CONTROL DATA 160-A COMPUTER

PROGRAMMING MANUAL

145e
Rev 3/63

Minor Revision (March, 1963)
This edition, publication 145e, is a minor revision but does not obsolete publication 145d. Pages changed in this edition are marked by a dot in the upper outside corner.

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Technical Publications Department
501 Park Ave.
Minneapolis 15, Minnesota

CONTENTS

CHAPTER 1 – DESCRIPTION

CHAPTER 2 – DESCRIPTION OF INSTRUCTIONS

CHAPTER 3 – OPERATION

CHAPTER 3 – OPERATION (Cont'd)

GLOSSARY OF PROGRAMMING TERMS

APPENDIXES

INDEX

FIGURES

160-A COMPUTER

Chapter 1

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

Internal and external interrupt

An expandable magnetic core memory (8,192 to 32,768 words)

A 350 character/second paper tape reader

A 110 character/second paper tape punch

Transistor-diode logic

A 160-A system may be expanded to include the following external equipment:

Up to 40 magnetic tape stations

Input/output typewriters

Punched card readers, punches, and low speed line printers

High speed (1000 lines/minute) printers

Plotting and digital display equipment

Analog-to-digital and digital-to-analog equipment

Two or more Control Data computers operating together in a satellite system

Two CONTROL DATA 160-A Computers operating independently but sharing the same magnetic core memory

*Registered trademark of Control Data Corporation

PHYSICAL DESCRIPTION

Physical characteristics and power requirements are:

Height – 29 inches
Width – 30 inches
Length – 61⅝ 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

