Carole Griffit

PROGRAMMERS' REFERENCE MANUAL

S-C 4060
PRODUCT CONTROL UNIT

June 1967

9500286

Stromberg-Carlson
A SUBSIDIARY OF GENERAL DYNAMICS CORPORATION

DATA PRODUCTS DIVISION

1895 HANCOCK STREET, POST OFFICE BOX 2449, SAN DIEGO, CALIFORNIA 92112 TELEPHONE: SAN DIEGO (714) 298-8331

PROGRAMMERS' REFERENCE MANUAL

S-C 4060 PRODUCT CONTROL UNIT

9500286

June 1967

INTRODUCTION

The Stromberg-Carlson S-C 4060 Product Control Unit (PCU), is an integrated circuit 16-bit binary word programmable buffer with a 1 μsec cycle time magnetic core memory. The PCU has a fully parallel machine organization and both indexing and multilevel indirect addressing. Memory size is 8192 words. Standard features include a flexible instruction repertoire of 72 commands, a hardware index register, a powerful I/O bus structure, and standard Teleprinter keyboard and paper tape I/O unit. An extensive programming package, including a symbolic assembler, ASA FORTRAN IV and diagnostic and utility routines, is provided with the basic unit.

The 16-bit word of the PCU allows a straightforward and efficient addressing scheme. Most internal operations can be performed in two cycle times (2 μsec), or less including instruction access and execution time. A single word instruction can directly address any one of 1024 words. The 16-bit word is directly compatible with the ASCII 8-bit character code.

TABLE OF CONTENTS

Section	Title	Page
I	PCU ORGANIZATION	1-1
	Specifications	1-1
	System Description	1-2
	Word Formats	1-4
	Data Formats	1-4
	Memory Addressing	1-8
	Direct Addressing	1-8
	Indexing	1-9
	Indirect Addressing	1-9
	Locations $(00001)_8$ to $(00017)_8$	1-10
	Memory Reference Instruction Logic and Timing	1-10
II	STANDARD INSTRUCTIONS	2-1
III	INPUT/OUTPUT	3-1
	Input/Output Control and Communication	3-1
	Single-Word Transfer Mode	3-3
	DMC Mode	3-5
	Standard Interrupt	3-7
	Power Failure Interrupt (PFI)	3-9
IV	MAGNETIC TAPE OPERATION	4-1
	Functional Characteristics	4-1
	TCU Interrupts	4-4
	Tape Function Code	4-4
	Basic Tape Operations	4-8

TABLE OF CONTENTS (cont)

Section	Title	Page
V	ASR-33 TELETYPE	5-1
	Keyboard and Carriage Features	5-1
	Operating Modes - ASR-33 Unit	5-2
	Tape Reader	5-2
	Tape Punch	5-2
	Off-Line Operation	5-2
	ASR-33 On-Line Operating Modes	5-3
	Character Modes	5-3
	Instructions	5-4
	Standard Interrupt	5-6
	Paper Tape Format and ASR Codes	5-6
VI	S-C 4060 GRAPHICS INSTRUCTIONS	6-1
	Print Head Word Formats	6-1
	Plot Specified Point	6-1
	Fast Plot	6-2
	Draw Vector	6-2
	Stroke Write (Optional)	6-4
	Special Functions	6-6
VII	PRINT HEAD OPERATIONS	7-1
	Functional Characteristics	7-1
	Basic Print Head Instructions	7-1
	Print Head Status Word	7-2
VIII	SAMPLE PROGRAMS	8-1
	Fixed Point, Double Precision Add Subroutine	8-1
	Fixed Point, Double Precision Subtract Subroutine	8-2
	Fixed Point, Single Precision Multiply Subroutine	8-3
	Fixed Point, Single Precision Divide Subroutine	8-4
	Output on ASR-33	8-7

TABLE OF CONTENTS (cont)

		Page
APPENDIX A	CRT PROGRAMMING SPECIFICATIONS. . . .	A-1
APPENDIX B	ASCII CODE	B-1
APPENDIX C	SUMMARY OF STANDARD INSTRUCTIONS . . . (Listed in Alphabetical Order)	C-1
APPENDIX D	DEDICATED LOCATIONS	D-1

