	STC	00	
				2345
	(r)0105	next instruction		

At location (r)0101 is a load constant (LDC) instruction. The operand address is (r)0102 and the quantity 7337 is transferred to the A register. At the completion of a (C) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P + 2. In this case, control continues at (r)0103. This address contains a store constant (STC) instruction. This causes the contents of the A register to be transferred to the operand address. In the above example, the operand address of the STC instruction is (r)0104. The quantity 7337, in the A register as a result of the LDC instruction in (r)0101, will be transferred to location (r)0104 and will replace the constant 2345 now in (r)0104. The final contents of (r)0104 will be 7337 and control will continue at (r)0105.

MEMORY ADDRESS MODE (M)

All memory address mode instructions occupy two sequential storage locations wherein the G portion of the 24-bit instruction word contains the address of the operand. E is always equal to zero.

Example:	Location	F	E	G
	(r)3477	LDM	00	
				1111
	(r)3501	STM	00	
				0024
	(r)3503	next instruction		
	(i)1111	67	66	
	(i)0024	02	34	

Location (r)3477 contains a load memory (LDM) instruction. The location (i)1111 becomes the operand address and the quantity 6766 is transferred to the A register. At the completion of an (M) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P + 2. In this case, control continues at location (r)3501 which contains a store memory (STM) instruction. The operand address of this instruction becomes (i)0024 and the quantity 6766 in the A register will be stored in location (i)0024, replacing the quantity 0234. Control will continue at location (r)3503.

STORAGE CONTROL

Magnetic core storage in the 160-A is composed of two, four, six or eight separate banks of cores. Each bank contains 4,096 12-bit words. A minimum memory contains 8,192 words and is expandable in modules of 8,192 words to a total of 32,768 words. Each bank is assigned a number, from 0 through 7, which is never changed. Associated with these banks is a set of four 3-bit registers called storage bank controls; each control may be set by a computer instruction to reference any one of the eight possible banks. Each bank has its own set of 10000₈ addresses numbered from 0000 to 7777. When a bank control is set so that it references a non-existent bank, the computer will stop with a fault indication if an instruction is executed which uses that particular control.

RELATIVE BANK CONTROL (r)

This control selects which storage bank will be referenced to obtain all instructions for execution. All instructions are executed from the bank to which the (r) control has been set. (r) also selects the bank which will be referenced by all instructions whose operation codes indicate relative or constant addressing, and as a first address for memory instructions (see section on addressing modes).

DIRECT BANK CONTROL (d)

This control selects the storage bank referenced by all instructions having direct addressing operation codes. For operation codes indicating indirect addressing, (d) selects the bank used for the first direct address selection.

INDIRECT BANK CONTROL (i)

For instructions with indirect or memory addressing operation codes, (i) selects which bank will be referenced to select the final operand. An exception is the jump instructions which cause an indirect transfer of control. JPI and JFI cause control to be transferred within (r). During normal input/output operations, (i) selects the bank to or from which information will be transferred.

BUFFER BANK CONTROL (b)

During operations which use the buffer input/output channel, (b) selects the storage bank to or from which information will be transferred.

Example: Assume (r) is set to reference bank 0
(i) is set to reference bank 1
(d) is set to reference bank 2
(b) is not used in this example

The following conditions exist within storage (the bank numbers in parentheses):

<u>Memory Location</u>	<u>Contents</u>
(1) 1577	7717
(1) 2345	5757
(2) 0005	1577
(2) 0067	2345
(2) 0070	temporary storage

Since (r) is set to 0, all program steps will be executed from bank 0.

<u>LOCATION</u>	<u>CONTENTS</u>		<u>COMMENTS</u>
	Symbolic F	Numeric E	
(0)0100	LDF 10	2210	Load contents of (r)0110 = (0)0110 into A. Upon completion A contains 3333.
(0)0101	STD 70	4070	Store contents of A in location (d)0070 = (2)0070. Upon completion A and (2)0070 both contain 3333.
(0)0102	LDI 05	2105	(d)0005 = (2)0005 contains 1577. (i)1577 = (1)1577 contains the operand 7717. 7717 is loaded into A. Upon completion A contains 7717.
(0)0103	ZJF 02	6002	If A contains all zeros, read next instruction from (r)0105 = (0)0105.
(0)0104	JPI 67	7067	Jump via (d)0067 = (2)0067. 2345, the contents of (2)0067 becomes (r)2345 or (0)2345, the address of next instruction.
(0)0105	STB 01	4301	Store current contents of A in (r)0104 = (0)0104.
(0)0106	JFI 01	7101	Jump to address specified by contents of (r)0107 = (0)0107. 2121, contents of (0)0107 become (r)2121, or (0)2121, the address of next instruction.
(0)0107		2121	Jump address.
(0)0110		3333	Working constant.

Chapter III

OPERATION

160-A COMPUTER CONSOLE

The control console consists of a display panel and a switch panel (figure 2). The display panel contains a 2-module status display and three 12-bit octal register displays. All information is displayed in octal, using arabic numerals. Displays are not illuminated when the computer is in the run mode. Below each 12-bit register display are 13 push buttons. The 12 blue buttons are used to enter information into the register, and the white button is used to clear the register. The only registers which can be so entered or cleared are the P, A, and Z registers.

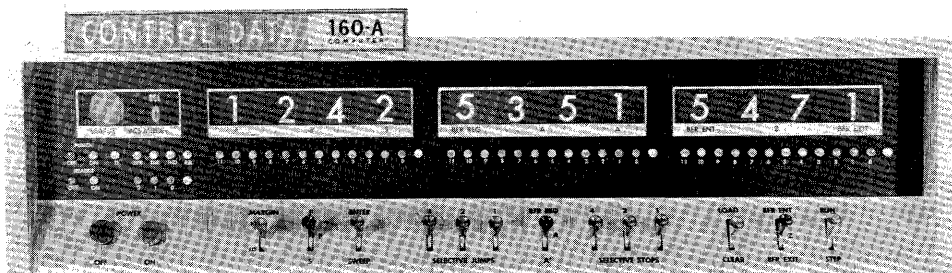


Figure 2. 160-A Console

REGISTER DISPLAY

The three 12-bit register displays can display the contents of nine 160-A registers. Centered beneath each register window is a three-position lever switch which determines the particular register to be displayed in that window.

P Register Group

	Switch Position	
F register	UP	Indicates the current effective function code The most significant digit shows a green background

P register	CENTER	Indicates the address of the current instruction
S register	DOWN	Indicates the address of the word about to be transferred to or from storage Least significant digit shows a green background

A red background indicates a non-existent storage bank reference, usually a programming error. This fault can be cleared by a console master clear.

A Register Group

	Switch Position	
BFR	UP	Indicates the last word processed during the last buffer operation The most significant digit shows a green background
A register	CENTER	Display indicates the current contents of the A register
A' register	DOWN	Display indicates the result of the last operation in the adder The least significant digit shows a green background

A red background indicates a main timing fault in the computer; normally occurs when the computer is first turned on and can be cleared by performing a console master clear. If this condition appears at any other time and cannot be cleared, call maintenance.

A blue background in the two center digit positions indicates an interrupt lockout.

Z Register Group

	Switch Position	
BER	UP	Indicates the address of the last word transferred on the buffer channel The most significant digit shows a green background
Z register	CENTER	Indicates the current contents of the Z register
BXR	DOWN	Indicates the LWA + 1 of the last buffer operation The least significant digit shows a green background

A blue background indicates the presence of an instruction in the Z register rather than a transient data word.

A red background in the two center digit positions indicates that the HSP is low on tape.

Status Mode Indicator

Green background Indicates the computer is in run status. This does not necessarily indicate that the instructions are being executed.

Red background Indicates that the computer is in either the stop or step status. Appears when:
a) a HLT or ERR instruction is executed
b) a SLS or SJS is executed and the computer stops
c) the computer is manually taken out of run status by lowering the Run/Step switch

Within the colored background, alphanumeric symbols are displayed which indicate computer conditions:

ERR The computer has executed an ERR instruction and is stopped.

SEL Displayed each time an EXF or EXC instruction is executed; will remain displayed until the selection is completed. A constant display of SEL with no apparent input/output action usually indicates that the computer has attempted an illegal selection.

OUT Displayed during all normal output operations. A constant display of OUT with no apparent output action usually indicates that output was attempted without proper unit selection.

IN Displayed during all normal input operations. A constant display of IN with no apparent input action usually indicates that input was attempted without proper unit selection. IN is also displayed when the computer is waiting for an external device to supply data, for instance, while data is being entered at the keyboard when typewriter input is requested.

IBA Displayed during all buffer input operations. See IN for additional comments.

OBA	Displayed during all buffer output operations. See OUT for additional comments.
A, B, C, or D	Indicates which storage reference cycle will be executed at the next operation of the Run/Step switch. Following a master clear, D is displayed. This indicates that the next operation executed (when the Run/Step switch is operated) will fetch the instruction from relative storage bank (r), at the address indicated by the P register, and insert it in Z.

MCS Mode Indicator

This display indicates the storage bank number to which each of the four 160-A storage bank controls has been set. The display is divided into upper and lower sections. The upper (alphabetic) indicates which storage bank control is displayed in the lower (numeric). The upper display indicates the following:

REL	Relative storage bank control (r)
IND	Indirect storage bank control (i)
DIR	Direct storage bank control (d)
BFR	Buffer storage bank control (b)

The lower display will contain one of the digits from 0 through 7.

Normally the storage bank to be used with the next storage reference is displayed in the MCS mode indicator. However, any of the other controls may be temporarily displayed by pressing one of the four buttons directly under the MCS mode indicator: BFR, DIR, IND, REL. A second, lower row of four buttons is used to set any of the storage bank controls from the console as follows:

The rightmost button (white) is a clear button which sets the bank number to zero; the other three buttons control the binary bit value of the bank number.

- 1) Press the button on the upper row corresponding to the bank control to be set. Hold this button down while performing the remaining steps.
- 2) Press the white clear button on the lower row and then release it. This will set the bank number to zero.
- 3) Enter the bank number into bank control by pressing the correct lower buttons. For instance, buttons 1 and 2 set the bank number to 6.
- 4) Release all buttons.

The relative storage bank control is set to zero when a console master clear is performed, but the direct, indirect, and buffer storage bank controls are unaltered.

The Switch Panel

Power On/Off	Blue button turns on power to the 160-A Red button turns off power to the 160-A
Margin	Maintenance control for varying the bias on magnetic core storage sense amplifiers This switch should not be used by the operator
F, P, S	3-position switch that chooses the register to be displayed in the P register group
Enter/Sweep	ENTER is used for entering information into the core storage of the 160-A from the console SWEEP is used to display the contents of core storage
Selective Jump Switches (4, 2, 1)	UP selects jump conditions for interrogation by SLJ and SJS instructions
BFR, A, A'	3-position switch that chooses the register to be displayed in the A register group
Selective Stop Switches (4, 2, 1)	UP selects stop conditions for interrogation by SLS and SJS instructions
Load/Clear	Momentary CLEAR performs a computer master clear which: a) clears the P, A, Z, F and F' registers b) clears the control flip-flops c) sets (r) = 0 d) clears all waiting interrupts and removes interrupt lockout This master clear does not alter core storage LOAD position enables specially prepared paper tapes to be read into storage by the 350 reader
BFR ENT, Z, BFR EXIT	3-position switch that chooses the register to be displayed in the Z register group

Run/Step

In RUN position, a program is executed at high speed starting at the location specified by the P register.

The center (neutral) position stops the computer. If the switch is in RUN and an ERR, SLS, SJS, or HLT instruction is executed, the switch must be returned to neutral and then placed in RUN to continue computation.

In STEP position, one storage cycle of an instruction is executed each time the switch is set. In this manner, a program may be executed one instruction at a time for debugging.

By simultaneously holding down the step switch and any one of the selective jump switches the computer is placed in the automatic step mode. In this mode, a program is executed at a rate of about 3 instructions per second with console display.

STARTING THE 160-A

- 1) Be sure the computer is plugged into the proper power source and that the room temperature is within the prescribed limits.
- 2) Press the Power On button on the console.
- 3) The P and A registers will normally appear against a red background. Master clear by momentarily pressing the Load/Clear switch to the CLEAR position.
- 4) The red backgrounds should go out. This indicates that the computer is ready to operate. If repeated master clears do not remove the red background from P and A, turn the 160-A off by pressing the Power Off button, and call maintenance.

LOADING A PROGRAM OR DATA IN PAPER TAPE LOAD FORMAT

- 1) Master clear
- 2) Turn on 350 reader by pressing the Reader On button, left side of the console.
- 3) Insert paper tape in reader (see page 3-8).
- 4) Set P to starting location.

- 5) Set relative storage bank control to select the bank into which the tape will be read.
- 6) Set the Load/Clear switch to LOAD.
- 7) Set the Run/Step switch to RUN. The paper tape will load and the computer will stop.

ENTERING DATA FROM THE CONSOLE

- 1) Master clear. Set the Enter/Sweep switch on ENTER.
- 2) Set relative storage bank control to select the bank into which data is to be entered.
- 3) Set P to the location into which data is to be entered.
- 4) Enter one word of data into the Z register.
- 5) Press the Run/Step switch to STEP, once. At this point Z is clear, the data word is in storage and in A, and P has been advanced by 1. (P advances each time except the first time.)
- 6) If data is to be entered into consecutive locations, go to step 4. If data is to be entered into non-consecutive locations, clear P. Go to step 3.

EXAMINING THE CONTENTS OF STORAGE AT THE CONSOLE

- 1) Master clear. Set the Enter/Sweep switch on SWEEP.
- 2) Set relative storage bank control to select the bank to be examined.
- 3) Set P to the location to be examined.
- 4) Press the Run/Step switch to STEP, once. The contents of the location specified by P will appear in Z, and P will be advanced by 1. (P advances each time except the first time.)
- 5) To examine consecutive locations, go to step 4. To examine non-consecutive locations, clear P. Go to step 3.

CLEARING AN ENTIRE STORAGE BANK

- 1) Master clear. Set the Enter/Sweep switch on SWEEP.
- 2) Set the relative storage bank control to select the bank to be cleared.

- 3) Set the Run/Step switch on RUN.
- 4) Press the Z-clear button and hold for about a second.

OPERATING THE 350 PAPER TAPE READER

The 350 reader is turned on and off from the I60-A console.

TAPE LOADING

- 1) Turn reader on.
- 2) Rotate the tape release handle clockwise to separate the idler rollers and raise the tape guide plate.
- 3) Set the tape width guide by pressing down the guide to release the locking fingers and slide it so that the marker rests above the correct etched mark on the tape deck surface.
- 4) Set the tape level switch for 5, 7, or 8 channel reading.
- 5) Insert the tape in the reader between the idler rollers (figure 3). The highest level hole is toward the open side. Be sure the tape is seated correctly in the tape guide.
- 6) Engage the tape release handle by turning it counterclockwise.

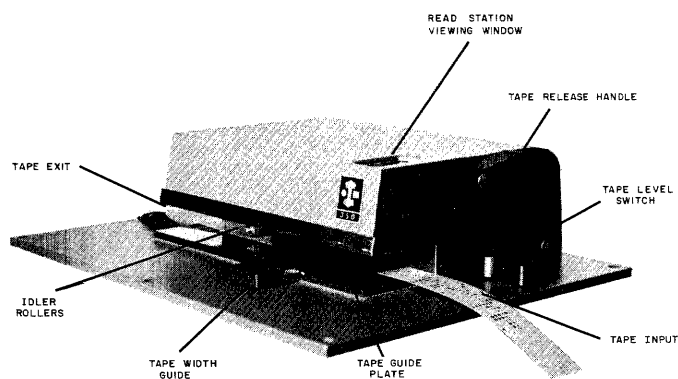
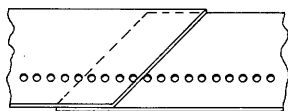


Figure 3. Paper Tape Reader

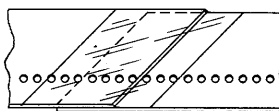
SPLICING PAPER TAPE

Improperly spliced tape can cause tape snagging or twisting. Avoid any splice

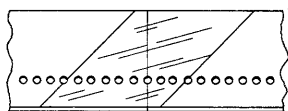
with a loose edge capable of rising in the direction of tape travel. Align feed holes on the strips before joining. Acceptable splices are shown below.