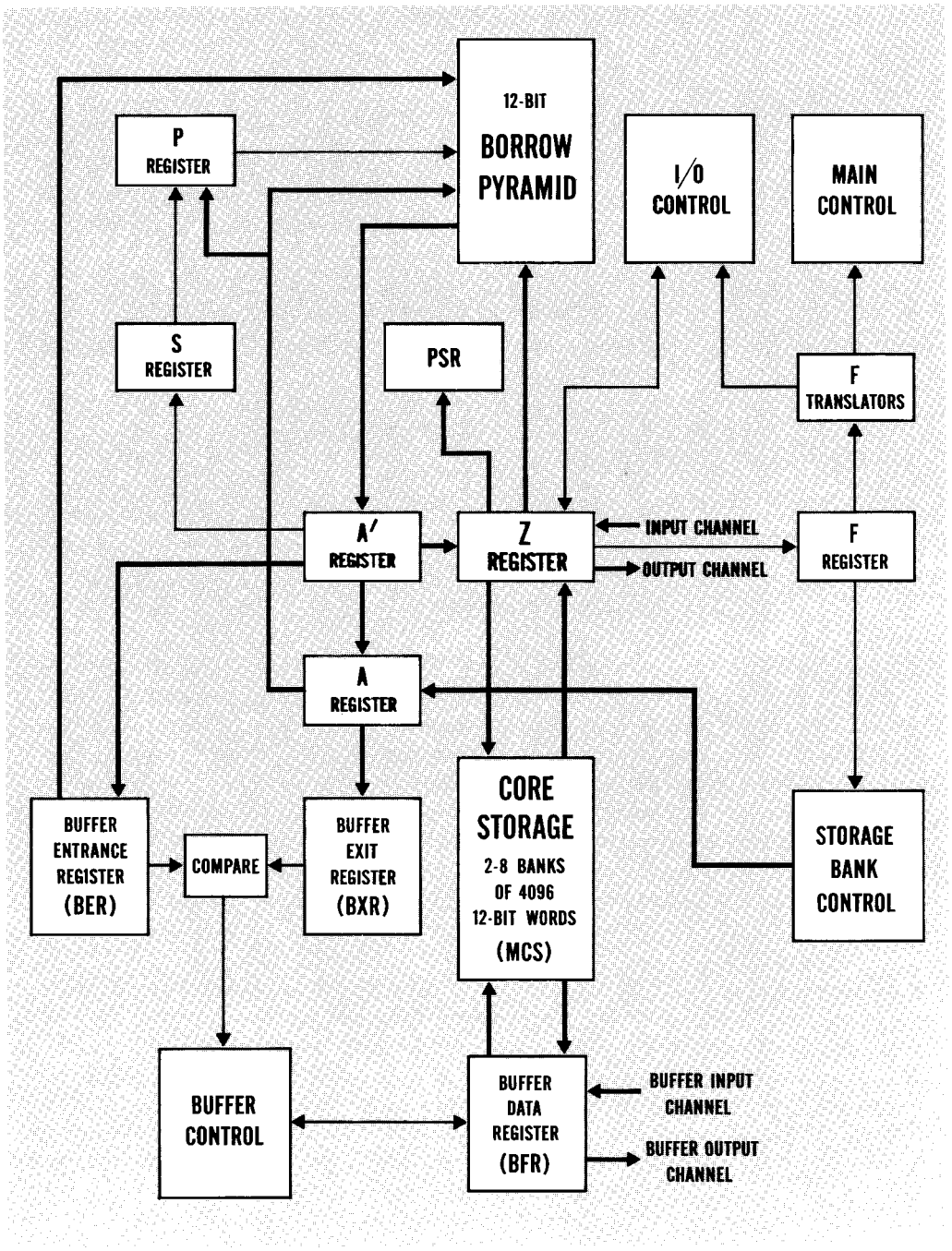


Figure 1. Block Diagram of Registers

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:

	+5 =	000	000	000	101
	+6 =	000	000	000	110
Complementing +6 produces		111	111	111	001
and subtraction takes place,					
	+5 =	000	000	000	101
	− −(+6)=	111	111	111	001
	Borrow (1)	000	000	001	100
Subtract the borrow	−				1
	produces	000	000	001	011 = 13_8, the corrected answer

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:

000 000 000 000 (plus zero)
111 111 111 111 (minus zero)

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:

$-0 + (-0)$ 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

Nine 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 the first 100 octal storage locations. The contents of this address are used as the address of the operand or as the jump address.

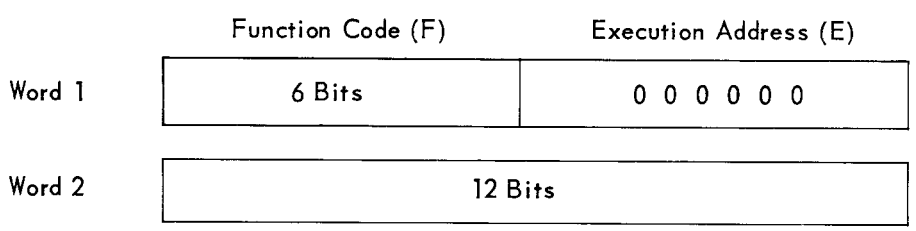
Relative Address Forward (F)	Operand or jump address is obtained by adding E to the current contents of P to specify one of the 63 (77 octal) addresses immediately following the address of the current instruction.
Relative Address Backward (B)	Operand or jump address is obtained by subtracting E from the current contents of P to specify one of the 63 (77 octal) addresses immediately preceding the address of the current instruction.
Forward Indirect Addressing (FI)	Certain instructions combine relative forward and indirect addressing. E, when added to P, produces a sum which specifies the location of the operand address or jump address. This indirect address cannot be further than 63 locations ahead of the current instruction. The operand or jump address may be located anywhere within storage. P remains unchanged.
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 nine 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 one of the first 100 octal locations in

* All numbers are in octal unless stated otherwise.

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 in the relative forward reference.
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.

FORWARD INDIRECT ADDRESS MODE (FI)

In the forward indirect mode, E is added to the contents of P. The sum then becomes the location of the indirect address. For Normal Input (INP) or Normal Output (OUT) instructions, this indirect address is the First Word Address (FWA) of the I/O storage block. For a Jump Forward Indirect (JFI) instruction, it is the address of the next instruction. The section on 160-A Instructions contains examples of the FI mode for each of the cases mentioned above.

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 10000_8 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 or address. 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):

Memory Location	Contents
(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.

LOCATION	CONTENTS			COMMENTS
	Symbolic F	E	Numeric	
(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.