ILLUSTRATIONS

Figure	Title	Page
1-1	PCU Simplified Block Diagram	1-4
1-2	Data Word Format, Single Precision	1-5
1-3	Data Word Format, Double Precision	1-5
1-4	Memory Reference Instruction Format	1-6
1-5	Input/Output Instruction Format	1-6
1-6	Shift Instruction Format	1-7
1-7	Generic Instruction Format	1-7
1-8	Memory Sectors for the First 4096 Words	1-8
1-9	Indirect Address Format	1-9
1-10	Fetch, Indexing and Indirect Addressing, Logic Flow Diagram	1-11
5-1	ASR-33 Paper Tape Format	5-8
A-1	CRT Orientation	A-1

TABLE OF CONTENTS (cont)

TABLES

Table	Title	Page
2-1	Glossary of Symbols	2-2
2-2	Instruction Repertoire	2-3
3-1	Input/Output Bus Lines	3-1
3-2	DMC Start and Terminal Memory Address Locations	3-6
3-3	Standard Interrupt Mask Assignments	3-8
4-1	Bit Significance in Read or Write Words	4-3
4-2	Interrupts During Tape Operations	4-5
4-3	Tape Function Code	4-6
4-4	Bit Significance in the Sense Word	4-10
5-1	ASR-33 Characters and Symbol Codes	5-7
6-1	Summary of Print Head Commands	6-7
7-1	Print Head Status Word	7-5
A-1	Character Conversion Codes	A-2
A-2	Character and Line Spacing	A-5
A-3	Standard Dashes in Raster Units	A-5
A-4	Standard Line Widths in Raster Units	A-6

SECTION I

PCU ORGANIZATION

SPECIFICATIONS

Type

Parallel Binary

Addressing

Single address with indexing and indirect addressing

Word Length

16 Bits

Machine Code

Two's complement

Memory Type

Magnetic Core

Memory Size

8, 192

Memory Cycle Time

1 μsec

Speed

Add: 2 μsec
Subtract: 2 μsec

Multiply
(hardware): 5.5 μsec max

Divide
(hardware): 11.0 μsec max

Standard Peripheral Equipment

ASR-33 Teletype Unit providing the following capabilities:

a. Read paper tape at 10 cps
b. Punch paper tape at 10 cps
c. Type at 10 cps
d. Keyboard input
e. Off-line paper-tape preparation, reproduction and listing

Optional Peripheral Equipment

300 cps photoelectric paper-tape reader
110 cps paper-tape punch
200 card-per-minute card reader

Standard Input/Output Lines

16-bit input bus
16-bit output bus
10-bit device address bus
External control and sense lines

Input/Output Modes

Two modes are available for data transfer between peripheral devices and the PCU.

a. Single word transfer with or without interrupt
b. Direct multiplex control (DMC)

Interrupt

Single interrupt line standard.

Power Failure Protection

Power failure interrupt standard. Core memory protected against loss of information on ac power failure.

SYSTEM DESCRIPTION

Figure 1-1, a block diagram of the PCU, shows the data storage registers, the control unit of the central processor and the input/output controls. The random access memory, shown as a single block, is a magnetic core unit containing 8192 16-bit words. Data from the memory is transferred to and from the PCU registers through the M-register. The functional units of the central processor and the input/output controls are as follows:

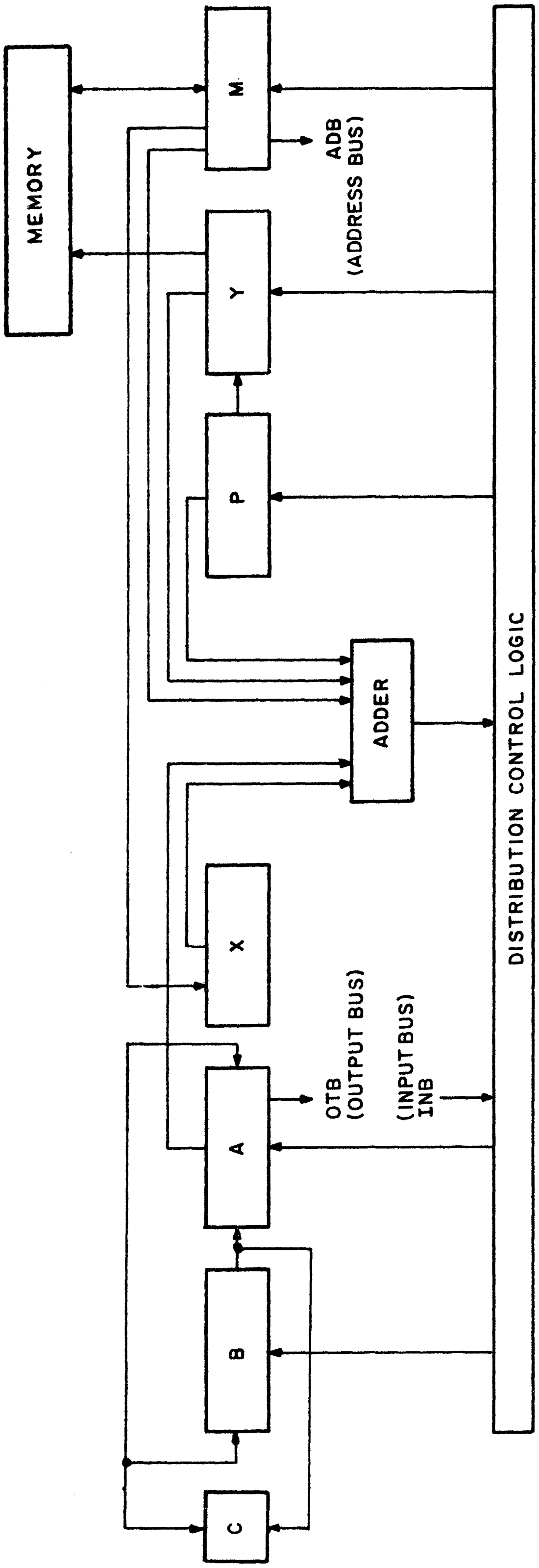


Figure 1-1. PCU Simplified Block Diagram

A-Register (A): A 16-bit register used as the primary arithmetic and logic register of the computer.

B-Register (B): A 16-bit secondary arithmetic register used primarily to hold arithmetic operands which exceed one word in length.

Program Counter (P): A 16-bit register that contains the location of the next instruction to be executed.

Adder: Performs the basic arithmetic processes of addition and subtraction.

M-Register (M): A 16-bit register used to transfer information to and from the magnetic core memory.

Y-Register(Y): A 16-bit register used to store the address for the memory.

C-Bit (C): A 1-bit indicator associated with the A- and B-registers, which stores overflow status resulting from the execution of arithmetic instructions, and stores the last bit shifted out of the A- or B-register during the execution of shift instructions.

Index Register (X): A 16-bit register used for address modification. Any memory write cycle addressing memory location zero also loads the X-register.

Output Bus (OTB): Sixteen lines that transmit data from the PCU A-register to an I/O device.

Input Bus (INB): Sixteen lines that transmit data from an I/O device to the PCU A-register.

Address Bus (ADB): Ten lines used in conjunction with I/O devices. Bits on lines 7 through 10 define the function to be performed by the I/O device. Bits on lines 11 through 16 designate the I/O device to be used.

WORD FORMATS

Data Formats

Single Precision. -- The format for data words stored in memory is shown in Figure 1-2.

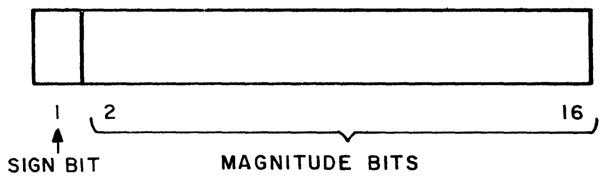


Figure 1-2. Data Word Format, Single Precision

Sixteen-bit data words are stored in two's complement form. The first bit of a data word may be considered the arithmetic sign and is zero for positive data.