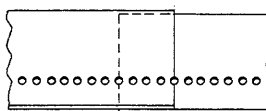
ACCEPTABLE:
DIAGONAL LAP - CEMENTED
(AVOID LOOSE EDGES)



ACCEPTABLE:
DIAGONAL LAP - WITH TRANSPARENT
SPLICING TAPE
(TOTAL ALLOWABLE THICKNESS .012")



ACCEPTABLE:
SQUARE (OR DIAGONAL) BUTT SPLICE
WITH DIAGONAL SP LICING TAPE



NOT ACCEPTABLE:
SQUARE LAP SPLICE

Figure 4. Tape Splices

OPERATING THE PAPER TAPE PUNCH

The 110 character per second punch is turned on and off from the 160-A console. To the right of the punch On/Off buttons is a button labeled TL (tape load). When the punch is on, press and hold down the TL button to feed blank tape (with sprocket hole only) through the punch.

The punch may be operated with the unit remaining in the left pedestal of the computer as the tape feeds out of a special slot in the left side of the computer desk. However, the left pedestal door may be opened and the punch extended.

LOADING A NEW ROLL OF TAPE

- 1) Turn punch on.
- 2) Tear off old tape and run all tape out of the punch block by pressing the TL button.
- 3) Turn off punch.
- 4) Lift the tape reel out of the punch and remove the old reel by unscrewing the X-shaped side plate.
- 5) Place a new roll of tape on the reel so that the tape unwinds counterclockwise, and attach the side plate.

- 6) Place the tape reel in the punch. Thread the tape through the loop on the reel brake arm, through the loop at the front of the punch, around the two rollers and into the tape guide.
- 7) Slowly slide the tape through the punch block. Pull the tape through the punch block, lift the tape tension lever and feed the tape between the tape tension lever and the tape feed wheel.
- 8) Turn punch on.
- 9) Gently pull on the tape and press the tape feed lever. The tape will begin to feed automatically into the punch block. After about 6 inches of tape have been fed, the sprocket holes will be correctly punched.

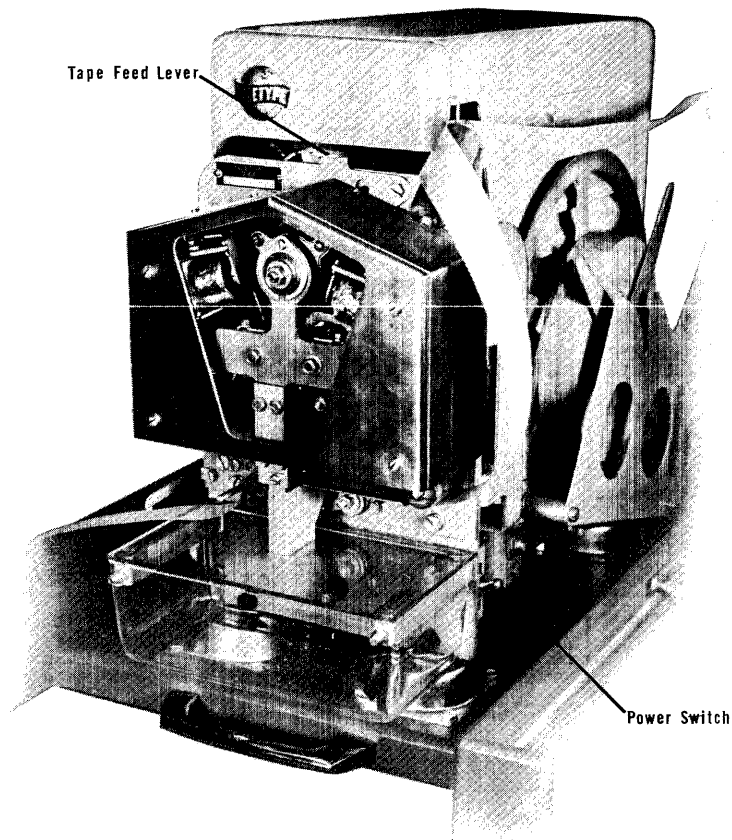


Figure 5. Paper Tape Punch

160-A INSTRUCTIONS

- 1) All instruction times are in storage cycles where 1 cycle equals 6.4 usec.
- 2) All numbers are in octal notation unless otherwise stated.
- 3) In the description of an operation code, the operations involved are listed in sequence if more than one operation is performed.
- 4) Instructions are described by general function rather than in numerical sequence. Instructions which may assume more than one address mode are listed under the general function. The operand formation for each addressing mode is described in the section on addressing modes.
- 5) The G portion of an instruction is shown with only those instructions which occupy two words of storage.

STOP INSTRUCTIONS

This group of instructions stops the computer unconditionally.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
00	00	ERR	Error Stop	1
77	00	HLT	Halt	1
77	77	HLT	Halt	1

Description:

Stop computation. When switch is in RUN position on the console, control continues at $(r) (P) + 1$. After the computer stops as the result of an ERR instruction, the letters ERR will be displayed in the status mode indicator. There is no difference in the action of the two HLT instructions, 7700 and 7777.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
00	0X	NOP	No Operation	1

Description:

When a NOP instruction is executed the computer does not perform any function but passes on to the next instruction at location $(r) (P) + 1$.

DATA TRANSMISSION INSTRUCTIONS

This group of instructions transfers information to and from the accumulator with-in or between storage banks, and also between the internal registers.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	00	YYYY	BLS	Block Store	(no jump) 1 (jump) 2

Description: If the buffer is busy, YYYY \rightarrow P
 If the buffer is not busy, (A) \rightarrow BFR
 (P) + 2 \rightarrow P
 Start the cycle: * (BFR) \rightarrow (b) (BER)
 (BER) + 1 \rightarrow BER

 If (BER) \neq (BXR), go to *
 If (BER) = (BXR), terminate buffer operation

BLS is used to set selected positions of storage to any desired value. The FWA of the area to be set is placed in BER, the LWA + 1 of the area to be set is placed in BXR. (b) is then set to reference the storage bank that is to be set. The value to which the area is to be set is placed in A and then BLS is given. If, at the time a BLS is given, the buffer is in operation, the program jumps to (r)YYYY. If buffer is not in operation when BLS is given, the store operation takes place and control continues at (r) (P) + 2.

The buffer storage bank control (b) may be set at any time prior to executing BLS. Note that, although BLS is a buffer operation, because of the need to constantly refer to storage for the store operation the next instruction will not be executed until BLS is completed. The total execution time for the BLS operation is 1 + N cycles, where N is the number of locations to be set + 1.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	01	PTA	Transfer P to A	1

Description: (P) \rightarrow A
 (P) + 1 \rightarrow P

Transfer the contents of P to A. Control continues at (r)(P) + 1. At the time PTA is executed, P will contain the address of the PTA instruction.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	05	YYYY	ATE	A to Buffer Entrance Register	(no jump) 1 (jump) 2

Description: If the buffer is busy, YYYY \rightarrow P
 If the buffer is not busy, (A) \rightarrow BER
 (P) + 2 \rightarrow P

Transfer the contents of A to BER. Control continues at (r)(P) + 2. If the buffer

is in operation when an ATE is given, A is not transferred and control continues at (r)YYYY.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	06	YYYY	ATX	A to Buffer Exit Register	(no jump) 1 (jump) 2

Description: If the buffer is busy, YYYY → P
 If the buffer is not busy, (A) → BXR
 (P) + 2 → P

Transfer the contents of A to BXR. Control continues at (r)(P) + 2. If the buffer is in operation when an ATX is given, A is not transferred and control continues at (r)YYYY.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	07	ETA	Buffer Entrance Register to A	1

Description: (BER) → A
 (P) + 1 → P

Transfer the contents of BER to A. This instruction may be given at any time, even while the buffer is in operation. Control continues at (r)(P) + 1.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	30	CTA	Bank Controls to A	1

Description: (b) → A₁₁₋₉*
 (d) → A₈₋₆
 (i) → A₅₋₃
 (r) → A₂₋₀
 (P) + 1 → P

Transfer the contents of the four storage bank controls to A as octal digits which occupy the A register bit positions shown above. Control continues at (r)(P) + 1. Note that the storage bank controls are not changed by the execution of a CTA instruction.

Example: Assume (b) is set to reference bank 4
 (d) is set to reference bank 0
 (i) is set to reference bank 1
 (r) is set to reference bank 2

Under the above conditions, if a CTA were given, A would contain the number 4012 at the completion of the instruction.

*Subscripts indicate bit positions in A.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	5Y	STP	Store P at Location 5Y	3

Description: (P) → (d) 005Y
(P) + 1 → P

Transfer the contents of P to storage location (d) 005Y, where Y is any octal digit from 0 through 7. Control continues at location (r)P + 1. This instruction allows the contents of P, which will contain the address of STP, to be stored in the direct storage bank at any of the locations 0050 - 0057.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
04	XX		LDN	Load No Address	1
20	YY		LDD	Load Direct	2
21	00	YYYY	LDM	Load Memory	3
21	YY		LDI	Load Indirect	3
22	00	XXXX	LDC	Load Constant	2
22	XX		LDF	Load Forward	2
23	00		LDS	Load Specific	2
23	XX		LDB	Load Backward	2

Description: operand → A

Transfer the operand to A. The operand in storage is not altered by the execution of a load instruction. The proper operand is formed for each address mode and control continues as described in the section on address modes.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	6Y	STE	Store Buffer Entrance Register at Location 6Y and Transfer A to Buffer Entrance Register	3

Description: (BER) → (d) 006Y
(A) → BER
(P) + 1 → P

Transfer the contents of BER to storage location (d) 006Y, where Y is any octal digit from 0 through 7, and transfer the contents of A to BER. A is unchanged by this instruction. Control continues at location (r)(P) + 1. This instruction allows the contents of BER to be exchanged whenever desired, even while the buffer is in operation. The old contents of BER will be stored in the direct storage bank (d) at any of the locations 0060 - 0067.

Note that the E portion of LDI, LDF, and LDB instructions cannot be zero or the operation code is interpreted as LDM, LDC, and LDS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
05	XX		LCN	Load Complement No Address	1
24	YY		LCD	Load Complement Direct	2
25	00	YYYY	LCM	Load Complement Memory	3
25	YY		LCI	Load Complement Indirect	3
26	00	XXXX	LCC	Load Complement Constant	2
26	XX		LCF	Load Complement Forward	2
27	00		LCS	Load Complement Specific	2
27	XX		LCB	Load Complement Backward	2

Description: operand → A

Transfer the one's complement of the operand to A. The operand in storage is not altered by the execution of a load complement instruction. The one's complement of a number is the arithmetic negative of that number. The proper operand is formed for each address mode and control continues as described in the section on address modes.

Note that the E portion of LCI, LCF, and LCB instructions cannot be zero or the operation code is interpreted as LCM, LCC, and LCS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
40	YY		STD	Store Direct	3
41	00	YYYY	STM	Store Memory	4
41	YY		STI	Store Indirect	4
42	00	XXXX	STC	Store Constant	3
42	XX		STF	Store Forward	3
43	00		STS	Store Specific	3
43	XX		STB	Store Backward	3

Description: (A) → operand address

Transfer the contents of A to the operand address. The contents of A are not altered by a store instruction. The proper operand address is formed for each address mode and control continues as described in the section on address modes.

Note that the E portion of STI, STF, and STB instructions cannot be zero or the operation code is interpreted as SFM, STC, and STS, respectively.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
76	YY	HWI	Half Write Indirect	4

Description: (A) _L → operand address _L
 (P) + 1 → P

Transfer the lower order 6 bits of A into the lower order 6 bits at the operand address. A and the higher order 6 bits at the operand address remain unchanged. The proper operand address is formed as described in the section on address modes. Control will continue at (r)(P) + 1.

Example:	Location	F	E
	(r) 3777	HWI	23
	(d) 0023	41	14
	(i) 4114	75	01
	(A)	46	57

When the HWI instruction at location (r)3777 is executed, the lower 6 bits of A will be transferred to the lower 6 bits of location (i)4114. At the completion of the instruction, A will contain the quantity 4657, location (i)4114 will contain the quantity 7557, location (d)0023 will be unchanged, and control will continue at location (r)4000.

If the E portion of the HWI instruction is equal to 00 or 77, the operation code will be interpreted as INA or OTA, respectively. The effective indirect address reference ranges from (d)0001 to (d)0076.

ARITHMETIC INSTRUCTIONS

This group of instructions performs various arithmetic operations. The accumulator is always used when initiating these instructions; however, only a replace type instruction affects storage. There are no single instructions to multiply or divide, but there are special instructions that multiply by 10^1 or 10^2 .

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	12	MUT	Multiply A by 10	1

Description: $10_{10}(A) \rightarrow A$,
 (P) + 1 → P

Multiply the contents of A by 10_{10} . The result is placed in A at the completion of the instruction. For the range of numbers -314_8 to $+314_8$, the result will be algebraically correct. If $(A) > +314_8$ or $< -314_8$, the result will be correct modulo $2^{12}-1$. Control will continue at location (r)(P) + 1.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	13	MUH	Multiply A by 100	1

Description: $100_{10}(A) \rightarrow A$
 $(P) + 1 \rightarrow P$

Multiply the contents of A by 100_{10} . The result is placed in A at the completion of the instruction. For the range of numbers -24_8 to $+24_8$, the result will be algebraically correct. If $(A) > +24_8$ or $< -24_8$, the result will be correct modulo $2^{12}-1$. Control will continue at location $(r)(P) + 1$.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
06	XX		ADN	Add No Address	1
30	YY		ADD	Add Direct	2
31	00	YYYY	ADM	Add Memory	3
31	YY		ADI	Add Indirect	3
32	00	XXXX	ADC	Add Constant	2
32	XX		ADF	Add Forward	2
33	00		ADS	Add Specific	2
33	XX		ADB	Add Backward	2

Description: $(A) + \text{operand} \rightarrow A$

Place in A the sum of the original contents of A and the operand. The operand is unchanged by an add instruction. The correct operand is formed for each address mode and control will continue as described in the section on address modes.

Note that the E portion of ADI, ADF, and ADB instructions cannot be zero or the operation code is interpreted as ADM, ADC, and ADS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
07	XX		SBN	Subtract No Address	1
34	YY		SBD	Subtract Direct	2
35	00	YYYY	SBM	Subtract Memory	3
35	YY		SBI	Subtract Indirect	3
36	00	XXXX	SBC	Subtract Constant	2
36	XX		SBF	Subtract Forward	2
37	00		SBS	Subtract Specific	2
37	XX		SBB	Subtract Backward	2

Description: $(A) - \text{operand} \rightarrow A$

Place in A the difference between the original contents of A and the operand. The operand is unchanged by a subtract instruction. The correct operand is formed for each address mode and control continues as described in the section on address modes.

Note that the E portion of SBI, SBF, and SBB instructions cannot be zero or the operation code is interpreted as SBM, SBC, and SBS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
50	YY		RAD	Replace Add Direct	3
51	00	YYYY	RAM	Replace Add Memory	4
51	YY		RAI	Replace Add Indirect	4
52	00	XXXX	RAC	Replace Add Constant	3
52	XX		RAF	Replace Add Forward	3
53	00		RAS	Replace Add Specific	3
53	XX		RAB	Replace Add Backward	3

Description: (A) + operand → A
(A) → operand address

Place in A the sum of the original contents of A and the operand, and transfer this sum to the operand address. At the completion of the replace add instruction, both the operand address and A will contain the new sum. The proper operand is formed for each address mode and control continues as described in the section on address modes.