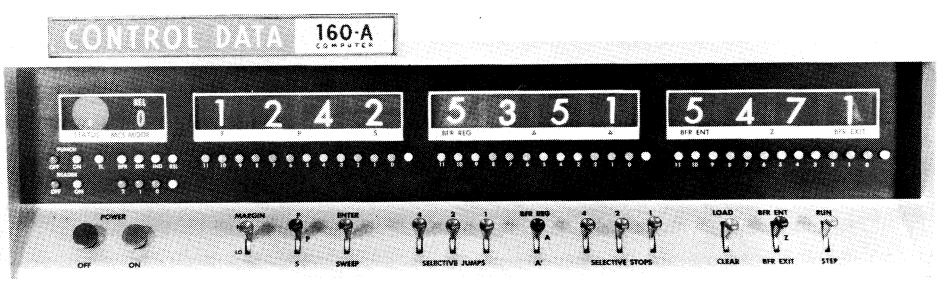


Figure 2. 160-A Console

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 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 reg sters 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 clea:

2) Turn on 350 reader by pressing the Reader On button, left side of the console.

3) Insert paper tape n 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.

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.

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 160-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.

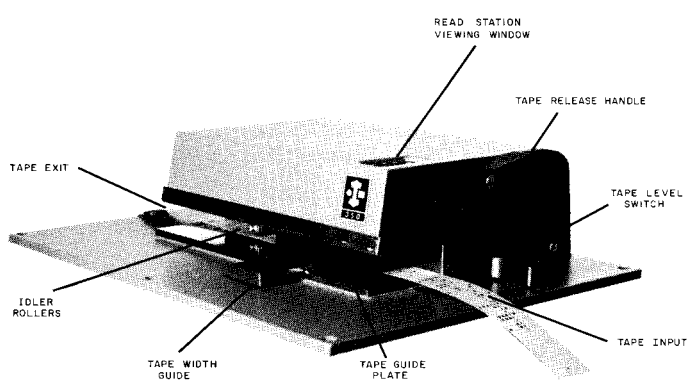


Figure 3. Paper Tape Reader

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.

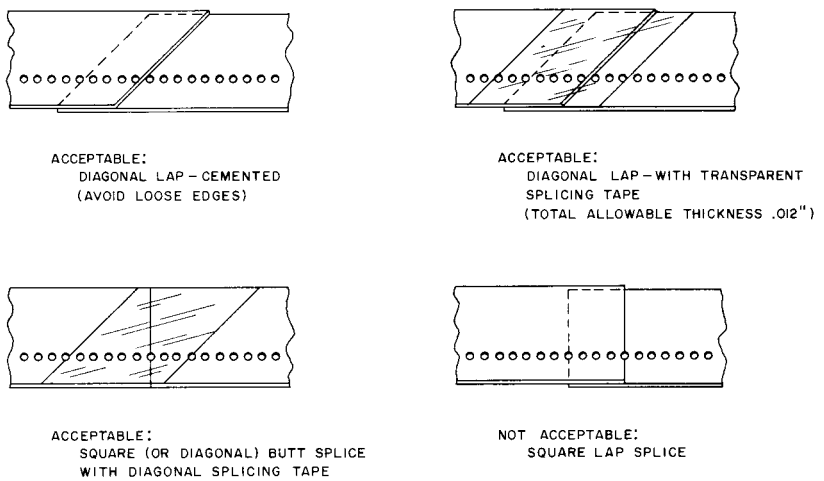


Figure 4. Tape Splices

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.