Double Precision. -- When greater precision is required than that obtainable when using the single precision format, the double precision format is used (Figure 1-3). The sign position of the second (least significant) word is always zero. Thirty bits of magnitude are obtainable. This is the format for the product of the multiplication of two single precision words. It is also the data format for double precision operations.

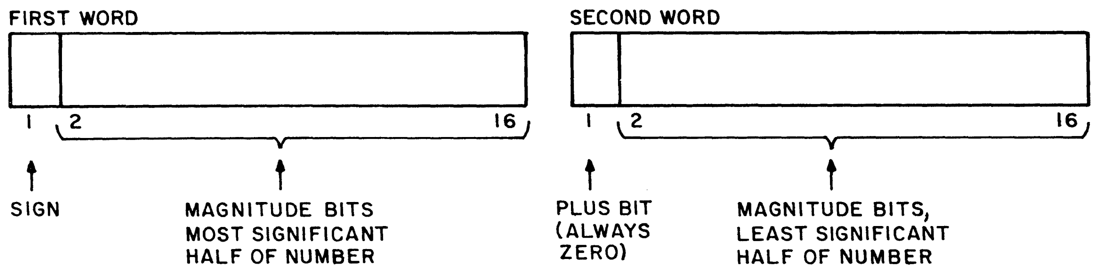

FIRST WORD
SECOND WORD
1
2
16
1
2
16
SIGN
MAGNITUDE BITS MOST SIGNIFICANT HALF OF NUMBER
PLUS BIT (ALWAYS ZERO)
MAGNITUDE BITS, LEAST SIGNIFICANT HALF OF NUMBER

Figure 1-3. Data Word Format, Double Precision

Logical Data. -- Logical data, such as the condition of sixteen binary indicators, can be stored in a single data word. This type of data is generally not treated arithmetically by the program but logically by means of Boolean operators such as "AND" and "exclusive OR." In this case, bit 1 of a word does not represent the sign but the first of sixteen conditions.

Instruction Words. -- Instruction words are divided into four types: memory reference, input-output, shift, and generic.

The basic instruction word format in the PCU is that for a memory reference instruction, which is shown in Figure 1-4. Bits 3 to 6 contain the operation code, which defines the function to be performed. For example, if bits 3 to 6 contain 0110 $(06)_8$ the instruction is identified as an add instruction; if they contain 1001 $(11)_8$ the instruction is a compare. For ease of communication, operation codes are generally expressed either in octal or as a mnemonic. "Subtract," for example, which has an op-code bit configuration of 0111, is referenced in machine language as $(07)_8$ and has a mnemonic of SUB. The latter is the way the programmer writes an op code when programming in the PCU assembly language.

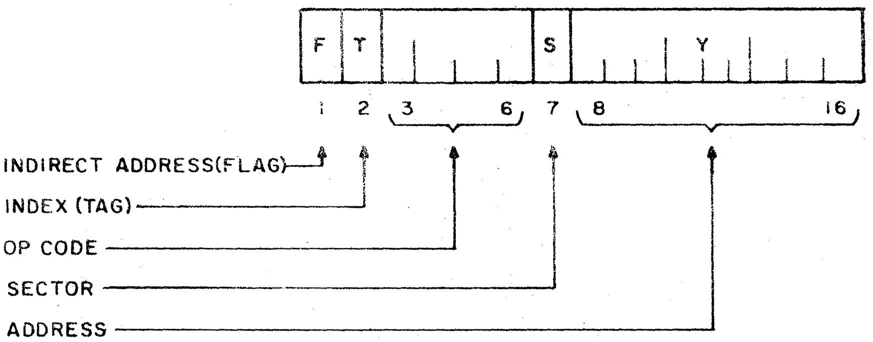


Figure 1-4. Memory Reference Instruction Format

The input/output instruction word format is shown in Figure 1-5. Bits 1 through 6 specify the particular I/O instruction; bits 11 through 16 specify which device is being addressed. Bits 7 through 10 define the function to be performed by the instruction.