Note that the E portion of RAI, RAF, and RAB instructions cannot be zero or the operation code is interpreted as RAM, RAC, and RAS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
54	YY		AOD	Replace Add One Direct	3
55	00	YYYY	AOM	Replace Add One Memory	4
55	YY		AOI	Replace Add One Indirect	4
56	00	XXXX	AOC	Replace Add One Constant	3
56	XX		AOF	Replace Add One Forward	3
57	00		AOS	Replace Add One Specific	3
57	XX		AOB	Replace Add One Backward	3

Description: operand → A
(A) + 1 → A
(A) → operand address

Form in A the sum of the operand plus one and transfer this sum to the operand address. At the completion of the replace add one instruction, both the A register and the operand address will contain the original operand increased by 1. The proper operand is formed for each address mode and control continues as described in the section on address modes.

Note that the E portion of AOI, AOF, and AOB instructions cannot be zero or the operation code is interpreted as AOM, AOC, and AOS, respectively.

SHIFT INSTRUCTIONS

This group of instructions moves information within the accumulator. When the left shift is initiated, the information moves in a circular path. When the right shift is initiated, the information drops off on the open right end of the accumulator.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	02	LS1	Left Shift One	1
01	03	LS2	Left Shift Two	1
01	10	LS3	Left Shift Three	1
01	11	LS6	Left Shift Six	1
01	14	RS1	Right Shift One	1
01	15	RS2	Right Shift Two	1

Description:

Shift A right or left the number of bit positions specified. Control continues at (r)(P) + 1.

All left shifts in the 160-A are circular; bits shifted out of bit position 11 are shifted into bit position 00. From bit position 00, bits are shifted into bit position 01, etc.

Example:	Location	F	E
	(r)6173	LS3	
	(A)	61	02

At location (r)6173 is an LS3 instruction. E is not specified since all of the above shift instructions use E as part of the operation code. A contains the number 6102. At the completion of LS3, A will contain the number 1026 and control will continue at (r)6174.

All right shifts in the 160-A are end-off shifts; for each bit position shifted, the sign is extended and the bit in position 00 is shifted off and lost.

Example:	Location	F	E
	(r)2234	RS1	
	(A)	40	01

At location (r)2234 is an RS1 instruction. Again, E is not specified since it is part of the operation code. A contains the number 4001. At the completion of RS1, A will contain the number 6000 and control will continue at location (r)2235.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
44	YY		SRD	Shift Replace Direct	3
45	00	YYYY	SRM	Shift Replace Memory	4
45	YY		SRI	Shift Replace Indirect	4
46	00	XXXX	SRC	Shift Replace Constant	3
46	XX		SRF	Shift Replace Forward	3
47	00		SRS	Shift Replace Specific	3
47	XX		SRB	Shift Replace Backward	3

Description: operand → A
 shift A left circular 1 bit position
 (A) → operand address

The operand is placed in A, shifted left one bit position (left shift), and the contents of A are transferred back to the operand address. The operand for each address mode is formed and control will continue as described in the section on address modes. At the completion of a shift replace instruction, both A and the operand address will contain the shifted original operand.

Note that the E portion of SRI, SRF, and SRB instructions cannot be zero or the operation code is interpreted as SRM, SRC, and SRS, respectively.

LOGICAL INSTRUCTIONS

This group of instructions complements or extracts any portion of the accumulator according to a prescribed bit pattern.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
02	XX		LPN	Logical Product No Address	1
10	YY		LPD	Logical Product Direct	2
11	00	YYYY	LPM	Logical Product Memory	3
11	YY		LPI	Logical Product Indirect	3
12	00	XXXX	LPC	Logical Product Constant	2
12	XX		LPF	Logical Product Forward	2
13	00		LPS	Logical Product Specific	2
13	XX		LPB	Logical Product Backward	2

Description:

Form in A the logical product of the operand and the original contents of A. The operand in storage is not altered by a logical product instruction. The proper operand is formed for each address mode and control will continue as described in the section on address modes.

The logical product of two operands is defined as follows:

operand 1 (bit value)	0 0 1 1
operand 2 (bit value)	<u>0 1 0 1</u>
logical product of 1 and 2	0 0 0 1 (bit value)

From the above definition it will be seen that, using the proper operand as a mask, selected portions of A may be cleared or retained in A.

Note that the E portion of LPI, LPF, and LPB instructions cannot be zero or the operation code is interpreted as LPM, LPC, and LPS, respectively.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
03	XX		SCN	Selective Complement No Address	1
14	YY		SCD	Selective Complement Direct	2
15	00	YYYY	SCM	Selective Complement Memory	3
15	YY		SCI	Selective Complement Indirect	3
16	00	XXXX	SCC	Selective Complement Constant	2
16	XX		SCF	Selective Complement Forward	2
17	00		SCS	Selective Complement Specific	2
17	XX		SCB	Selective Complement Backward	2

Description:

Form in A the bit complement of A for each bit in the operand equal to 1. The operand in storage is not altered by the selective complement instruction. The proper operand for each address mode is formed and control will continue as described in the section on address modes.

The selective complement operation is defined as follows:

(A) register (bit value)	0 0 1 1
operand (bit value)	0 1 0 1
final contents of the A register	0 1 1 0 (bit value)

Note that the E portion of SCI, SCF, and SCB instructions cannot be zero or the operation code is interpreted as SCM, SCC, and SCS, respectively.

STORAGE BANK CONTROL INSTRUCTIONS

This group of instructions assigns address modes to the various storage banks before transferring program control to a given bank reference. When the relative mode is assigned to a storage bank, separately or collectively, program control

is relinquished to the relative bank address specified by the contents of the accumulator.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
00	1X	SRJ	Set Relative Bank Control and Jump	1
00	2X	SIC	Set Indirect Bank Control	1
00	3X	IRJ	Set Indirect and Relative Bank Control and Jump	1
00	4X	SDC	Set Direct Bank Control	1
00	5X	DRJ	Set Direct and Relative Bank Control and Jump	1
00	6X	SID	Set Indirect and Direct Bank Control	1
00	7X	ACJ	Set Direct, Indirect, and Rela- tive Bank Control and Jump	1
01	4X	SBU	Set Buffer Bank Control	1

Description:

Set the specified storage bank control or controls to reference bank X. For the instructions SIC, SDC, SID, and SBU, control will continue at $(r)(P) + 1$. The remaining instructions of this group, SRJ, IRJ, DRJ, and ACJ, are the only 160-A instructions which can be used to transfer program control between storage banks. It is the act of setting the relative bank control (r) which alters the bank from which the next program instruction will be taken. Whenever an instruction is given which sets (r), the next program instruction will be taken from the new (r) at the address specified by the contents of the A register. Notice that not only may (r) be set by itself but combinations of memory bank controls may be set at the same time.

Example: Set (d) to reference storage bank 3. At the same time transfer program control to storage bank 3, location 2217.

Location	F	E	G
(r) a	LDC	00	2217
(r) a + 2	DRJ	53	

In the above example, the program to transfer control was not located in any specific place in the (r) bank but rather the letter "a" was used to denote that, wherever the program was placed, the next instruction would be taken from "a + 2". This two-instruction sequence is all that is required to transfer program control and set (d) as specified. Executing the LDC instruction transfers the number 2217 to A. This number represents the jump address in the new relative (r) bank. The next instruction, DRJ, sets (d) to reference bank 3, sets (r) to reference bank 3, and then jumps to location 2217 (the contents of A) in bank 3.

The contents of A are not changed when the storage bank control instructions are executed.

JUMP INSTRUCTIONS

This group of instructions changes its sequential references and resumes control at another address. Certain jump instructions unconditionally change the sequence. Other jump instructions are conditional depending on the content of the accumulator or the position of the jump switches on the console panel.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
60	XX	ZJF	Zero Jump Forward	1
61	XX	NZF	Non-Zero Jump Forward	1
62	XX	PJF	Positive Jump Forward	1
63	XX	NJF	Negative Jump Forward	1
64	XX	ZJB	Zero Jump Backward	1
65	XX	NZB	Non-Zero Jump Backward	1
66	XX	PJB	Positive Jump Backward	1
67	XX	NJB	Negative Jump Backward	1

Description:

The above are conditional jump instructions; A is tested for the condition stated. If the condition is met, control is transferred forward or backward in the relative (r) bank XX locations as described in the section on address modes. If the condition is not met, control continues at location (r)(P) + 1.

The conditions for testing are:

- 1) A zero: the contents of A are equal to 0000, or plus zero. Minus zero is not considered equivalent to plus zero to meet the zero jump condition.
- 2) A not zero: A contains any quantity other than 0000.
- 3) A positive: bit 11 of A is 0.
- 4) A negative: bit 11 of A is 1.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
70	YY	JPI	Jump Indirect	2

Description: ((d) 00YY) → P

Transfer program control to the location in the relative (r) bank specified by the contents of (d)00YY.

Example:	Location	F	E
	(r) a	JPI	43
	(d) 0043	36	62

At some location in the (r) bank is the instruction JPI 43. When this instruction is executed, since (d)0043 contains the number 3662, program control will be transferred to location (r)3662.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
71	00	YYYY	JPR	Return Jump	3

Description: (P) + 2 → (r)YYYY
 YYYY + 1 → P

The contents of P are increased by 2 to give the address of the instruction following the JPR instruction. This address is transferred to location (r)YYYY. Program control is then transferred to location (r)YYYY + 1.

Example:	Location	F	E	G
	(r)1173	JPR	00	2100

At location (r)1173 is a JPR instruction. When this instruction is executed, the number 1175 will be transferred to location (r)2100. Program control will then continue at location (r)2101.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
71	XX	JFI	Jump Forward Indirect	2

Description: (r)((P) + 00XX) → P

Transfer program control to the address in (r) specified by the contents of the storage location XX positions forward of the JFI instruction.

Example:	Location	F	E
	(r)5162	JFI	07
	(r)5171	04	23

When the JFI instruction at (r)5162 is executed, a reference is made to the location in the relative bank 0007 positions forward, location (r)5171. This location contains the number 0423 and program control will be transferred to (r)0423.

Note that the E portion of the JFI instruction cannot be equal to zero or the operation code will be interpreted as a JPR instruction.

INPUT/OUTPUT INSTRUCTIONS

This group of instructions permits communication and transfer of information between the computer and the external equipment. The path of in/out transmission can be on the buffer channel or the normal channel.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	04	CBC	Clear Buffer Controls	1

Description:

Stop all buffer operations in progress and clear the buffer controls. This instruction does not clear BER, BXR, or BFR, but it stops any buffer operation in progress. If an input/output operation is stopped when some piece of unit record equipment (such as magnetic tape, punched cards, etc.) is being read or written, the buffer will disconnect from the unit but the peripheral unit will move to the end of the unit record. The remaining data will not be stored. On non-unit record peripheral equipment, such as a paper tape input device, the paper tape* will stop in position to read the next frame of tape and the buffer controls will be cleared.

No buffer complete (interrupt 20) is generated when CBC is executed.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
01	20	CIL	Clear Interrupt Lockout	1

Description:

Clear the interrupt lockout and allow any waiting interrupts to function. When CIL is executed, interrupt lockout is not cleared until the instruction following CIL has been executed. Interrupt lockout is set by any of the following:

- 1) Execution of an interrupt.
- 2) Execution of an EXF instruction.
- 3) Execution of an EXC instruction. (See section on interrupt.)

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
72	00	YYYY	IBI	Initiate Buffer Input	(no jump) 1 (jump) 2

*Paper tape is used as an example only, since the console paper tape reader cannot be read from the buffer input/output channel.

Description: If buffer is busy, YYYYY → P
 If buffer is not busy, (P) + 2 → P
 Start input buffering operation and, when complete,
 generate an interrupt 20 signal.

The instruction IBI initiates an input buffer operation on the buffer input/output channel. Prior to an IBI, however, the external device must be selected and BER, BXR, and the buffer storage bank control (b) must be set up. If the buffer is in operation when an IBI is given, no buffer action will be taken and control will continue at location (r)YYYY. If the buffer is not in operation when IBI is given, the buffering operation will be started and control will immediately continue at location (r)(P) + 2. When the buffer operation terminates, an interrupt signal will appear on interrupt line 20.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
73	00	YYYY	IBO	Initiate Buffer Output	(no jump) 1 (jump) 2

Description: If buffer is busy, YYYYY → P
 If buffer is not busy, (P) + 2 → P
 Start output buffering operation and, when complete,
 generate an interrupt 20 signal.

The IBO instruction initiates an output buffer operation on the buffer input/output channel. Prior to an IBO the external device must be selected, and BER, BXR, and the buffer storage bank control (b) must be set up. If the buffer is in operation when IBO is given, no buffer action will be taken and control will continue at location (r)YYYY. If the buffer is not in operation when IBO is given, the buffering operation will be started and control will immediately continue at location (r)(P) + 2. When the buffer operation terminates, an interrupt signal will appear on interrupt line 20.

If either IBI or IBO is given and no external device has been selected, the computer will continue in operation but the buffer will be placed in an indefinite busy status and no input or output operation will take place. This busy status may be removed by a clear buffer controls (CBC) instruction or by a master clear from the console.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
72	XX	YYYY	INP	Normal Input	*
73	XX	YYYY	OUT	Normal Output	*

Description: (r)(P) + 00XX = the FWA of the input or output area
 YYYYY = the LWA + 1 of the input or output area

*Since INP and OUT are not buffered instructions, their execution time varies with the speed of the external equipment being used.

Read or write using the previously selected external device. The input/output area is defined as follows:

- 1) The FWA of the input/output area is found XX locations forward in the relative memory bank (r). This location (r) P + 00XX, specifies a FWA in the indirect memory bank (i).
- 2) The LWA + 1 of the input/output area is location YYYYY in the indirect memory bank (i)YYYYY.

If the external device has been properly selected, the input or output operation will take place and, at its completion, control will continue at (r)(P) + 2. If no external device has been properly selected, the computer will be indefinitely delayed and the operation mode (INP, OUT) displayed on the status mode indicator.

The instructions INP and OUT are used to read or write data on the normal input/output channel and are not buffered; that is, the computer will wait while the input or output operation is in progress and the next instruction will not be executed until the input/output operation is complete. The contents of A at the completion of an INP or OUT instruction will indicate the LWA + 1 actually read into or out of during the input or output operation. Although the FWA and LWA of the input/output area are found in the relative memory bank (r) they actually specify locations in the indirect memory bank (i).

Example:	Location	F	E	G
	(r)1134	INP	33	2000
	(r)1136			
	(r)1167	10	00	

At location (r)1134 is a normal input instruction. The FWA of the input area is found in location (r)1167 and is (i)1000. The LWA + 1 of the input area is found in G and is (i)2000. The LWA itself is therefore (i)1777. Assuming an external device has been selected, INP is interpreted as "Read from selected device 1000 words and store them in the indirect storage bank starting at location 1000."

If 1000 words are read, A will contain the number 2000 at the completion of INP and control will continue at location (r)1136.

The E portion of INP and OUT instructions cannot be equal to zero, or the operation codes are interpreted as IBI and IBO, respectively.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
74	XX	OTN	Output No Address	*

*Execution time varies with the speed of the external equipment being used.

Description:

Write, on the previously selected output device, one word with zero in the 6 high order bits and XX in the 6 low order bits. The output operation takes place under normal input/output using either the normal or buffer channel. At the completion of the output operation, control continues at location $(r)(P) + 1$. If an OTN instruction is given and no external device has been properly selected, the computer will be indefinitely delayed and the operation mode (OUT) displayed on the status mode indicator.