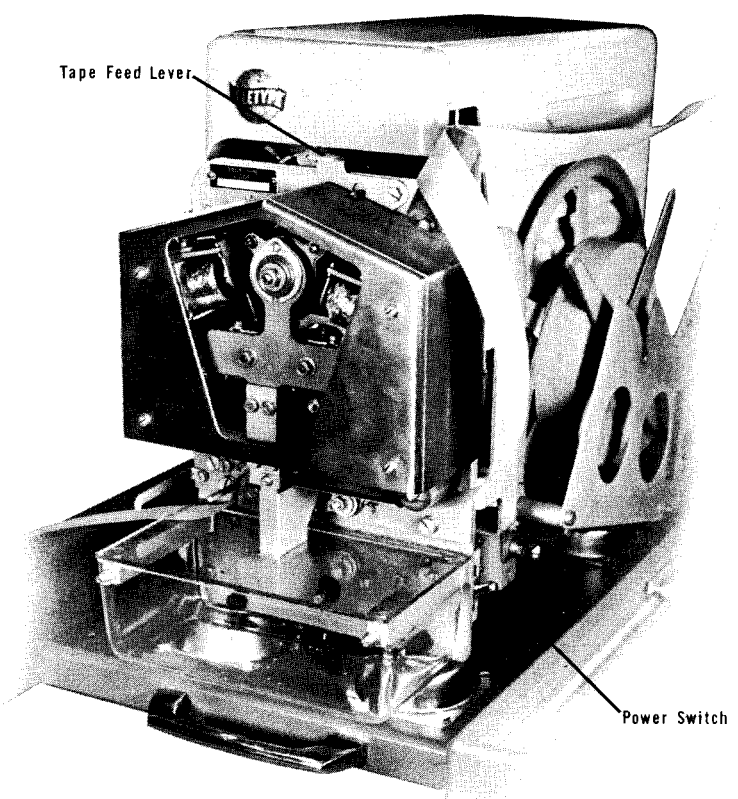


Figure 5. Paper Tape Punch

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.

F	E	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
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 within or between storage banks, and also between the internal registers.

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

Description:

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

If (BER) ≠ (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.

F	E	Mnemonic	Name	Timing
01	01	PTA	Transfer P to A	1

Description:

(P) → A
(P) + 1 → 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.

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

Description:

If the buffer is busy, YYYY → P
If the buffer is not busy, (A) → BER
(P) + 2 → 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.

F	E	G	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
01	30	CTA	Bank Controls to A	1

Description: (b) → A_{11-9}*
(d) → A_{8-6}
(i) → A_{5-3}
(r) → A_{2-0}
(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.

F	E	Mnemonic	Name	Timing
01	5X	STP	Store P at Location 5X	3

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

Transfer the contents of P to storage location (d) 005X, where X 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.

F	E	G	Mnemonic	Name	Timing
04	XX		LDN	Load No Address	1
20	XX		LDD	Load Direct	2
21	00	YYYY	LDM	Load Memory	3
21	XX		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.

F	E	Mnemonic	Name	Timing
01	6X	STE	Store Buffer Entrance Register at Location 6X and Transfer A to Buffer Entrance Register	3

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

Transfer the contents of BER to storage location (d) 006X, where X 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.

F	E	G	Mnemonic	Name	Timing
05	XX		LCN	Load Complement No Address	1
24	XX		LCD	Load Complement Direct	2
25	00	YYYY	LCM	Load Complement Memory	3
25	XX		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.

F	E	G	Mnemonic	Name	Timing
40	XX		STD	Store Direct	3
41	00	YYYY	STM	Store Memory	4
41	XX		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.

F	E	Mnemonic	Name	Timing
76	XX	HWI	Half Write Indirect	4

Description: $(A)_E \rightarrow$ operand address $_E$
$(P) + 1 \rightarrow P$

Transfer the E portion of A to the E portion of the operand address. A and the F portion of the operand address are 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 E portion of A will be transferred to the E portion 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.

F	E	Mnemonic	Name	Timing
01	12	MUT	Multiply A by 10	1

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

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.

F	E	Mnemonic	Name	Timing
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.

F	E	G	Mnemonic	Name	Timing
06	XX		ADN	Add No Address	1
30	XX		ADD	Add Direct	2
31	00	YYYY	ADM	Add Memory	3
31	XX		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) + operand → 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.

F	E	G	Mnemonic	Name	Timing
07	XX		SBN	Subtract No Address	1
34	XX		SBD	Subtract Direct	2
35	00	YYYY	SBM	Subtract Memory	3
35	XX		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) - operand → 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.

F	E	G	Mnemonic	Name	Timing
50	XX		RAD	Replace Add Direct	3
51	00	YYYY	RAM	Replace Add Memory	4
51	XX		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.

F	E	G	Mnemonic	Name	Timing
54	XX		AOD	Replace Add One Direct	3
55	00	YYYY	AOM	Replace Add One Memory	4
55	XX		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.

F	E	Mnemonic	Name	Timing
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.

F	E	G	Mnemonic	Name	Timing
44	XX		SRD	Shift Replace Direct	3
45	00	YYYY	SRM	Shift Replace Memory	4
45	XX		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.