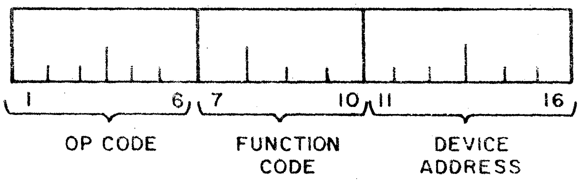


Figure 1-5. Input/Output Instruction Format

The shift instruction word format is shown in Figure 1-6. Bits 1 through 10 specify the type of shift; and bits 11 through 16 are used to define the number of shifts to be performed. The number of shifts must be represented in two's complement form.

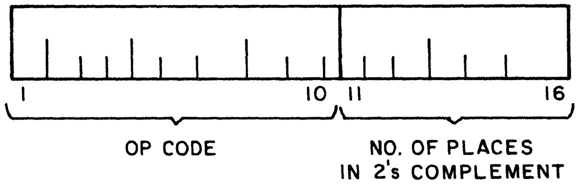


Figure 1-6. Shift Instruction Format

The generic instruction word format is shown in Figure 1-7. All 16 bits are used to specify the instruction.

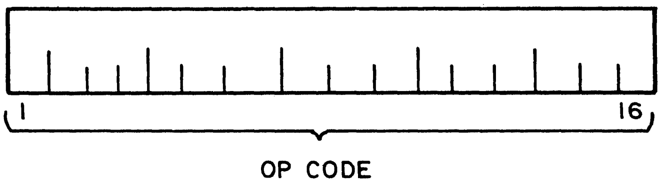


Figure 1-7. Generic Instruction Format

The op code expressed in binary, octal, and mnemonic for representative instructions of each of the four types, are listed in the following:

			Operation Code	
Instruction	Type	Binary	Octal	Mnemonic
Subtract	Memory Reference	x x 0 111 x xxx xxx xxx	07	SUB
Input to A	Input/Output	101 100 x xxx xxx xxx	54	INA
Arithmetic Left Shift	Shift	0 100 001 101 xxx xxx	0415	ALS
Clear A	Generic	1 100 000 000 100 000	140040	CRA

MEMORY ADDRESSING

Several techniques are used in the PCU for memory addressing: direct addressing, indexing and indirect addressing.

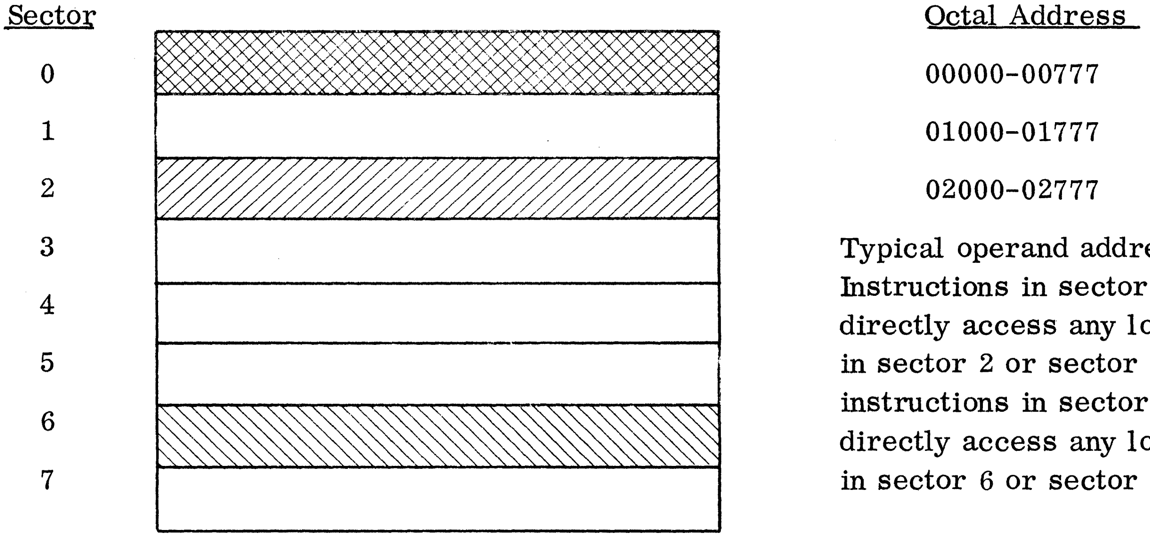
Direct Addressing

The memory of the PCU is considered to be divided into sectors of 512 words each making a total of 16 sectors. Any word in a sector can be addressed with 9 bits ($2^9 = 512$). The address portion of a memory reference instruction (bits 8 to 16) can thus define a unique word in a sector. Addresses within sectors run from $(000)_8$ to $(777)_8$. The sector bit, bit 7 of the instruction, identifies the sector of the word addressed in accordance with the following rules:

Sector Bit = 0 The address is in sector 0 (octal address 00000 - 00777).
Sector Bit = 1 The address is in the same sector as the instruction being executed.

For example, assume an ADD 444 instruction is in address $(02100)_8$, or sector 2 word 100. If the sector bit in the instruction is 0, the instruction references word 444 in sector 0, or $(00444)_8$. If the sector bit is 1, then the instruction references word 444 in sector 2, or $(02444)_8$, because the instruction itself is in sector 2.

A single instruction can thus directly address 1024 words, half of which are in sector 0 and half of which are determined by the location of the instruction. Figure 1-8 represents the memory that can be directly addressed by an instruction in sector 2 and an instruction in sector 6.



Typical operand addressing: Instructions in sector 2 can directly access any location in sector 2 or sector 0; instructions in sector 6 can directly access any location in sector 6 or sector 0.

Figure 1-8. Memory Sectors for the First 4096 words

Indexing

Further addressing flexibility is implemented through the use of indexing. The index register is a 16-bit hardware register whose contents can be added to the direct address of an instruction to produce a new effective operand address. Indexing is specified by putting a one in bit 2 of a memory reference instruction. In assembly language, indexing is specified by placing a comma followed by a one after the operand (that is, ADD B, 1).

The value in the index register can be positive or negative. If negative, the effective address is less than the base (instruction) address. The latter is the usual means of utilizing the index register when controlling program loops, each time through the loop the negative content of the index register is incremented by one by means of an IRS (increment, replace and skip) instruction. When the index register reaches zero, the program automatically breaks out of the loop.

Indirect Addressing

If bit 1 of a memory reference instruction is set, indirect addressing takes place. When indirect addressing is specified, the effective address of the operand is assumed to be in the location specified by the address portion of the instruction and the sector bit. The format of the indirect address location is shown in Figure 1-9.

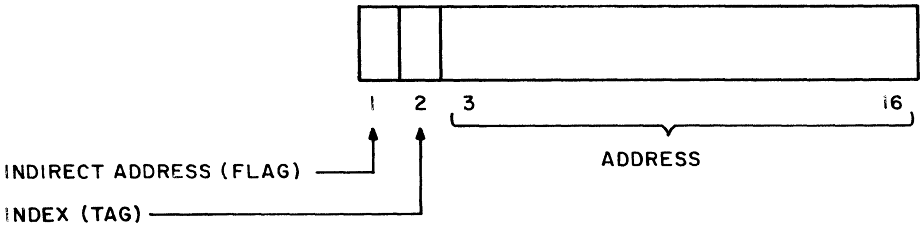


Figure 1-9. Indirect Address Format

To illustrate indirect addressing, consider that an add command in sector 2 is flagged for indirect addressing (this is specified in the assembly language format by placing an asterisk after the op code).

ADD* 444

Location 444 contains

$(06231)_8$

The effective address would then be $(06231)_8$, which is in sector 6. The content of location 06231 would be added to the A-register.

If the indirect bit within an indirect address location is set, a further level of indirect addressing takes place. This chaining of indirect addresses can continue indefinitely.

If both the indirect bit and the index bit are set in an instruction, indexing takes place first. To have indexing occur after indirect addressing (that is, the effective address is the sum of the indirect address and the contents of the index register), the index bit is set in the indirect location.