<u>F</u>	<u>E</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
76	00	INA	Input to A	*
76	77	OTA	Output from A	*

Description:

Read or write, on the previously selected external device, one word to or from A. The operation takes place under normal input/output using either the normal or buffer channel. On devices which transmit less than one full computer word at a time, such as the 350 paper tape reader, the information is transmitted to and from the low order portion of A. At the completion of the operation, control continues at location $(r)(P) + 1$. If an INA or OTA instruction is being executed and no external device has been properly selected, the computer will be indefinitely delayed and the operation mode (INP, OUT) displayed on the status mode indicator.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
75	00	XXXX	EXC	External Function Constant	2
75	XX		EXF	External Function Forward	2

Description:

Transmit the 12-bit operand to the external equipments. The operand for the various address modes is selected and control continues as described in the section on address modes. At the completion of an external function instruction, A will contain the 12-bit external function code.

The 12-bit operand to be transmitted is known as an external function code. (A complete list of external function codes is found in appendix 2.) The external function instruction is used to select an external device to perform some specific function. With the exception of a status request code, if an illegal selection is attempted the computer will be indefinitely delayed and the operation mode (SEL) displayed on the status mode indicator. One example of an illegal selection is the attempt to select a magnetic tape for reading when the magnetic tape is turned off.

*Execution time varies with the speed of the external equipment being used.

Most external devices have a status request code. When such a code is given and followed with an INA instruction, a 12-bit status response code will be sent to A. By examining this response code it is possible to determine whether further selection of the equipment is possible. A status request may be given even when an external device is turned off.

Only one device may be selected at any one time on each channel. Selection of any device automatically disconnects any other selected device. If a select is given and some other device on the same channel is busy, the computer will be delayed until a selection can be made. If a device is selected for some function and another select is made on the same device for some other function, the previous select is nullified. This applies to status request so that, if a device is selected for reading and then status is requested of that device, another read selection must be made before any further reading can take place on that device.

SELECTIVE STOP AND JUMP INSTRUCTIONS

This group of instructions expands the programming capability by making it possible to select certain conditions under which jumps or stops (or both) will be executed.

<u>F</u>	<u>E</u>	<u>G</u>	<u>Mnemonic</u>	<u>Name</u>	<u>Timing</u>
77	0X		SLS	Selective Stop	1
77	X0	YYYY	SLJ	Selective Jump	(no jump) 1 (jump) 2
77	XX	YYYY	SJS	Selective Stop and Jump	(no jump) 1 (jump) 2

Description:

The selective stop and jump instructions are controlled by six switches on the computer console: Stop switches 1, 2, and 4, and Jump switches 1, 2, and 4.

- 1) SLS: If the appropriate Stop switch is set, stop; otherwise do not stop. If the computer stops, computation may be resumed by placing the Run/Step switch in neutral and then back to the RUN position. Whether the computer stops or not, control will continue at location $(r)(P) + 1$.
- 2) SLJ: If the appropriate Jump switch is set, jump to location $(r)YYYY$; otherwise control continues at location $(r)(P) + 2$.
- 3) SJS: If the appropriate Jump switch is set, set up to jump to location $(r)YYYY$; otherwise set up to continue control at $(r)(P) + 2$. Then test the appropriate Stop switch. If the appropriate Stop switch is set, stop before executing the next instruction; control continues as selected by the Jump switches. If a stop

occurs, computation may be resumed by placing the Run/Step switch in neutral and then back to the RUN position.

The values of X and the switches they control are:

X = 1 test switch 1.

X = 2 test switch 2.

X = 3 test switch 1 and 2. If either switch is set, a stop or jump will be made as defined above.

X = 4 test switch 4.

X = 5 test switch 1 and 4. If either switch is set, a stop or jump will be made as defined above.

X = 6 test switch 4 and 2. If either switch is set, a stop or jump will be made as defined above.

X = 7 test switches 1, 2, and 4. If any are set, a stop or jump will be made as defined above.

The three low order bits of the E portion of the instructions control the testing of the Stop switches and the three high order bits of E control the testing of the Jump switches.

Any combination of X's is legal except 00 and 77, which are treated as a halt instruction (see HLT instruction).

USING THE INSTRUCTIONS

INTERRUPT

The main program must be notified of certain internal and external conditions. An interrupt is the program signal which transfers computer control to some fixed location in memory without losing the information needed to return to the main program.

The 160-A has four interrupt lines: two internal, 10 and 20, and two external, 30 and 40. When an interrupt signal occurs on one of these lines, the computer stores the contents of P at location (d)0010, (d)0020, (d)0030, or (d)0040, depending on the line which generates the interrupt, and then takes its next instruction from (r)0011, (r)0021, (r)0031, or (r)0041.

Interrupt 10

Interrupt 10 is a manual interrupt activated from the 160-A control console by

momentarily pressing any combination of a Selective Stop switch and a Selective Jump switch. When interrupt 10 occurs, the computer stores P at location (d)0010 and transfers to (r)0011.

Interrupt 20

The interrupt 20 line is activated each time a buffer operation is completed or when terminated by an input disconnect. When interrupt 20 occurs, the computer stores P at location (d)0020, and transfers to (r)0021.

Interrupts 30 and 40

External interrupt lines 30 and 40 may be activated by any peripheral device which provides an interrupt signal. (The meaning of these interrupts is a function of the device causing the signal. Since several devices may be connected to each line, each must be interrogated following an interrupt to determine which device generated the signal.)

Interrupt signals are recognized in a priority sequence; the lower numbered lines are recognized first. If an interrupt 10 and 20 occur simultaneously, interrupt 10 will be recognized first. Once an interrupt signal is placed on a line it remains on the line until it is recognized or until a console master clear is performed.

Whenever any interrupt is recognized or whenever an external function instruction is executed, all further interrupts are locked out until a clear interrupt lockout (CIL) is executed. Any interrupt line which becomes active while interrupt lockout is imposed will remain active until a CIL is executed, at which time all interrupt lines will be checked for activity in priority sequence.

Whenever a console master clear is performed, all interrupt lines are set inactive and interrupt lockout is removed.

It is possible to internally impose interrupt lockout for as long as desired by executing any external function instruction (such as a status request) and not executing a CIL.

Once an interrupt has been recognized and the store P and jump executed, the programmer must have some program starting at locations (r)0011, (r)0021, (r)0031, and (r)0041, which will perform the function required by the interrupt.

Example 1: Problem: Whenever an interrupt 10 occurs, set location (i)2100 equal to zero and return to the main program.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(d)0010				Return address put here by interrupt's JPR
(r)0011	JFI	01		Jump to interrupt routine
(r)0012	02	00		Interrupt routine begins at loc (r)0200

(r)0200	STF	05		Save A in (r)0205 (see ****)
(r)0201	LDN	00		Set A equal to zero
(r)0202	STM		2100	Set (i)2100 equal to zero
(r)0204	LDC		****	Restore A
(r)0206	CIL			Clear interrupt lockout
(r)0207	JPI	10		Return to main program

Example 2: Problem: Whenever an interrupt 20 occurs, clear lockout and return to the main program.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>Comments</u>
(d)0020			Return address put here by interrupt's JPR
(r)0021	CIL		Clear lockout
(r)0022	JPI	20	Return to main program

INPUT/OUTPUT

The 160-A has two input/output channels, the normal channel and the buffer channel. Both channels may be used simultaneously for any combination of input/output operations.

A device on the normal channel may be read or written on only the normal channel, but a device on the buffer channel may be read and written on either the normal or the buffer channel. A device connected to the buffer channel may not be read or written on the normal channel if the buffer channel is busy.

The 350 paper tape reader and BRPE-11 paper tape punch which are standard equipment on the 160-A are always connected to the normal channel and cannot be buffered. Any other peripheral device may be connected to either the normal or the buffer channel and may be easily changed from one channel to the other by use of jumper connectors.

For the purpose of recognizing external function codes, all input/output devices are connected to the normal channel output line. When any external equipment selection is made, all other external devices on both the normal and buffer channels are disconnected logically from the computer. They must be reselected for use except when a device on the normal channel is selected while the buffer channel is in operation. In this case, the normal channel is selected, the device on the buffer channel is not disconnected, and the buffer operation is completed.

When an input or output operation is performed on the normal channel, the computer is not free for computation but must wait until the operation is completed. Once an input or output operation is started on the buffer channel, however, the computer is free to either continue computing or perform some other input/output operation on the normal channel.

NOTE

If an Input Disconnect is generated while doing a normal input (72XX) instruction, the address which would have received the next data word will contain all 0's. This applies whether the 72XX uses the normal or the buffer input cable.

No 0's are stored in this manner when doing a buffer input (7200) instruction.

If an automatic disconnect is generated by the 161 typewriter, the last data word will be followed by a location containing the carriage return code (45_g), and by a second location containing all 0's.

The general procedure for performing an input/output operation is as follows:

- 1) Request the status of the selected unit.
- 2) Test the status of the unit for capability of performing the required function.
- 3) Select the unit to perform the input/output function.
- 4) Select the proper storage bank and initiate the input/output operation on the correct channel.
- 5) At the completion of the input/output operation, request the status of the selected unit and test to verify that the input/output operation was successfully completed.

Notice that three of the above five operations are concerned with status requests. To allow the 160-A to have proper input/output control, most peripheral units transmit codes to the 160-A which inform the computer whether or not a unit can be selected and whether or not an input operation was properly completed. External function codes and status response codes are listed in appendix 2.

Checking Status

Checking status is not mandatory for input/output operations; steps 3 and 4 may be used alone. However, if a selected unit is not capable of performing an input/output operation (power to the unit is off, for instance) when an external function instruction selects the unit, the computer will be indefinitely delayed and display SEL on the status mode indicator.

Even if a unit is turned off, its status may be requested and determined.

Example: Test the status of the input/output typewriter and, if status is okay, select the typewriter for output. (Refer to appendix 2 for codes.)

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	EXC		4240	Code 4240 requests typewriter status
(r) a + 2	INA			Read status response to A
(r) a + 3	ZJF	03		Go to select if typewriter ready
(r) a + 4	JPR		W	Return jump to subroutine to determine status trouble
(r) a + 6	EXC		4210	Select typewriter

Continue with program.

If the status response indicates that the unit is not ready, a jump is executed to the subroutine starting at location (r)W to determine the exact status response code and take the necessary action. Note from appendix 2 that a status response code of zero signifies that the typewriter is ready for input/output.

Status is also requested at the completion of an input/output operation to test for conditions which might have occurred during the operation. Not all peripheral devices have codes for such conditions (for example, the typewriter). A peripheral device such as magnetic tape, however, does check for such conditions as:

- End of file
- End of tape
- Parity errors
- Length errors

After an initial status check is made, the external device is selected for input or output and the correct input or output instructions are given to perform the operation.

A status response code may be read using either INA or INP instruction. INP takes more time, however. In addition, a load instruction is necessary to put the response in A for checking the status conditions when INP is used.

Input on the Normal Channel

After the external device has been selected for input, use any combination of the following two instructions to read the incoming information into storage:

- INP - reads from 1 to 7777 words into storage
- INA - reads 1 word into A

When INP is used, the information will be sent to the storage bank specified by the indirect storage bank control (i).

Example: Read 21 frames of paper tape into the area starting at (i)2222, and then read one more frame into A.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	EXC		4102	Select 350 paper tape reader
(r) a + 2	INP	04	2243	2243 is the LWA + 1 of where information is to be stored
(r) a + 4	INA			Read one frame to A
(r) a + 5	PJF	02		Continue
(r) a + 6	22	22		2222 is the FWA of where the information is to be stored. That this location ((r) a + 6) is selected to hold the FWA is indicated by the 04 in the E portion of the INP instruction at (r) a + 2.

Output on the Normal Channel

After the external device has been selected for output, use any combination of the following instructions to write the outgoing information from storage onto the selected device:

- OUT - writes from 1 to 7777 words from storage.
- OTA - writes one word from A.
- OTN - writes one word composed of 6 high order zero bits and the 6 low order bits from the E portion.

Example: Punch 100 frames of paper tape from the area starting at location (i)3200 and then punch 120 frames from the area starting at (i)2701.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	EXC		4104	Select the punch
(r) a + 2	OUT	05	3300	3300 is the LWA + 1 of the first output area
(r) a + 4	OUT	04	3021	3021 is the LWA + 1 of the second output area
(r) a + 6	PJF	03		Exit to next program step
(r) a + 7	32	00		3200 is the FWA of the first output area
(r) a + 10	27	01		2701 is the FWA of the second output area

The Buffer Channel

Perform the following operations before an external device is selected for an input or output operation on the buffer channel:

- 1) Load the Buffer Entrance register with the FWA of the input/output area.
- 2) Load the Buffer Exit register with the LWA + 1 of the input/output area.
- 3) Select the proper storage bank for (b).

After the above steps are completed, the external device is selected and an IBI or IBO instruction starts the buffer operation. It is possible to give the external function instruction to select the external device prior to setting up the buffer registers and (b). However, on certain external equipments, such as magnetic tape, the external function instruction actually starts tape motion and the operation will not be properly completed if the correct data is not available to the tape unit at the time a data transfer request is issued by the tape unit.

If an interrupt 20 is wanted at the completion of the buffer operation, give a CIL instruction following the external function instruction. The execution of any external function instruction automatically sets interrupt lockout. If a CIL instruction is not given, all interrupts remain on the lines and do not take effect until a CIL instruction is given.

Example: Read a message of not more than 50 characters from the input/output typewriter over the buffer channel into storage bank 3, location 3700.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	LDC		3700	FWA of input area
(r) a + 2	ATE		a + 2	Put in BER; if busy, wait
(r) a + 4	LDC		3750	LWA + 1 of input area
(r) a + 6	ATX		a + 6	Put in BXR; if busy, wait
(r) a + 10	SBU	3		Set (b) to 3
(r) a + 11	EXC		4240	Request typewriter status
(r) a + 13	INA			Read status
(r) a + 14	ZJF	04		Go if status okay
(r) a + 15	SBN	20		Test to see if input waiting. If yes, go ahead. If no, status response indicates typewriter not ready.
(r) a + 16	ZJF	02		
(r) a + 17	HLT			Stop if not ready
(r) a + 20	EXC		4220	Select typewriter input
(r) a + 22	CIL			Clear lockout
(r) a + 23	IBI		a + 23	Start buffer operation. Computer is now free to compute and will be interrupted when the buffer operation is complete.
(r) a + 25				Continue program while the input is in progress.

PROGRAMMING EXAMPLES

INTERNAL PROGRAMMING

The following are several internal programming problems with solutions which illustrate various uses of the 160-A instructions. Some of the problems can be programmed in more than one way; but the method chosen, although in some cases not the best, serves well for illustration.

One programming convention, occurring throughout the examples, is used in most utility and general purpose programs developed by Control Data: the locations 0070 - 0077 of all storage banks are used for temporary or transient storage of data, counters, etc. These locations should be avoided for program, table, or constant storage.

Problem: Set up a program switch so that, as the switch is executed, program control is alternately transferred to locations (r)W and (r)V. (W and V are any two arbitrary locations.)