F	E	G	Mnemonic	Name	Timing
02	XX		LPN	Logical Product No Address	1
10	XX		LPD	Logical Product Direct	2
11	00	YYYY	LPM	Logical Product Memory	3
11	XX		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)	0 1 0 1
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.

F	E	G	Mnemonic	Name	Timing
03	XX		SCN	Selective Complement No Address	1
14	XX		SCD	Selective Complement Direct	2
15	00	YYYY	SCM	Selective Complement Memory	3
15	XX		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.

F	E	Mnemonic	Name	Timing
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 Relative 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 sequencial 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.

F	E	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
70	XX	JPI	Jump Indirect	2

Description: ((d) 00XX) → P

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

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.

F	E	G	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
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.

F	E	Mnemonic	Name	Timing
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.)

F	E	G	Mnemonic	Name	Timing
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, YYYY → 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.

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

Description: If buffer is busy, YYYY → 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.

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

Description: (r)(P) + 00XX = the FWA of the input or output area
YYYY = 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 YYYY in the indirect memory bank (i)YYYY.

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.

F	E	Mnemonic	Name	Timing
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 on the normal input/output 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.

F	E	Mnemonic	Name	Timing
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 on the normal input/output 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.

F	E	G	Mnemonic	Name	Timing
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.

F	E	G	Mnemonic	Name	Timing
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) 2 (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 half instructions (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. 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.