Locations $(00001)_8$ to $(00017)_8$

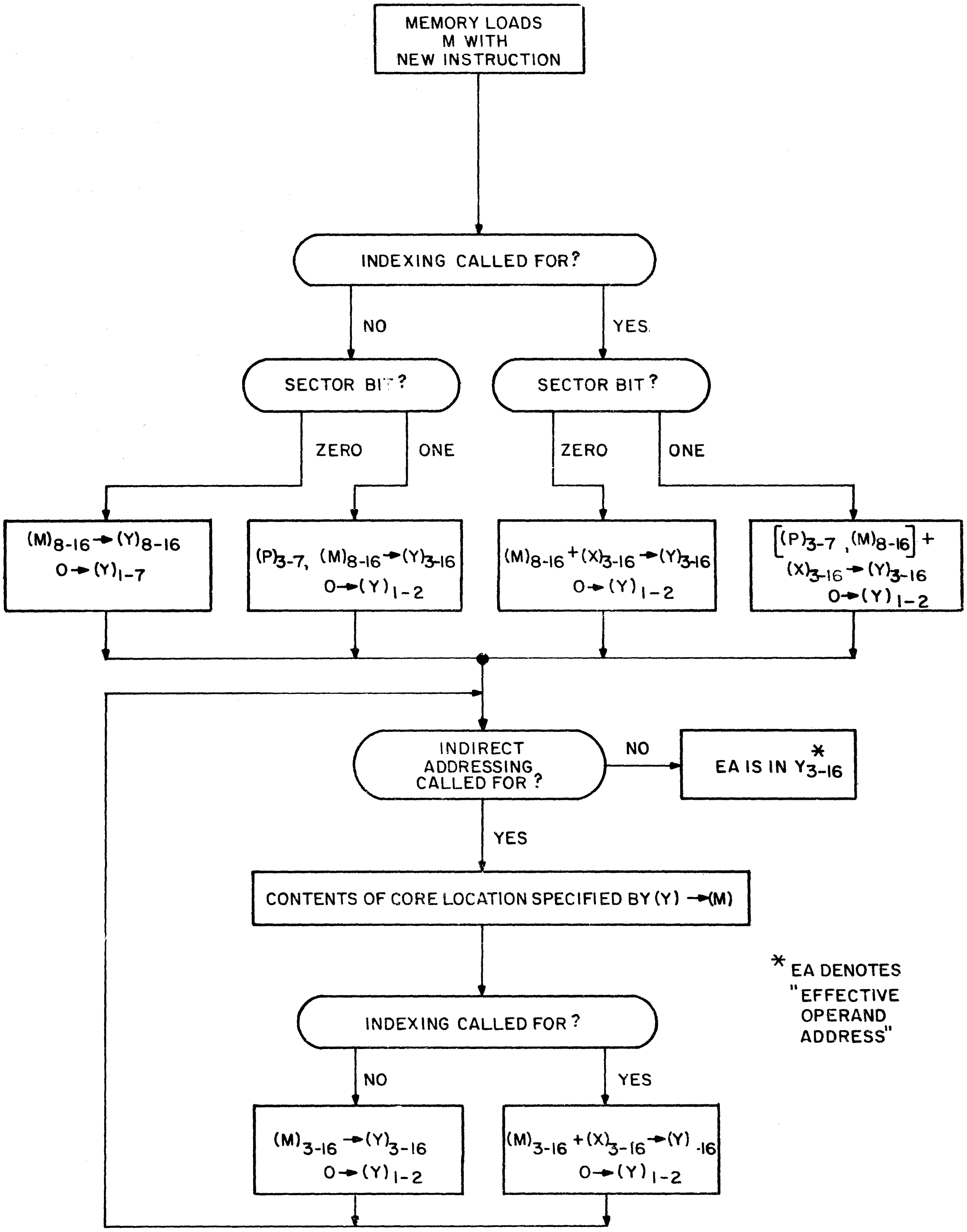
Memory locations $(00001)_8$ through $(00017)_8$ are protected in the PCU against being written into under program control. Information may be read from these locations in the normal manner, however, all instructions which attempt to write in them will be aborted. The only way in which these locations may be loaded is through the use of the memory access feature of the console. The locations provide protected storage for the Key-In Loader utilized with the software system.