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	SRC		5252	5252 is an alternating pattern of "0" and "1" bits
(r) a + 2	PJF	03		If positive, jump to (r)W
(r) a + 3	JFI	01		Jump to (r)V
(r) a + 4	V			
(r) a + 5	JFI	01		Jump to (r)W
(r) a + 6	W			

Problem: Starting at location (i)0200 are 10_8 words of packed BCD data. Unpack these words into a one character per word format starting at location (i)0300 with the character in the lower half of the word. Assume the unpacked area to be clear. A BCD character is 6 bits long.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	LDC		-10	Set loop counter
(r) a + 2	STD	77		(d)0077 is counter location
(r) a + 3	LDC		0300	FWA of unpacked area
(r) a + 5	STD	76		Put in (d)0076
(r) a + 6	ADN	01		A + 1 A - 0301
(r) a + 7	STD	75		Put in (d)0075
(r) a + 10	LDC		0200	FWA of packed area
(r) a + 12	STD	74		Put in (d)0074
(r) a + 13	LDI	74		Packed word to A
(r) a + 14	HWI	75		Lower character to unpacked area
(r) a + 15	LS6			Shift left 6

(r) a + 16	HWI	76	Upper character to unpacked area
(r) a + 17	AOD	74	Increase addresses
(r) a + 20	LDN	02	
(r) a + 21	RAD	75	
(r) a + 22	LDN	02	
(r) a + 23	RAD	76	
(r) a + 24	AOD	77	Loop counter + 1
(r) a + 25	NZB	12	Return if not done

All done - continue

Problem: Transfer 100_g words from location (i)0700 to location (i)2300 and perform this as a JPR subroutine whose entrance line is a.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a - 1	JFI	01		Exit line
(r) a				Return address stored here
(r) a + 1	LCC		100	Loop count to A
(r) a + 3	STD	77		(d)0077 is count location
(r) a + 4	LDC		0700	FWA or area one
(r) a + 6	STF	05		Initialize a + 13
(r) a + 7	LDC		2300	FWA of area two
(r) a + 11	STF	04		Initialize a + 15
(r) a + 12	LDM		****	Parameterized FWA of area one
(r) a + 14	STM		****	Parameterized FWA of area two
(r) a + 16	AOB	03		FWA + 1
(r) a + 17	AOB	02		FWA + 2
(r) a + 20	AOD	77		Count + 1
(r) a + 21	NZB	07		Loop if not complete
(r) a + 22	ZJB	23		Go to exit - all done

Problem: Count the number of positive, negative, and zero numbers in a table stored in (i)0100 to (i)0200. Put the 3 counts in locations (d)0070, (d)0071, and (d)0072. The program should start at location (r)0500.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r)0500	LDC		0100	Initialize table address
(r)0502	STF	06		Put in load command (r)0510
(r)0503	LDN	00		Set A - 0 to initialize counts
(r)0504	STD	70		Positive count - 0
(r)0505	STD	71		Negative count - 0
(r)0506	STD	72		Zero count - 0
(r)0507	LDM		0100	Table word to A
(r)0511	ZJF	10		Jump if zero to (r)0521
(r)0512	PJF	11		Jump if positive to (r)0523

(r)0513	AOD	71		Negative count + 1
(r)0514	AOB	04		Table address + 1
(r)0515	SBC		0201	Test for end of table
(r)0517	NZB	10		Return to (r)0507 if table not finished
(r)0520	ZJF	05		Exit - all done
(r)0521	AOD	72		Zero count + 1
(r)0522	NZB	06		Return to tally
(r)0523	AOD	70		Positive count + 1
(r)0524	NZB	10		Return to tally

Problem: Set all storage locations in storage bank (1) between the limits (1)0100 and (1)7700 inclusive, equal to zero.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a	LDC		0100	FWA of area to A
(r) a + 2	ATE		a + 2	FWA to BER; if buffer busy, delay
(r) a + 4	LDC		7701	LWA + 1 to A
(r) a + 6	ATX		a + 6	LWA + 1 to BXR; if buffer busy, delay
(r) a + 10	SBU	1		Set (b) - 1
(r) a + 11	LDN	00		Set A - 0
(r) a + 12	BLS		a + 12	Start clear operation using the buffer
(r) a + 14	Next instruction			

INPUT/OUTPUT PROGRAMMING

Paper Tape Reader

The 350 paper tape reader reads 5, 7, or 8 level punched paper tape at a rate of 350 frames per second. Since each frame of paper tape can contain a character, 350 characters per second may be read. Tape made of paper, parchment, Mylar, or Mylar-aluminum laminate is acceptable in any color and either strips or loops of tape may be read.

The 350 reader can be programmed to read continually, stopping on any single character, or it can be programmed to read one character at a time and stop between each character. In order to keep the reader in a continual mode, an input instruction must be given no later than 250 usec following the completion of the previous input instruction. If not, a stop will be initiated and the paper tape will come to a complete stop before the next frame is read.

The 350 reader, which is standard equipment on the 160-A, is always connected to the normal input/output channel. Selection is made at the reader itself to transmit either 5, 7, or 8 bits of information per frame of tape. This data enters the 160-A in the least significant portion of each computer word, one frame of paper tape per word, with the upper bits zero.

The only external function code for the 350 reader is the code 4102. This code selects the reader, provided it has been turned on and is in proper working order. Once the 350 reader has been selected, it will remain selected until another external function instruction is executed. There are no status codes or responses for the 350 reader.

To read information under program control, select the 350 reader. Use either of the instructions INP or INA in any combination to transmit data from paper tape to the 160-A.

A sample 7 level paper tape is shown in figure 6.

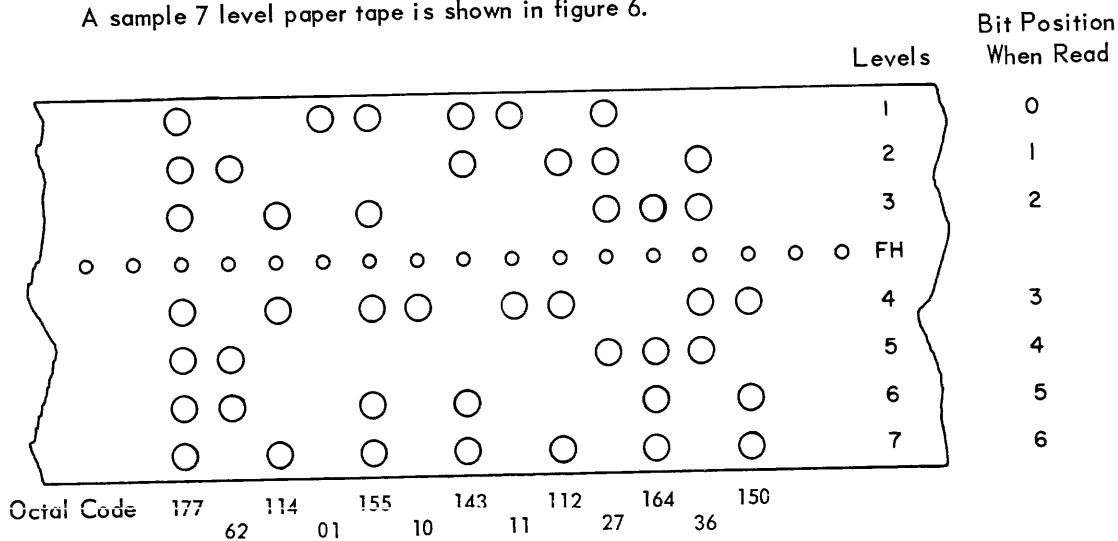


Figure 6. Punched Paper Tape Levels

Example: Read the sample paper tape as shown in figure 6 into consecutive storage locations starting at location (i)0400. Ignore any blank leader. Stop after the last data frame (octal 150) has been read.

Location	F	E	G	Comments
(r) a	EXC		4102	Select reader
(r) a + 2	INA			Read one frame
(r) a + 3	ZJB	01		If frame is part of blank reader, go back
(r) a + 4	STM		0400	Store data starting at (i)0400
(r) a + 6	AOB	01		Increase store address
(r) a + 7	INA			Read next frame
(r) a + 10	NZB	04		If data, go to store
(r) a + 11	HLT			End of data-stop

When the example above has been executed using the paper tape shown in figure 6, the data read will appear as follows in storage :

<u>Location</u>	<u>F</u>	<u>E</u>	<u>Location</u>	<u>F</u>	<u>E</u>
(i)0400	01	77	(i)0407	00	11
(i)0401	00	62	(i)0410	01	12
(i)0402	01	14	(i)0411	00	27
(i)0403	00	01	(i)0412	01	64
(i)0404	01	55	(i)0413	00	36
(i)0405	00	10	(i)0414	01	50
(i)0406	01	43			

Paper Tape Load Format

The 160-A has special circuits which allow automatic loading of programs and data from specially prepared paper tapes. The information is prepared on paper tape in a two-frames-per-word format. The first frame of each word contains a 7th level punch and the six higher order bits of the word; the second frame contains the six lower order bits of the word and no 7th level punch.

Successive words must follow each other on tape. The automatic load will stop when a frame is read which should contain a 7th level punch and none exists. Tape may be placed in the reader any place on the leader; the automatic load will not begin until the first 7th level punch is sensed. Prior to starting automatic load, the FWA where the data is to be stored must be placed in P, and A should be cleared. When the load is completed, P will contain the LWA where data was stored and A will contain a check sum of the data read, modulus $2^{12}-1$.

Example: If the paper tape shown in figure 6 had been read under paper tape load control and P had been set initially to zero, at the completion of the load memory the data would appear as follows:

<u>Location</u>	<u>F</u>	<u>E</u>	<u>Location</u>	<u>F</u>	<u>E</u>
(r)0000	77	62	(r)0004	12	27
(r)0001	14	01	(r)0005	64	36
(r)0002	55	10	(r)0006	50	00
(r)0003	43	11			

Stop Load

Paper Tape Punch

The 110 character per second punch punches 5, 6, 7, or 8 level punched paper tape at 110 frames per second. This punch is always connected to the normal input/output channel and is standard equipment on the 160-A.

Each computer word transmitted to the punch punches one frame of paper tape. The 8 bits punched are taken from the least significant portion of the word. A "1" bit punches a hole in the corresponding level; a "0" bit leaves the corresponding level blank. The upper bits of the word are ignored.

The only external function code for the punch is the code 4104. This code selects the punch, provided it has been turned on and is in proper working order. Once the punch has been selected, it remains selected until another external function instruction is executed. There are no status responses for the punch.

To punch information, first select the punch. Then use any combination of the instructions OTN, OTA, or OUT to transmit data to the punch.

Example: Punch out locations (i)0100 to (i)0200 in paper
 tape load format (see 350 Reader Programming).
 Punch a one-foot leader before and after the data.

<u>Location</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>Comments</u>
(r) a - 1	EXC		4104	Select punch
(r) a + 1	LDC		0144	Number frames for leader (in A)
(r) a + 3	OTN	00		Punch a blank frame
(r) a + 4	SBN	01		Reduce count in A
(r) a + 5	NZB	02		If count not zero, punch more
(r) a + 6	STM		0072	Store
(r) a + 10	LDC		0100	
(r) a + 12	STD	70		FWA of punch area
(r) a + 13	LDI	70		((i)0100) to A
(r) a + 14	STD	71		Put lower half in (d)0071
(r) a + 15	LPC		7700	Clear lower
(r) a + 17	ADN	01		Add 1 for 7th level
(r) a + 20	LS6			Shift 6
(r) a + 21	OTA			Output upper
(r) a + 22	LDD	71		Lower half
(r) a + 23	LPN	77		
(r) a + 24	OTA			
(r) a + 25	AOD	70		FWA + 1
(r) a + 26	SBC		0201	Test for end
(r) a + 30	NZB	15		Go back if not end
(r) a + 31	LDB	27		Number frames for leader
(r) a + 32	OTN	00		
(r) a + 33	SBN	01		
(r) a + 34	NZB	02		Punch leader loop
(r) a + 35	HLT			

End of example.

GLOSSARY

GLOSSARY OF PROGRAMMING TERMS

The following glossary gives the meaning of terms that are used in a relatively specialized sense in this manual.

ADDER	In general, a device used to add two quantities. Specifically, the borrow structure in the subject computer.
ADDRESS	The number designating a storage location; also the storage location itself.
BANK	A unit of core storage with provisions for storing 4096 words. The use of more than one bank permits increasing the storage capacity of a computer without increasing the word length implicitly necessary to extend the range of storage addresses. In the 160-A, this address is effectively increased by a separate instruction (set storage bank control) which determines which bank(s) will be used during a program. Although the total 160-A storage capacity may be extended to eight banks, no more than four may be addressed at any time.
BIT	Binary digit; may be either "1" or "0".
BORROW	In a subtractive counter or accumulator, a signal indicating that in stage n, a "1" was subtracted from a "0".
BUFFER	Noun: A device in which data are stored temporarily in the course of transmission from one point to another. Verb: To store data temporarily.
BUFFERED INPUT/OUTPUT	A term indicating that the computer may carry on high speed computation at the same time it is exchanging data with a peripheral device. In the 160-A, this term must be distinguished from normal I/O, during which the computer cannot engage in computation.
CARRY	In an additive counter or accumulator, a signal indicating that in stage n, a "1" was added to a "1".
CHANNEL	A transmission path that connects the computer to a given external equipment.
CHARACTER	Information handled by the computer: 1) A group of 6 bits representing bi-octal information; may denote a binary quantity, a digit, letter or symbol. In load mode, two 6-bit characters are assembled into one computer word.

2) A group of 7 bits representing an item of information. When the capacity of an input device is 6 or 7 bits, those bits will be deposited in the lower portion of the selected storage address, the remaining bits will be zeros.

CLEAR	A command that removes a quantity from a register by placing every stage in the "0" state.
COMMAND	A signal that performs a unit operation, such as transmitting contents of one register to another, shifting a register, setting a flip-flop.
COMPLEMENT	Noun: See One's Complement or Two's Complement. Verb: A command which produces the one's complement of a given quantity.
CONTENT	The quantity or word held in a register or storage location.
CORE	A small ferromagnetic toroid used as the bistable device for storing a bit in a memory plane.
COUNTER	A register with provisions for increasing or decreasing its content by 1 upon receiving the appropriate command.
DIRECT ADDRESSING	A mode of addressing wherein the execution address portion (E) of the instruction word contains the address of the operand to be acted upon. In the 160-A, the 6-bit execution address limits direct addressing to the first 64 of the possible 4096 storage locations ($2^6 = 64$) in the direct bank.
END-AROUND BORROW	A borrow that is generated in the highest order of an accumulator or counter, and is sent directly to the lowest order stage.
ENTER	To manually place in a register a quantity that is not from storage. In the 160-A, quantities may be entered in only the A, P, and Z registers.
EXECUTION ADDRESS	The lower 6 bits of a 12-bit instruction. Most often used to specify the storage address of an instruction or operand. Sometimes used as the operand.
FUNCTION CODE	The upper 6 bits of a 12-bit instruction.

INDIRECT ADDRESSING	A mode of addressing which extends the length of the execution address (E) to a full computer word, thereby permitting operand references to be made upon any location in storage. All indirect addresses must be contained in storage locations which are available by means of direct addressing. In the 160-A, the constant and memory modes are special forms of indirect addressing.
INPUT DISCONNECT	During an input instruction, a signal sent to the computer by the external device to indicate that the device has completed all available transmissions to the computer.
INPUT REQUEST	A request, by the computer, for information from an external device. Occurs during input instruction only. (See Resume.)
INSTRUCTION	A 12-bit or 24-bit quantity consisting of a function (or operation) code and an execution address.
INTERRUPT	A signal (or class thereof) which, when received and recognized by the computer, forces the computer to forestall its current operation and jump to a subroutine, the starting address of which is determined by the class of the interrupt. A subroutine may have any number of options. It may merely stop the computer, it may determine the nature of the interrupt in order to take corrective measures, or it may return the computer to another phase of the main program.
LOAD	To place a quantity from storage in the A register.
LOCKOUT	Any function (usually of machine logic) that inhibits an action which would normally occur were the lockout not imposed.
LOGICAL PRODUCT	In Boolean algebra, the AND function of several terms. The product is "1" only when all the terms are "1"; otherwise it is "0". Sometimes referred to as the result of "bit-by-bit" multiplication.
LOGICAL SUM	In Boolean algebra, the OR function of several terms. The sum is "1" when any or all of the terms are "1"; it is "0" only when all are "0".