Location	F	E	G	Comments
(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.

```
Location   F  E   Comments
(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_8), 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.)

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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.)

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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_8 words from location (i)0700 to location (i)2300 and perform this as a JPR subroutine whose entrance line is a.

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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.

Location	F	E	G	Comments
(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 fuction 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.

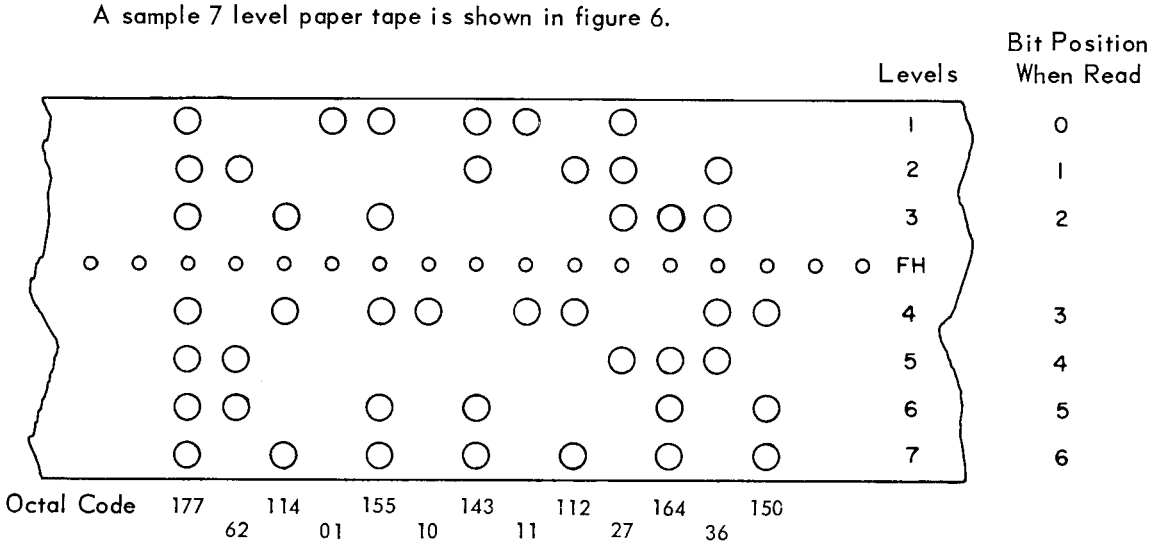


Figure 6. Punched Paper Tape Levels

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:

Location	F	E	Location	F	E
(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 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:

Location	F	E	Location	F	E
(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.

Location	F	E	G	Comments
(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 pyramid 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 FF.
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.
PYRAMID	A network of inverters that senses borrow conditions and produces borrow signals.
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	1) The condition of an external device, as reflected in the response given to a status request interrogation by the computer. 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.
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

F	E	MNE-MONIC	NAME	TIMING
00	00	ERR	Error Stop	1
00	0X	NOP	No Operation	1
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	00	BLS	Block Store (no jump) (jump)	1+n 2
01	01	PTA	Transfer P to A	1
01	02	LS1	Left Shift One	1
01	03	LS2	Left Shift Two	1
01	04	CBC	Clear Buffer Controls	1
01	05	ATE	A to Buffer Entrance Register (no jump) (jump)	1 2
01	06	ATX	A to Buffer Exit Register (no jump) (jump)	1 2
01	07	ETA	Buffer Entrance Register to A	1
01	10	LS3	Left Shift Three	1
01	11	LS6	Left Shift Six	1
01	12	MUT	Multiply A by Ten	1
01	13	MUH	Multiply A by One Hundred	1
01	14	RS1	Right Shift One	1
01	15	RS2	Right Shift Two	1
01	20	CIL	Clear Interrupt Lockout	1
01	30	CTA	Bank Controls to A	1
01	4X	SBU	Set Buffer Bank Control	1
01	5X	STP	Store P at Location 5X	3
01	6X	STE	Store Buffer Entrance Register at Location 6X & Transfer A to Buffer Entrance Register	3
02	XX	LPN	Logical Product No Address	1
03	XX	SCN	Selective Complement No Address	1
04	XX	LDN	Load No Address	1
05	XX	LCN	Load Complement No Address	1
06	XX	ADN	Add No Address	1
07	XX	SBN	Subtract No Address	1
10	XX	LPD	Logical Product Direct	2
11	00	LPM	Logical Product Memory	3
11	XX	LPI	Logical Product Indirect	3
12	00	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
14	XX	SCD	Selective Complement Direct	2
15	00	SCM	Selective Complement Memory	3
15	XX	SCI	Selective Complement Indirect	3
16	00	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
20	XX	LDD	Load Direct	2
21	00	LDM	Load Memory	3
21	XX	LDI	Load Indirect	3
22	00	LDC	Load Constant	2
22	XX	LDF	Load Forward	2
23	00	LDS	Load Specific	2
23	XX	LDB	Load Backward	2
24	XX	LCD	Load Complement Direct	2
25	00	LCM	Load Complement Memory	3
25	XX	LCI	Load Complement Indirect	3
26	00	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
30	XX	ADD	Add Direct	2
31	00	ADM	Add Memory	3
31	XX	ADI	Add Indirect	3
32	00	ADC	Add Constant	2
32	XX	ADF	Add Forward	2
33	00	ADS	Add Specific	2
33	XX	ADB	Add Backward	2
34	XX	SBD	Subtract Direct	2
35	00	SBM	Subtract Memory	3
35	XX	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	XX	STD	Store Direct	3

APPENDIX I

Table of Instructions (Cont'd)

F	E	MNE-MONIC	NAME	TIMING
41	00	STM	Store Memory	4
41	XX	STI	Store Indirect	4
42	00	STC	Store Constant	3
42	XX	STF	Store Forward	3
43	00	STS	Store Specific	3
43	XX	STB	Store Backward	3
44	XX	SRD	Shift Replace Direct	3
45	00	SRM	Shift Replace Memory	4
45	XX	SRI	Shift Replace Indirect	4
46	00	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
50	XX	RAD	Replace Add Direct	3
51	00	RAM	Replace Add Memory	4
51	XX	RAI	Replace Add Indirect	4
52	00	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
54	XX	AOD	Replace Add One Direct	3
55	00	AOM	Replace Add One Memory	4
55	XX	AOI	Replace Add One Indirect	4
56	00	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
60	XX	ZJF	Zero Jump Forward	1

F	E	MNE-MONIC	NAME		TIMING
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
70	XX	JPI	Jump Indirect		2
71	00	JPR	Return Jump		3
71	XX	JFI	Jump Forward Indirect		2
72	00	IBI	Initiate Buffer Input	(no jump) (jump)	1 2
72	XX	INP	Normal Input		*
73	00	IBO	Initiate Buffer Output	(no jump) (jump)	1 2
73	XX	OUT	Normal Output		*
74	XX	OTN	Output No Address		*
75	00	EXC	External Function Constant		2
75	XX	EXF	External Function Forward		2
76	00	INA	Input to A		*
76	XX	HWI	Half Write Indirect		4
76	77	OTA	Output from A		*
77	00	HLT	Halt		1
77	0X	SLS	Selective Stop		1
77	X0	SLJ	Selective Jump	(no jump) (jump)	1 2
77	XX	SJS	Selective Stop & Jump	(no jump) (jump)	2 2
77	77	HLT	Halt		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

F	E	G	MNE-MONIC	NAME	TIMING
STOP INSTRUCTIONS					
00	00		ERR	Error Stop	1
77	00		HLT	Halt	1
77	77		HLT	Halt	1
DATA TRANSMISSION INSTRUCTIONS					
01	00	YYYY	BLS	Block Store (no jump) (jump)	1 2
01	01		PTA	Transfer P to A	1
01	05	YYYY	ATE	A to Buffer Entrance Register (no jump) (jump)	1 2
01	06	YYYY	ATX	A to Buffer Exit Register (no jump) (jump)	1 2
01	07		ETA	Buffer Entrance Register to A	1
01	30		CTA	Bank Controls to A	1
01	5X		STP	Store P at Location 5X	3
01	6X		STE	Store Buffer Entrance Register at Location 6X & Transfer A to Buffer Entrance Register	3
04	XX		LDN	Load No Address	1
20	XX		LDD	Load Direct	2
21	00	YYYY	LDM	Load Memory	3
21	XX		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
05	XX		LCN	Load Complement No Address	1
24	XX		LCD	Load Complement Direct	2
25	00	YYYY	LCM	Load Complement Memory	3
25	XX		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
40	XX		STD	Store Direct	3
41	00	YYYY	STM	Store Memory	4
41	XX		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
76	XX		HWI	Half Write Indirect	4

F	E	G	MNE-MONIC	NAME	TIMING
ARITHMETIC INSTRUCTIONS					
01	12		MUT	Multiply A by 10	1
01	13		MUH	Multiply A by One Hundred	1
07	XX		SBN	Subtract No Address	1
34	XX		SBD	Subtract Direct	2
35	00	YYYY	SBM	Subtract Memory	3
35	XX		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
06	XX		ADN	Add No Address	1
30	XX		ADD	Add Direct	2
31	00	YYYY	ADM	Add Memory	3
31	XX		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
50	XX		RAD	Replace Add Direct	3
51	00	YYYY	RAM	Replace Add Memory	4
51	XX		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
54	XX		AOD	Replace Add One Direct	3
55	00	YYYY	AOM	Replace Add One Memory	4
55	XX		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
SHIFT INSTRUCTIONS					
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
44	XX		SRD	Shift Replace Direct	3
45	00	YYYY	SRM	Shift Replace Memory	4

APPENDIX I

Table of Instructions Arranged by Functions (Cont'd)

F	E	G	MNE-MONIC	NAME	TIMING
SHIFT INSTRUCTIONS (cont'd)					
45	XX		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
LOGICAL INSTRUCTIONS					
02	XX		LPN	Logical Product No Address	1
10	XX		LPD	Logical Product Direct	2
11	00	YYYY	LPM	Logical Product Memory	3
11	XX		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
03	XX		SCN	Selective Complement No Address	1
14	XX		SCD	Selective Complement Direct	2
15	00	YYYY	SCM	Selective Complement Memory	3
15	XX		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
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

F	E	G	MNE-MONIC	NAME	TIMING
JUMP INSTRUCTIONS					
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
70	XX		JPI	Jump Indirect	2
71	00	YYYY	JPR	Return Jump	3
71	XX		JFI	Jump Forward Indirect	2
INPUT/OUTPUT INSTRUCTIONS					
01	04		CBC	Clear Buffer Controls	1
01	20		CIL	Clear Interrupt Lockout	1
72	00	YYYY	IBI	Initiate Buffer Input (no jump) / (jump)	1 / 2
73	00	YYYY	IBO	Initiate Buffer Output (no jump) / (jump)	1 / 2
72	XX	YYYY	INP	Normal Input	*
73	XX	YYYY	OUT	Normal Output	*
74	XX		OTN	Output No Address	*
76	00		INA	Input to A	*
76	77		OTA	Output from A	*
75	00	XXXX	EXC	External Function Constant	2
75	XX		EXF	External Function Forward	2
SELECTIVE STOP AND JUMP INSTRUCTIONS					
00	0X		NOP	No Operation	1
77	0X		SLS	Selective Stop	1
77	X0	YYYY	SLJ	Selective Jump (no jump) / (jump)	1 / 2