MASK	In the information of the logical products of two quantities, one of them may be used as a mask for the other. The mask determines what part of the other quantity is to be considered. Wherever the mask is "0", that part of the other quantity is cleared, but wherever the mask is a "1", the other quantity is left unaltered.
MASTER CLEAR (MC)	A general command produced by placing the Load/Clear switch in the down (CLEAR) position. An MC clears all of the crucial registers and control FFs to prepare for a new mode of operation.
MODULUS	An integer which describes certain arithmetic characteristics of registers, especially counters and accumulators, within a digital computer. The modulus of a device is defined by r^n for an open ended device and r^n-1 for a closed (end-around) device, where r is the base of the number system used and n is the number of digit positions (stages) in the device. Generally, devices with modulus r^n use two's complement arithmetic procedures, and devices with modulus r^n-1 use one's complement procedures.
NORMAL JUMP	An instruction that jumps from one sequence of instructions to a second, and makes no preparation for returning to the first sequence.
ONE'S COMPLEMENT	With reference to a binary number, that number which results from subtracting each bit of the given number from the bit "1". A negative number is expressed by the one's complement of the corresponding positive number.
OPERAND	Usually refers to the quantity specified by the execution address. This quantity is operated upon in the execution of the instruction.
OPERATION CODE	The upper 6 bits of a 12-bit instruction which identifies the instruction. After the code is translated, it conditions the computer for execution of the specified instruction. The letter F is used to designate this code, which is expressed by two octal digits.
OVERFLOW	The condition in which the capacity of a register is exceeded.
PARTIAL ADD	An addition without carries. Accomplished by toggling each bit of the augend where the corresponding bit of addend is a "1".

PROGRAM	A precise sequence of instructions that accomplishes a computer routine; a plan for the solution of a problem.
READ	To place a quantity from a storage location into a register. The quantity in storage remains unchanged.
READY	The input/output control signal sent by either the computer or an external equipment to alert the device that is to receive a transmission. The ready signal indicates that the word or character has been transmitted.
REFLECTED BINARY (COUNTER)	A reflected binary, or Gray-code counter is one in which only one element changes state for successive counts.
RELATIVE ADDRESSING	A mode of addressing wherein the address of the operand is determined by adding (or subtracting) the contents of the execution address portion (E) of the instruction word to (or from) the instruction address.
REPLACE	In the title of an instruction, the result of the execution of the instruction is stored in the location from which the initial operand was obtained.
RESUME	The output control signal sent by an external equipment to indicate that it is prepared to receive another word or character. The resume signal is thus a request for data. (See Input Request.)
RETURN JUMP	A jump instruction which prepares for continuing the first sequence after the second is completed.
ROUTINE	The sequence of operations which the computer performs under the direction of a program.
SHIFT	To move the bits of a quantity right or left.
SIGN BIT	The bit in the highest-order stage of the register (in registers where a quantity is treated as signed by use of one's complement notation). If the bit is "1", the quantity is negative; if the bit is "0", the quantity is positive.
SIGN EXTENSION	The duplication of the sign bit in the higher-order stages of a register.

STATUS	<p>1) The condition of an external device, as reflected in the response given to a status request interrogation by the computer.</p> <p>2) The condition of the computer as shown by the Status Mode indicator on the console. May variously indicate what it is presently doing, why it stopped, or what it will do when it next starts.</p>
TRANSMISSION, FORCED	A transmission where both set and clear inputs, only one of which will be a "1", are simultaneously gated into a FF which has not been cleared previously.
TRANSLATION	An indication of the content of a group of bit registers. A complete translation gives the exact content, while a partial translation indicates only that the content is within certain limits.
TWO'S COMPLEMENT	That number which results from subtracting each bit of a number from "0". The two's complement may be formed by complementing each bit of the given number and then adding one to the result, performing the required carries.
WORD	A unit of information which has been coded for use in the computer as a series of bits. The normal word length is 12 bits.
WRITE	To enter a quantity into a storage location.

APPENDIX SECTION

APPENDIX I

Table of Instructions

<u>F</u>	<u>E</u>	<u>MNE-</u> <u>MONIC</u>	<u>NAME</u>	<u>TIMING</u>	<u>F</u>	<u>E</u>	<u>MNE-</u> <u>MONIC</u>	<u>NAME</u>	<u>TIMING</u>
00	00	ERR	Error Stop	1	11	00	LPM	Logical Product Memory	3
00	0X	NOP	No Operation	1	11	YY	LPI	Logical Product Indirect	3
00	1X	SRJ	Set Relative Bank Control & Jump	1	12	00	LPC	Logical Product Constant	2
00	2X	SIC	Set Indirect Bank Control	1	12	XX	LPF	Logical Product Forward	2
00	3X	IRJ	Set Indirect & Relative Bank Control & Jump	1	13	00	LPS	Logical Product Specific	2
00	4X	SDC	Set Direct Bank Control	1	13	XX	LPB	Logical Product Backward	2
00	5Y	DRJ	Set Direct & Relative Bank Control & Jump	1	14	YY	SCD	Selective Complement Direct	2
00	6X	SID	Set Indirect & Direct Bank Control	1	15	00	SCM	Selective Complement Memory	3
00	7X	ACJ	Set Direct, Indirect, & Relative Bank Control & Jump	1	15	YY	SCI	Selective Complement Indirect	3
01	00	BLS	Block Store	(no jump) 1+n (jump) 2	16	00	SCC	Selective Complement Constant	2
01	01	PTA	Transfer P to A	1	16	XX	SCF	Selective Complement Forward	2
01	02	LS1	Left Shift One	1	17	00	SCS	Selective Complement Specific	2
01	03	LS2	Left Shift Two	1	17	XX	SCB	Selective Complement Backward	2
01	04	CBC	Clear Buffer Controls	1	20	YY	LDD	Load Direct	2
01	05	ATE	A to Buffer Entrance Register	(no jump) 1 (jump) 2	21	00	LDM	Load Memory	3
01	06	ATX	A to Buffer Exit Register	(no jump) 1 (jump) 2	21	YY	LDI	Load Indirect	3
01	07	ETA	Buffer Entrance Register to A	1	22	00	LDC	Load Constant	2
01	10	LS3	Left Shift Three	1	22	XX	LDF	Load Forward	2
01	11	LS6	Left Shift Six	1	23	00	LDS	Load Specific	2
01	12	MUT	Multiply A by Ten	1	23	XX	LDB	Load Backward	2
01	13	MUH	Multiply A by One Hundred	1	24	YY	LCD	Load Complement Direct	2
01	14	RS1	Right Shift One	1	25	00	LCM	Load Complement Memory	3
01	15	RS2	Right Shift Two	1	25	YY	LCI	Load Complement Indirect	3
01	20	CIL	Clear Interrupt Lockout	1	26	00	LCC	Load Complement Constant	2
01	30	CTA	Bank Controls to A	1	26	XX	LCF	Load Complement Forward	2
01	4X	SBU	Set Buffer Bank Control	1	27	00	LCS	Load Complement Specific	2
01	5X	STP	Store P at Location 5X	3	27	XX	LCB	Load Complement Backward	2
01	6Y	STE	Store Buffer Entrance Register at Location 6X & Transfer A to Buffer Entrance Register	3	30	YY	ADD	Add Direct	2
02	XX	LPN	Logical Product No Address	1	31	00	ADM	Add Memory	3
03	XX	SCN	Selective Complement No Address	1	31	YY	ADI	Add Indirect	3
04	XX	LDN	Load No Address	1	32	00	ADC	Add Constant	2
05	XX	LCN	Load Complement No Address	1	32	XX	ADF	Add Forward	2
06	XX	ADN	Add No Address	1	33	00	ADS	Add Specific	2
07	XX	SBN	Subtract No Address	1	33	XX	ADB	Add Backward	2
10	YY	LPD	Logical Product Direct	2	34	YY	SBD	Subtract Direct	2
					35	00	SBM	Subtract Memory	3
					35	YY	SBI	Subtract Indirect	3
					36	00	SBC	Subtract Constant	2
					36	XX	SBF	Subtract Forward	2
					37	00	SBS	Subtract Specific	2
					37	XX	SBB	Subtract Backward	2
					40	YY	STD	Store Direct	3

APPENDIX I

Table of Instructions (Cont'd)

<u>F</u>	<u>E</u>	<u>MNE- MONIC</u>	<u>NAME</u>	<u>TIMING</u>	<u>F</u>	<u>E</u>	<u>MNE- MONIC</u>	<u>NAME</u>	<u>TIMING</u>
41	00	STM	Store Memory	4	61	XX	NZF	Non-Zero Jump Forward	1
41	YY	STI	Store Indirect	4	62	XX	PJF	Positive Jump Forward	1
42	00	STC	Store Constant	3	63	XX	NJF	Negative Jump Forward	1
42	XX	STF	Store Forward	3	64	XX	ZJB	Zero Jump Backward	1
43	00	STS	Store Specific	3	65	XX	NZB	Non-Zero Jump Backward	1
43	XX	STB	Store Backward	3	66	XX	PJB	Positive Jump Backward	1
44	YY	SRD	Shift Replace Direct	3	67	XX	NJB	Negative Jump Backward	1
45	00	SRM	Shift Replace Memory	4	70	YY	JPI	Jump Indirect	2
45	YY	SRI	Shift Replace Indirect	4	71	00	JPR	Return Jump	3
46	00	SRC	Shift Replace Constant	3	71	XX	JFI	Jump Forward Indirect	2
46	XX	SRF	Shift Replace Forward	3	72	00	IBI	Initiate Buffer Input	(no jump) 1 (jump) 2
47	00	SRS	Shift Replace Specific	3	72	XX	INP	Normal Input	*
47	XX	SRB	Shift Replace Backward	3	73	00	IBO	Initiate Buffer Output	(no jump) 1 (jump) 2
50	YY	RAD	Replace Add Direct	3	73	XX	OUT	Normal Output	*
51	00	RAM	Replace Add Memory	4	74	XX	OTN	Output No Address	*
51	YY	RAI	Replace Add Indirect	4	75	00	EXC	External Function Constant	2
52	00	RAC	Replace Add Constant	3	75	XX	EXF	External Function Forward	2
52	XX	RAF	Replace Add Forward	3	76	00	INA	Input to A	*
53	00	RAS	Replace Add Specific	3	76	YY	HWI	Half Write Indirect	4
53	XX	RAB	Replace Add Backward	3	76	77	OTA	Output from A	*
54	YY	AOD	Replace Add One Direct	3	77	00	HLT	Halt	1
55	00	AOM	Replace Add One Memory	4	77	0X	SLS	Selective Stop	1
55	YY	AOI	Replace Add One Indirect	4	77	X0	SLJ	Selective Jump	(no jump) 1 (jump) 2
56	00	AOC	Replace Add One Constant	3	77	XX	SJS	Selective Stop & Jump	(no jump) 1 (jump) 2
56	XX	AOF	Replace Add One Forward	3	77	77	HLT	Halt	1
57	00	AOS	Replace Add One Specific	3					
57	XX	AOB	Replace Add One Backward	3					
60	XX	ZJF	Zero Jump Forward	1					

NOTE:

- 1) All timings are given in memory cycles where 1 memory cycle equals 6.4 usec.
- 2) All numeric operation codes not listed in the above table will be executed as if they were a NOP instruction.

* Execution time varies with speed of external equipment in use.

APPENDIX I

Table of Instructions Arranged by Functions

<u>F</u>	<u>E</u>	<u>G</u>	<u>MNE-</u> <u>MONIC</u>	<u>NAME</u>	<u>TIMING</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>MNE-</u> <u>MONIC</u>	<u>NAME</u>	<u>TIMING</u>
STOP INSTRUCTIONS						ARITHMETIC INSTRUCTIONS					
00	00		ERR	Error Stop	1	01	12		MUT	Multiply A by 10	1
77	00		HLT	Halt	1	01	13		MUH	Multiply A by One Hundred	1
77	77		HLT	Halt	1	07	XX		SBN	Subtract No Address	1
DATA TRANSMISSION INSTRUCTIONS						34	YY		SBD	Subtract Direct	2
01	00	YYYY	BLS	Block Store	(no jump) 1 (jump) 2	35	00	YYYY	SBM	Subtract Memory	3
01	01		PTA	Transfer P to A	1	35	YY		SBI	Subtract Indirect	3
01	05	YYYY	ATE	A to Buffer Entrance Register	(no jump) 1 (jump) 2	36	00	XXXX	SBC	Subtract Constant	2
01	06	YYYY	ATX	A to Buffer Exit Register	(no jump) 1 (jump) 2	36	XX		SBF	Subtract Forward	2
01	07		ETA	Buffer Entrance Register to A	1	37	00		SBS	Subtract Specific	2
01	30		CTA	Bank Controls to A	1	37	XX		SBB	Subtract Backward	2
01	5Y		STP	Store P at Location 5X	3	06	XX		ADN	Add No Address	1
01	6Y		STE	Store Buffer Entrance Register at Location 6X & Transfer A to Buffer Entrance Register	3	30	YY		ADD	Add Direct	2
04	XX		LDN	Load No Address	1	31	00	YYYY	ADM	Add Memory	3
20	YY		LDD	Load Direct	2	31	YY		ADI	Add Indirect	3
21	00	YYYY	LDM	Load Memory	3	32	00	XXXX	ADC	Add Constant	2
21	YY		LDI	Load Indirect	3	32	XX		ADF	Add Forward	2
22	00	XXXX	LDC	Load Constant	2	33	00		ADS	Add Specific	2
22	XX		LDF	Load Forward	2	33	XX		ADB	Add Backward	2
23	00		LDS	Load Specific	2	50	YY		RAD	Replace Add Direct	3
23	XX		LDB	Load Backward	2	51	00	YYYY	RAM	Replace Add Memory	4
05	XX		LCN	Load Complement No Address	1	51	YY		RAI	Replace Add Indirect	4
24	YY		LCD	Load Complement Direct	2	52	00	XXXX	RAC	Replace Add Constant	3
25	00	YYYY	LCM	Load Complement Memory	3	52	XX		RAF	Replace Add Forward	3
25	YY		LCI	Load Complement Indirect	3	53	00		RAS	Replace Add Specific	3
26	00	XXXX	LCC	Load Complement Constant	2	53	XX		RAB	Replace Add Backward	3
26	XX		LCF	Load Complement Forward	2	54	YY		AOD	Replace Add One Direct	3
27	00		LCS	Load Complement Specific	2	55	00	YYYY	AOM	Replace Add One Memory	4
27	XX		LCB	Load Complement Backward	2	55	YY		AOI	Replace Add One Indirect	4
40	YY		STD	Store Direct	3	56	00	XXXX	AOC	Replace Add One Constant	3
41	00	YYYY	STM	Store Memory	4	56	XX		AOF	Replace Add One Forward	3
41	YY		STI	Store Indirect	4	57	00		AOS	Replace Add One Specific	3
42	00	XXXX	STC	Store Constant	3	57	XX		AOB	Replace Add One Backward	3
42	XX		STF	Store Forward	3	SHIFT INSTRUCTIONS					
43	00		STS	Store Specific	3	01	02		LS1	Left Shift One	1
43	XX		STB	Store Backward	3	01	03		LS2	Left Shift Two	1
76	YY		HWI	Half Write Indirect	4	01	10		LS3	Left Shift Three	1
						01	11		LS6	Left Shift Six	1
						01	14		RS1	Right Shift One	1
						01	15		RS2	Right Shift Two	1
						44	YY		SRD	Shift Replace Direct	3
						45	00	YYYY	SRM	Shift Replace Memory	4

APPENDIX I

Table of Instructions Arranged by Functions (Cont'd)