77	XX	YYYY	SJS	Selective Stop & Jump (no jump) / (jump)	2 / 2

* 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
      111X Write tape X (6-bit) if OUT is given
      111X Write End-of-File mark if no OUT is given
      211X Write tape X (12-bit) if OUT is given
      112X Backspace tape X one record if INA is given
      112X Backspace tape X to End-of-File mark if no INA is given
      113X Read forward tape X (6-bit) if INPUT is given
      213X Read forward tape X (12-bit) if INPUT is given
      113X Search tape X for End-of-File mark if no INPUT is given
      114X Request tape X status (6-bit)
      115X Rewind unload tape X
      116X Rewind load tape X
      1171 Set tapes to odd parity
      1172 Set tapes to even parity (BCD)
      214X Request tape X status (12-bit)
      210X High density, tape X
      110X Low density, tape X

   B. Status Response Codes
      0000 Odd parity selected - no errors
      0001 Even parity selected - no errors
      0002 Tape X not ready
      0004 Horizontal and/or vertical parity error
      0015 Illegal BCD detected on Write
      0020 End-of-File mark read
      0040 End-of-Tape or Load Point sensed
      0100 Tape X high density
      0200 Tape X busy

   NOTE: The master bits 12, 13, 22, and 23 are used for second and third tape control. The "X" in the above codes can range from 0 to 7. 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 H→BCD and pack
      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	3314	Divide result
3301	Multiply	3315	Multiply result
3302	Left shift	3316	Left shift result
3303	Right shift	3317	Right shift result
3304	Status request		
3310	Reselect*		
3321	Addition	3335	Add to result
3342	Norm. & Count	3356	Norm. result & count
3361	Subtraction	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. 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

14. 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.

15. 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 $\overline{\text{Ready}}$ to read
- X1XX $\overline{\text{Ready}}$ to write
- XX4X Read parity error
- XX2X Write reply parity error
- XX1X End-of-File mark
- XXX4 End-of-Tape mark

16. 1617 CARD READER

A. External Function Codes

- 4500 EF clear
- 4501 Free run read
- 4502 Single cycle read
- 4504 H→BCD
- 4505 FRR, H→BCD
- 4506 SCR, H→BCD
- 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

Use This Page for Additional Codes

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 IV

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

APPENDIX IV

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

APPENDIX IV

OCTAL-DECIMAL INTEGER CONVERSION TABLE

4000 to 4777 (Octal)	2048 to 2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal)	2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

APPENDIX IV

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

Octal	Decimal
6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

Octal	Decimal
7000 to 7777 (Octal)	3584 to 4095 (Decimal)

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	.00109	.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 UC	CHARACTERS LC	CODE	CHARACTERS UC	CHARACTERS LC	CODE
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

CHAR	CODE	CHAR	CODE	CHAR	CODE	CHAR	CODE
Blank	20	F	66	V	25	≤	15
0	12	G	67	W	26	↑	16
1	01	H	70	X	27	⊏	17
2	02	I	71	Y	30	⊐	32
3	03	J	41	Z	31	→	35
4	04	K	42	.	73	≡	36
5	05	L	43	-	40	~, ∧	37
6	06	M	44	+	60	% or ∨	52
7	07	N	45	=	13	$ or ¬	53
8	10	O	46	(	34	↑	55
9	11	P	47	)	74	↓	56
A	61	Q	50	/	21	>	57
B	62	R	51	*	54	<	72
C	63	S	22	,	33	≥	75
D	64	T	23	:	00	?	76
E	65	U	24	≠	14	;	77

In last column, codes ~ % $ appear if business application, ∧ ∨ ¬ for scientific application.

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	○	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	66
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
			Upper Case (UC)		47
T	t	01	Lower Case (LC)		57
			Back Space (BS)		61
U	u	34	Color Shift (CS)		02
			Tabulate (TAB)		51
V	v	17	Stop		43
			Space		04
W	w	31	Tape Feed		00
			Delete		77
X	x	27			

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	60	—	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

INDEX

CONTROL DATA

CORPORATION

501 PARK AVENUE, MINNEAPOLIS 15, MINNESOTA • FEDERAL 9-0411