<u>F</u>	<u>E</u>	<u>G</u>	<u>MNE-MONIC</u>	<u>NAME</u>	<u>TIMING</u>	<u>F</u>	<u>E</u>	<u>G</u>	<u>MNE-MONIC</u>	<u>NAME</u>	<u>TIMING</u>
SHIFT INSTRUCTIONS (cont'd)						JUMP INSTRUCTIONS					
45	YY		SRI	Shift Replace Indirect	4	60	XX		ZJF	Zero Jump Forward	1
46	00	XXXX	SRC	Shift Replace Constant	3	61	XX		NZF	Non-Zero Jump Forward	1
46	XX		SRF	Shift Replace Forward	3	62	XX		PJF	Positive Jump Forward	1
47	00		SRS	Shift Replace Specific	3	63	XX		NJF	Negative Jump Forward	1
47	XX		SRB	Shift Replace Backward	3	64	XX		ZJB	Zero Jump Backward	1
LOGICAL INSTRUCTIONS						INPUT/OUTPUT INSTRUCTIONS					
02	XX		LPN	Logical Product No Address	1	01	04		CBC	Clear Buffer Controls	1
10	YY		LPD	Logical Product Direct	2	01	20		CIL	Clear Interrupt Lockout	1
11	00	YYYY	LPM	Logical Product Memory	3	72	00	YYYY	IBI	Initiate Buffer Input	(no jump) 1 (jump) 2
11	YY		LPI	Logical Product Indirect	3	73	00	YYYY	IBO	Initiate Buffer Output	(no jump) 1 (jump) 2
12	00	XXXX	LPC	Logical Product Constant	2	72	XX	YYYY	INP	Normal Input	*
12	XX		LPF	Logical Product Forward	2	73	XX	YYYY	OUT	Normal Output	*
13	00		LPS	Logical Product Specific	2	74	XX		OTN	Output No Address	*
13	XX		LPB	Logical Product Backward	2	76	00		INA	Input to A	*
03	XX		SCN	Selective Complement No Address	1	76	77		OTA	Output from A	*
14	YY		SCD	Selective Complement Direct	2	75	00	XXXX	EXC	External Function Constant	2
15	00	YYYY	SCM	Selective Complement Memory	3	75	XX		EXF	External Function Forward	2
15	YY		SCI	Selective Complement Indirect	3	SELECTIVE STOP AND JUMP INSTRUCTIONS					
16	00	XXXX	SCC	Selective Complement Constant	2	00	0X		NOP	No Operation	1
16	XX		SCF	Selective Complement Forward	2	77	0X		SLS	Selective Stop	1
17	00		SCS	Selective Complement Specific	2	77	X0	YYYY	SLJ	Selective Jump	(no jump) 1 (jump) 2
17	XX		SCB	Selective Complement Backward	2	77	XX	YYYY	SJS	Selective Stop & Jump	(no jump) 1 (jump) 2
STORAGE BANK CONTROL INSTRUCTIONS											
00	1X		SRJ	Set Relative Bank Control & Jump	1						
00	2X		SIC	Set Indirect Bank Control	1						
00	3X		IRJ	Set Indirect & Relative Bank Control & Jump	1						
00	4X		SDC	Set Direct Bank Control	1						
00	5X		DRJ	Set Direct & Relative Bank Control & Jump	1						
00	6X		SID	Set Indirect & Direct Bank Control	1						
00	7X		ACJ	Set Direct, Indirect, & Relative Bank Control & Jump	1						
01	4X		SBU	Set Buffer Bank Control	1						

* Execution time varies with speed of external equipment in use.

APPENDIX II

Table of External Function Codes and Status Responses

1. 350 PAPER TAPE READER

- A. External Function Codes
4102 Select Reader

NOTE: There are no status responses for this equipment.

2. BRPE-11 PAPER TAPE PUNCH

- A. External Function Codes
4104 Select Paper Tape Punch

NOTE: There are no status responses for this equipment.

3. 161 INPUT/OUTPUT TYPEWRITER

- A. External Function Codes
4210 Select typewriter output
4220 Select typewriter input
4240 Request typewriter status

- B. Status Response Codes
0000 Typewriter ready
0004 Typewriter power off
0010 Typewriter not in computer status
0020 Input character ready
0040 Output in use

NOTE: If a second typewriter is added, the master bits will be 43.

4. 162 MAGNETIC TAPE SYNCHRONIZER

- A. External Function Codes
Y11X Write if OUT is given
Y11X Write End-of-File mark if no OUT is given
Y12X Backspace one record if INA is given
Y12X Search backward to End-of-File mark if no INA is given
Y13X Read forward if INPUT is given
Y13X Search forward to End-of-File mark if no INPUT is given
Y14X Request status
Y15X Rewind unload
Y16X Rewind load
Y171 Set tapes to odd parity
Y172 Set tapes to even parity
210X High density
110X Low density

- B. Status Response Codes
0000 Odd parity selected - no errors
0001 Even parity selected - no errors

- 0002 Tape X not ready
0004 Parity error
0015 Illegal BCD detected on Write
0020 End-of-File mark read
0040 End-of-Tape or Load Point sensed
0100 High density
0200 Tape X busy

NOTES: Y=1: 6-bit mode.

Y=2: 12-bit mode.

X=(0 to 7): designates one of the four (eight) 60X's. The master bits 12, 13, 22, and 23 are used for second and third tape control. If the tape transport is a 606, a 6-bit, high density selection is illegal (a programmer consideration).

5. 165 PLOTTER

- A. External Function Codes
4401 Select Plotter for Write operation
4440 Select Plotter for Read operation

B. Follow 4401 with Output instruction and transmit one of these:

- 0001 Move carriage and pen .01" in +X direction
0002 Move carriage and pen .01" in -X direction
0004 Rotate drum .01" in -Y direction
0005 Carriage and pen move .01" in +X direction, drum rotates in -Y direction .01"
0006 Carriage and pen move .01" in -X direction, drum rotates in -Y direction .01"
0010 Rotate drum .01" in +Y direction
0011 Carriage and pen move .01" in +X direction, drum rotates in +Y direction .01"
0012 Carriage and pen move .01" in -X direction, drum rotates in +Y direction .01"
0020 Move pen down to paper
0040 Move pen away from paper

C. Status Response Codes

Status is obtained by selecting the unit for reading. The obtained status is the value of the 12 switches on the unit.

6. 166-2 LINE PRINTER

- A. External Function Codes
 - 0700 Asynchronous print
 - 0710 Synchronous print
 - 0740 Check status
 - 072X Advance forms
- B. Status Response Codes
 - 0000 166-2 ready
 - 0001 Buffer busy
 - 0002 Out of paper
 - 0004 Paper moving
 - 0010 Drum stationary
 - 0020 Off-line

7. 167-1 CARD READER

- A. External Function Codes
 - 4500 EF clear
 - 4501 Free run read
 - 4502 Single cycle read
 - 4540 Check status
- B. Status Response Codes
 - 0000 Card Reader ready
 - 0001 Hopper empty
 - 0002 Stacker full
 - 0004 Feed failure
 - 0010 Program error
 - 0020 Amplifier failure
 - 0040 Motor power off

8. 167-2 CARD READER (Hollerith Facility)

- A. External Function Codes
 - 4500 EF clear
 - 4501 Free run read
 - 4502 Single cycle read
 - 4504 NEGATE translate, H→BCD
 - 4505 FRR, H→BCD and pack
 - 4506 SCR, H→BCD and pack
 - 4540 Check status
- B. Status Response Codes
 - 0000 Card Reader ready
 - 0001 Hopper empty
 - 0002 Stacker full
 - 0004 Feed failure
 - 0010 Program error
 - 0020 Amplifier failure
 - 0040 Motor power off

9. 168-1 AUXILIARY ARITHMETIC UNIT

- A. External Function Codes
 - 3300 Short divide
 - 3301 Short multiply
 - 3302 Long divide
 - 3303 Long multiply
 - 3304 Status request
 - 3310 Reselect*
 - 3323 Addition
 - 3363 Subtraction
- B. Status Response Codes
 - 0000 Unit ready
 - 0004 Add/Subtract overflow
 - 0010 Divide fault
 - 0020 Unload not completed
 - 0040 Busy computing

10. 168-2 AUXILIARY ARITHMETIC UNIT

- A. External Function Codes
 - 3300 Divide
 - 3301 Multiply
 - 3302 Left shift
 - 3303 Right shift
 - 3304 Status request
 - 3310 Reselect*
 - 3321 Addition
 - 3342 Norm. & Count
 - 3361 Subtraction
 - 3314 Divide result
 - 3315 Multiply result
 - 3316 Left shift result
 - 3317 Right shift result
 - 3335 Add to result
 - 3356 Norm. result & count
 - 3375 Subt. from result
- B. Status Response Codes
 - 0000 Unit ready
 - 0010 Add/Subtract overflow
 - 0020 Unload not completed
 - 0040 Busy computing
 - 4000 Divide fault**

*Reselect is used if another external equipment has been selected prior to receiving the result of a selected 168 operation. It cannot be used to initially select the unit.

**A divide fault occurs when the divisor is equal to or smaller than the most significant 27 bits of the dividend.

11. 169 AUXILIARY MEMORY UNIT

A. External Buffer Select Codes

- 4701 Select external buffer mode
- 4702 Clear external buffer controls
- 4704 Select BER read
- 4710 Select channel extension mode
- 4720 Clear channel extension mode
- 4740 Select external buffer status

B. External Buffer Status Responses

- 1XXX Interrupt from other computer
- 2XXX Interrupt from peripheral equipment
- 4XXX Buffer interrupt
- X1XX This CHX active
- X2XX Other CHX active
- X4XX Illegal bank selection
- XX0X OBA-T
- XX1X IBA-T
- XX2X OBA-NT
- XX3X IBA-NT
- XX4X Buffer initiation
- XXX2-7 External bank selection
- 0000 External buffer ready

12. 170 CARD PUNCH CONTROL UNIT

A. External Function Codes

- 3002 Punch
- 3040 Check status

B. Status Response Codes

- 0000 170 ready
- 0200 MS in 1604 position
- 2000 Punch not ready

13. 177 CARD READER

A. External Function Codes

- 4500 EF clear
- 4501 Free run read
- 4502 Single cycle read
- 4505 Negate translate, H → BCD, free run read
- 4506 Negate translate, H → BCD, single cycle read
- 4510 Gate card
- 4540 Status request

B. Status Response Codes

- 0001 Input tray empty
- 0002 Primary or secondary stacker full
- 0004 Feed failure
- 0010 Late input request
- 0020 Pre-read error
- 0040 Manual on or motor power off
- 0100 Read comparison error
- 0200 End of file
- 0400 Ready

14. 1610 CONTROL UNIT

A. External Function Codes

- 0301 Read from primary read
- 0302 Read from secondary read
- 0340 Request status of input
- 3001 Print
- 3002 Punch
- 3040 Request status of output

B. Status Response Codes

- 0000 All units ready
- 0001 Reader not ready
- 0020 1604 selected on input
- 0200 1604 selected on output
- 2000 Punch not ready
- 4000 Printer not ready

15. 1612 HIGH SPEED PRINTER

A. External Function Codes

- 0600 Select printer and do not interrupt on ready
- 0601 Space paper one line
- 0602 Space paper two lines
- 0603 Skip to format channel 7
- 0604 Skip to format channel 8
- 0605 Print information and advance paper
- 0606 Do not advance paper after next print
- 0607 Select printer and interrupt on ready
- 0610 Clear monitor channels 1-6
- 0611 Select monitor channel 1
- 0612 Select monitor channel 2
- 0613 Select monitor channel 3
- 0614 Select monitor channel 4
- 0615 Select monitor channel 5
- 0616 Select monitor channel 6

B. Status Response Codes

- 0000 Printer not ready
- 4000 Printer ready

NOTE: Status is always available on the 1612.
No request is necessary.

16. 1614 CARD READER

EF and Status Codes same as 177 card reader.

17. 1615 MAGNETIC TAPE CONTROLLER

A. 160 External Function Codes

(N = 1→7)

Write Operations

- 60N1 Select tape N to write binary
- 60N2 Select tape N to write coded
- 6001 Prepare selected tape to write binary
- 6002 Prepare selected tape to write coded
- 6003 Write End-of-File on selected tape
- 6005 Rewind selected write tape
- 6006 Backspace selected write tape
- 6007 Rewind-unload selected write tape
- 6010 Set Low Density on selected write tape
- 6020 Set High Density on selected write tape
- 6030 Skip bad spot on selected write tape
- 6053 Request status

Read Operations

- 50N1 Select tape N to read binary one record
- 50N2 Select tape N to read coded one record
- 52N1 Select tape N to read binary one file
- 52N2 Select tape N to read coded one file
- 5001 Prepare selected tape to read binary one record
- 5002 Prepare selected tape to read coded one record
- 5201 Prepare selected tape to read binary one file
- 5202 Prepare selected tape to read coded one file
- 5003 Move selected read tape forward one record
- 5203 Search file mark forward
- 5005 Rewind selected read tape
- 5006 Backspace selected read tape
- 5206 Search file mark backward
- 5007 Rewind-unload selected read tape
- 5010 Set Low Density on selected read tape
- 5020 Set High Density on selected read tape

B. Status Response Codes

- X2XX Ready to read
- X1XX Ready to write
- XX4X Read parity error
- XX2X Write reply parity error
- XX1X End-of-File mark
- XXX4 End-of-Tape mark

The following additional select and status codes are available under the program control mode of operation:

A. External Function Codes

Write Operations

- 6052 Release write control to 1604
- 6050 Release action request
- 5051 Set communication flag I
- 6055 Clear communication flag I
- 6051 Set communication flag II
- 6056 Clear communication flag II

Read Operations

- 5052 Release read control to 1604
- 5053 Select interrupt

17. 1615 MAGNETIC TAPE CONTROLLER (Cont'd)

B. Status Response Codes

Write Operations

2XXX Write control available
XXX1 Communication flag I set

Read Operations

4XXX Read control available
1XXX Direct 160 to 1604
X4XX Direct 1604 to 160
XXX2 160 action request

18. 1619 DISK FILE CONTROLLER

A. Select Codes

7000 Select disc file - positioner power on
7001 Select clear disc file - positioner
power off
7002 Select check mode type I - no buffer
Select check mode type II - input
buffer activated
7003 Select checkword check
7004* Select interrupt on next available
7005* Clear interrupt on next available
7006* Select interrupt on next ready
7007* Clear interrupt on next ready
7010* Select interrupt on next fault
7011* Clear interrupt on next fault
7020 Request status

B. Status Response Codes

XXX0 Ready
XXX1 Not ready
XXX2 160 not selected
XXX4 1604 selected
XX1X Lost data
XX2X Checkword error
XX4X Internal fault
1XXX File off-line
2XXX File warning
4XXX Write lockout

* 160-A only

Use This Page for Additional Codes

Use This Page for Additional Codes

APPENDIX III

TABLE OF POWERS OF 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

APPENDIX V

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

APPENDIX V

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

APPENDIX V

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX VI

Input/Output Typewriter Codes

CHARACTERS		CODE	CHARACTERS		CODE
UC	LC		UC	LC	
A	a	30	X	x	27
B	b	23	Y	y	25
C	c	16	Z	z	21
D	d	22)	0	56
E	e	20	*	1	74
F	f	26	@	2	70
G	g	13	#	3	64
H	h	05	\$	4	62
I	i	14	%	5	66
J	j	32	¢	6	72
K	k	36	&	7	60
L	l	11	½	8	33
M	m	07	(9	37
N	n	06	-	-	52
O	o	03	?	/	44
P	p	15	"	'	54
Q	q	35	°	+	46
R	r	12	.	.	42
S	s	24	:	;	50
T	t	01	,	,	40
U	u	34	÷	=	02
V	v	17	tab	tab	51
W	w	31	space		04
Backspace		61	Carriage Return		45
Lower Case		57	Upper Case		47

APPENDIX VII

1612 Printer Codes

CHARACTER	DATA (Bits 0-5) (Octal)	ROW (Print Cylinder) (& Memory)	1612 MASTER TIMING GENERATOR COUNTER (Stages 1-6) (Octal)
:	00	48	57
1	01	2	01
2	02	3	02
3	03	4	03
4	04	5	04
5	05	6	05
6	06	7	06
7	07	8	07
8	10	9	10
9	11	10	11
0	12	1	00
=	13	40	47
≠	14	46	55
<	A 15	49	60
%	A 16	50	61
[A 17	51	62
blank	20	64	77
/	21	43	52
S	22	29	34
T	23	30	35
U	24	31	36
V	25	32	37
W	26	33	40
X	27	34	41
Y	30	35	42
Z	31	36	43
]	A 32	52	63
,	33	45	54
(34	41	50
→	A 35	53	64
≡	A 36	54	65
Λ	A 37	55	66
-	40	38	45
J	41	20	23
K	42	21	24
L	43	22	25

A - denotes Additional Characters

APPENDIX VII

1612 Printer Codes (Cont'd)

<u>CHARACTER</u>	<u>DATA (Bits 0-5) (Octal)</u>	<u>ROW (Print Cylinder) (& Memory)</u>	<u>1612 MASTER TIMING GENERATOR COUNTER (Stages 1-6) (Octal)</u>
M	44	23	26
N	45	24	27
O	46	25	30
P	47	26	31
Q	50	27	32
R	51	28	33
V	A 52	56	67
\$	53	47	56
*	54	44	53
↑	A 55	58	71
↓	A 56	59	72
>	A 57	60	73
+	60	39	46
A	61	11	12
B	62	12	13
C	63	13	14
D	64	14	15
E	65	15	16
F	66	16	17
G	67	17	20
H	70	18	21
I	71	19	22
<	A 72	61	74
.	73	37	44
)	74	42	51
∞	A 75	62	75
J	A 76	57	70
:	A 77	63	76

A - denotes Additional Characters

APPENDIX VIII

166 PRINT-DRUM CHARACTER ARRANGEMENTS AND CODES

At the time of printing this volume, four drums were available for the printer. The customer has a choice between the first two standard drums at the time of ordering. The latter two drums are available as optional equipment. The following charts list the characters in order as they appear around the circumference of the drums. To the right of each character is the octal code found on the accompanying code wheel.

CODE DISC 46M50 (48 CHARACTER DRUM) 200 Imp

Drum Position	Character	Octal Code	Drum Position	Character	Octal Code
1	A	61	25	Y	30
2	B	62	26	Z	31
3	C	63	27	0	00
4	D	64	28	1	01
5	E	65	29	2	02
6	F	66	30	3	03
7	G	67	31	4	04
8	H	70	32	5	05
9	I	71	33	6	06
10	J	41	34	7	07
11	K	42	35	8	10
12	L	43	36	9	11
13	M	44	37	.	7
14	N	45	38	-	40
15	O	46	39	+	60
16	P	47	40	=	13
17	Q	50	41	(34
18	R	51	42)	74
19	S	22	43	/	21
20	T	23	44	*	54
21	U	24	45	,	33
22	V	25	46	≠	14
23	W	26	47	\$	53
24	X	27	48	0	12

APPENDIX VIII

166 PRINTER CODES (Cont'd)

CODE DISC 46M39 (64 CHARACTER DRUM) 150 Imp

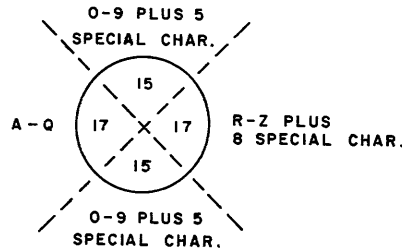
Drum Position	Character	Octal Code	Drum Position	Character	Octal Code
1	Blank	20	33	L	43
2	0	12	34	M	44
3	1	01	35	N	45
4	2	02	36	O	46
5	3	03	37	P	47
6	4	04	38	Q	50
7	5	05	39	R	51
8	6	06	40	S	22
9	7	07	41	T	23
10	8	10	42	U	24
11	9	11	43	V	25
12	.	73	44	W	26
13	-	40	45	X	27
14	+	60	46	Y	30
15	=	13	47	Z	31
16	≠	14	48	(34
17	/	21	49)	74
18	~	37	50	,	33
19	%	52	51	?	76
20	\$	53	52	;	77
21	*	54	53	'	16
22	A	61	54	:	00
23	B	62	55]	32
24	C	63	56	→	35
25	D	64	57	=	36
26	E	65	58	↑	55
27	F	66	59	↓	56
28	G	67	60	>	57
29	H	70	61	<	72
30	I	71	62	≥	75
31	J	41	63	≤	15
32	K	42	64	—	17

APPENDIX VIII

166 PRINTER CODES (Cont'd) OPTIONAL DRUM 42M121A (DRUM TYPE B)

64 Character Drum: one alpha field, two numeric fields, 48 different characters.

Layout Drawing:



Drum Position	Character	Octal Code	Drum Position	Character	Octal Code
1	A	61	33	R	51
2	B	62	34	S	22
3	C	63	35	T	23
4	D	64	36	U	24
5	E	65	37	V	25
6	F	66	38	W	26
7	G	67	39	X	27
8	H	70	40	Y	30
9	I	71	41	Z	31
10	J	41	42	+	60
11	K	42	43	=	13
12	L	43	44	%	52
13	M	44	45	\$	53
14	N	45	46	(34
15	O	46	47)	74
16	P	47	48	/	21
17	Q	50	49	blank	77
18	0	12	50	0	12
19	1	01	51	1	01
20	2	02	52	2	02
21	3	03	53	3	03
22	4	04	54	4	04
23	5	05	55	5	05
24	6	06	56	6	06
25	7	07	57	7	07
26	8	10	58	8	10
27	9	11	59	9	11
28	.	73	60	.	73
29	-	40	61	-	40
30	,	33	62	,	33
31	*	54	63	*	54
32	:	00	64	:	00

Result: 26 alpha, 10 numerics, and 13 special characters which are . - , : + = % ; \$ () / * total 48 characters.

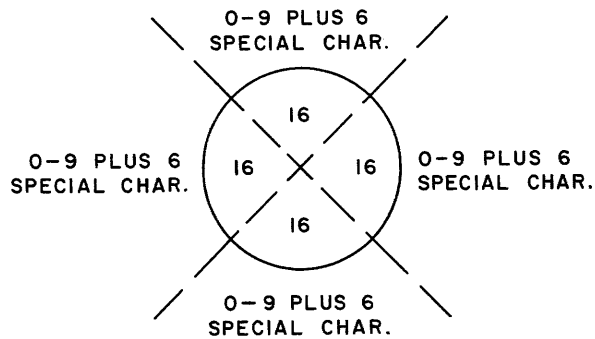
NOTE: Code 20 will give blanks even though no blank is left on the drum. Any other code for a character which is not on the drum will also result in blanks.

APPENDIX VIII

OPTIONAL DRUM 42M122A (DRUM TYPE C)

64 Character Drum: four numeric fields, 16 different characters.

Layout Drawing:



Drum Position	Character	Octal Code	Drum Position	Character	Octal Code
1, 17, 33, 49	0	12	9, 25, 41, 57	8	10
2, 18, 34, 50	1	01	10, 26, 42, 58	9	11
3, 19, 35, 51	2	02	11, 27, 43, 59	.	73
4, 20, 36, 52	3	03	12, 28, 44, 60	-	40
5, 21, 37, 53	4	04	13, 29, 45, 61	,	33
6, 22, 38, 54	5	05	14, 30, 46, 62	+	60
7, 23, 39, 55	6	06	15, 31, 47, 63	\$	53
8, 24, 40, 56	7	07	16, 32, 48, 64	*	54

Result: 10 numeric and 6 special characters.

NOTE: Code 20 or any other illegal code will result in blanks being printed.

APPENDIX VIII

Flexowriter Codes

UC	LC	CODE	UC	LC	CODE
A	a	30	Y	y	25
B	b	23	Z	a	21
C	c	16	o	0	56
D	d	22	1	1	74
E	e	20	2	2	70
F	f	26	3	3	64
G	g	13	4	4	62
H	h	05	5	5	68
I	i	14	6	6	72
J	j	32	7	7	60
K	k	36	8	8	33
L	l	11	9	9	37
M	m	07	-	-	52
N	n	06	'	/	44
O	o	03	()	54
P	p	15	+	,	46
Q	q	35	=	.	42
R	r	12	:	;	50
S	s	24	CR		45
T	t	01	Upper Case (UC)		47
			Lower Case (LC)		57
			Back Space (BS)		61
U	u	34	Color Shift (CS)		02
V	v	17	Tabulate (TAB)		51
			Stop		43
			Space		04
W	w	31	Tape Feed		00
X	x	27	Delete		77

- Note:
- 1) Leader - Blank tape, Delete - Deleted character
Stop - Stop Flexowriter reader,
 - 2) 10, 40, 41, 53, 55, 63, 65, 67, 71, 73, 75, and 76 - illegal

APPENDIX IX

Magnetic Tape BCD Codes

Character	Code (Octal)	Character	Code (Octal)
A	61	2	02
B	62	3	03
C	63	4	04
D	64	5	05
E	65	6	06
F	66	7	07
G	67	8	10
H	70	9	11
I	71	&	60
J	41	-	40
K	42	(blank)	20
L	43	/	21
M	44	. (period)	73
N	45	\$	53
O	46	*	54
P	47	, (comma)	33
Q	50	%	34
R	51	#	13
S	22	@	14
T	23	⌘	74
U	24	0 (numerical zero)	12
V	25	record mark	32
W	26	0 (minus zero)	52
X	27	0 (plus zero)	72
Y	30	group mark	77
Z	31	tape mark	17
0	12		
1	01		

APPENDIX X

Punched Card Codes

Char	Card	BCD	Char	Card	BCD	Char	Card	BCD	Char	Card	BCD
			+	12	80	-	11	40			20
1	1	01	A	12 1	61	J	11 1	41	/	0 1	21
2	2	02	B	12 2	62	K	11 2	42	S	0 2	22
3	3	03	C	12 3	63	L	11 3	43	T	0 3	23
4	4	04	D	12 4	64	M	11 4	44	U	0 4	24
5	5	05	E	12 5	65	N	11 5	45	V	0 5	25
6	6	06	F	12 6	66	O	11 6	46	W	0 6	26
7	7	07	G	12 7	67	P	11 7	47	X	0 7	27
8	8	10	H	12 8	70	Q	11 8	50	Y	0 8	30
9	9	11	I	12 9	71	R	11 9	51	Z	0 9	31
0	0	12									
=	8,3	13	.	12 8,3	73	\$	11 8,3	53	,	0 8,3	33
—	8,4	14)	12 8,4	74	*	11 8,4	54	(0 8,4	34

COMMENT SHEET

MANUAL TITLE CONTROL DATA 160-A COMPUTER
Reference Manual

PUBLICATION NO. 60014500 REVISION _____

FROM: NAME: _____
BUSINESS ADDRESS: _____

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

FORM CA231 REV. 1-67

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

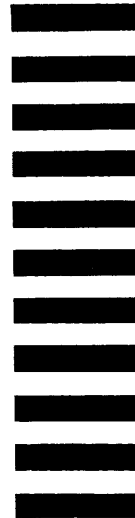
FOLD

FIRST CLASS
PERMIT NO. 8241
MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY
CONTROL DATA CORPORATION
8100 34TH AVENUE SOUTH
MINNEAPOLIS, MINNESOTA 55440

ATTN: TECHNICAL PUBLICATIONS DEPT.
PLANT TWO



CUT ALONG LINE

FOLD

FOLD

INDEX

Accumulator (see A Register)		MCS Mode Indicator	3-4
Add Instructions	3-17	Multiply A by 10 (MUT 0112)	3-16
Addressable Registers	1-2	Multiply by 100 (MUH 0113)	3-16
Addressing Modes (see also Instruction Word Format)	2-3	No Operation (NOP 000X)	3-11
Address Register (S)	1-4	Negative Jumps (NJF 61XX) (NJB 67XX)	3-23
A Register	1-2	Non-Zero Jumps (NZF 61XX) (NZB 65XX)	3-23
A Register Display (also A' Display)	3-2	Normal Input (INP 72XX)	3-26
A' Register	1-4	Normal Output (OUT 73XX)	3-26
A to BER (ATE 0105)	3-12	Operation (see Contents pages for heading desired)	3-1
A to BXR (ATX 0106)	3-13	Output from A (OTA 7677)	3-28
Arithmetic Instructions	3-16	Output No Address (OTN 74XX)	3-27
Bank Controls to A (CTA 0130)	3-13	Output on Normal Channel	3-35
BER Display	3-2	Paper Tape Load Format	3-41
BER to A (ETA 0107)	3-13	Paper Tape Punch	3-41
BFR Display	3-2	Paper Tape Reader	3-39
Block Store Instruction (BLS 0100)	3-12	Positive Jumps (PJF 62XX) (PJB 66XX)	3-23
Buffer Channel, use of	3-35	P Register	1-2
Buffer Data Register (BFR)	1-4	P Register Display	3-2
Buffer Entrance Register (BER)	1-4	Programming, Input/Output: using Paper Tape Reader	3-39
Buffer Exit Register (BXR)	1-4	using Paper Tape Punch	3-41
BXR Display	3-2	Programming, Internal (with sample problems)	3-37
Clear Buffer Controls (CBC 0104)	3-25	Register Display	3-1
Clear Interrupt Lockout (CIL 0120)	3-25	Registers (see specific register name)	
Data Transmission Instructions	3-11	Replace Add Instructions	3-18
Error Stop (ERR 0000)	3-11	Replace Add One Instructions	3-18
External Function Instructions, explanation of	3-28	Return Jump (see Jump, Return)	
F Register Display	3-1	Selective Complement Instructions	3-21
Function Register (F)	1-4	Selective Jump (SLJ 77X0)	3-29
Half Write Indirect Instruction (HWI 76XX)	3-15	Selective Jump and Stop (SJS 77XX)	3-29
Halt (HLT 7700) (HLT 7777)	3-11	Selective Stop (SLS 770X)	3-29
Initiate Buffer Input (IBI 7200)	3-25	Shift Instructions	3-19
Initiate Buffer Output (IBO 7300)	3-26	Shift Replace Instructions	3-20
Input on Normal Channel	3-34	S Register	1-4
Input/Output Instructions, see also specific I/O instruction	3-25	S Register Display	3-2
Input/Output, explanation of	3-32	Status, checking	3-33
Input/Output Programming	3-39	Status Mode Indicator	3-3
Input to A (INA 7600)	3-28	Status Request	3-29
Instruction Word Format (with examples of addressing modes)	2-4	Status Response	3-29
Internal Programming	3-26	Stop Instructions	3-11
Interrupt, explanation of	3-30	Storage Bank Control Instructions	3-21
Jump Instructions	3-23	Store BER (STE 016X)	3-14
Jump Forward Indirect (JFI 71XX)	3-24	Store Instructions	3-15
Jump Indirect (JPI 70XX)	3-23	Store P (STP 015X)	3-14
Jump, Return (JPR 7100)	3-24	Subtract Instructions	3-17
Load Complement Instructions	3-15	Switch Panel, Use of switches on	3-5
Load Instructions	3-14	Transfer P to A (PTA 0101)	3-12
Load Mode (Paper Tape Load Format)	3-41	Word Format	2-4
Logical Product Instructions	3-20	Zero Jumps (ZJF 60XX) (ZJB 64XX)	3-23
		Z Register	1-4
		Z Register Display	3-2

CONTROL DATA

3/4 1/2
1-1/4

->> CUT OUT FOR USE AS LOOSE-LEAF BINDER TITLE TAB

CONTROL DATA
CORPORATION

**CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD**

LITHO IN U.S.A