MEMORY REFERENCE INSTRUCTION LOGIC AND TIMING

Figure 1-10 is a logic flow diagram of the fetch, indexing and indirect addressing phases of an instruction. Initially, the P-register (program counter) contains the address of the instruction to be executed. The Y-register (memory address) also contains the same address. The instruction in the address specified by Y is then read out of memory into the M-register (memory information), and the operation code stored in the op-code register.

The index bit is first examined. If the index bit is a ZERO, no indexing is called for and the sector bit is examined. If the sector bit is set, the seven most significant bits of the program counter (the sector the instruction came from) and the 9 least significant bits of M (the address portion of the instruction) are transferred to Y. If the sector bit is ZERO, ZEROs are placed in the seven most significant bits of Y (thus addressing sector zero). If the indirect bit is not set, no indirect addressing is required and the contents of Y represent the effective operand address of the instruction. The PCU then proceeds to the execution phase of the instruction.

If the index bit is a one, indexing is required. The value which would otherwise have been put in the Y register as a result of the examination of the sector bit as described above is first added to the contents of the index register (X) and the sum placed in the Y register. If the indirect bit is not set, the contents of Y represent the effective operand address and the instruction proceeds to the execution phase.



* EA DENOTES "EFFECTIVE OPERAND ADDRESS"

Figure 1-10. Fetch, Indexing and Indirect Addressing, Logic Flow Diagram

If, when the indirect bit is examined, it is a ONE, indirect addressing is required. The contents of Y formed as a result of examination of the sector bit and indexing, if called for, is then treated as the address of an indirect address word in memory rather than the effective operand address. The indirect address is then read out of memory (into the M-register) and its 14 least significant bits placed in the Y register unless the index bit in the indirect address word is set. If the index bit is set, the contents of the index register is added to the 14 least significant bits in M and the result placed in Y.

If the indirect bit in the indirect address word is a ZERO, the contents of Y represent the effective operand address of the instruction. If the indirect bit is a ONE, Y represents the address of another indirect address word which is read out of memory and processed in the same manner as the first. There is no basic limit to the number of indirect words which can be called for before the generation of the effective operand address.

SECTION II

STANDARD INSTRUCTIONS

INSTRUCTION REPERTOIRE

The instructions which comprise the PCU Instruction Repertoire are described in detail in this section. Mnemonics and symbols used in the instruction descriptions are listed in Table 2-1. A thorough knowledge of the data presented in Table 2-1 is necessary to understand the instruction descriptions.

Table 2-2 lists all standard instructions. Each instruction is identified by its assigned three-letter mnemonic, type symbol, and octal Op-Code. Definitions, descriptions, and timing data for each instruction are also included in Table 2-2. Refer to Section I for instruction word formats.

The standard instructions in Table 2-2 are grouped into the following operational categories:

Load and Store
Arithmetic
Logical
Shift
Input/Output
Control
Half-Word

Arithmetic instructions which provide overflow detection are indicated by the designation Overflow Status → (A). If overflow occurs on a particular instruction, the C-bit is set to a one If overflow does not occur, the C-bit is reset to a zero. Thus, after each arithmetic instruction, the contents of the C-bit indicates whether overflow occurred on that instruction.

Instructions which reference double-precision operands must produce even effective addresses (after all indirection and indexing). An odd effective address will cause the instruction to be executed as if it had the next lower even effective address in the case of double load, add, or subtract. An odd effective address in a double-precision store will cause the B-register content to be stored in the specified location without affecting any other register location.

Table 2-1. Glossary of Symbols

Symbol	Definition
EA	Effective operand address; the address from which the operand will be obtained. This is determined only after all selection of sectors, indexing, and indirect addressing required have been performed.
n	Specified number of shifts to be performed.
N	Two's complement of the number of shifts to be performed.
ADB	Address Bus
INB	Input Bus
OTB	Output Bus
DP Mode	Double Precision Mode
A	A-Register (16-bits)
P	Program Counter (16-bits) -
B	B-Register (16-bits)
E	E-Register (16-bits)
X	Index Register (16-bits)
M	M-Register (16 bits)
C	C-bit (1 bit)
→	Replaces
⇄	Is exchanged with
↴	Is discarded
∧	Logical AND
∨	Logical OR
⊻	Exclusive OR
+	Algebraic Addition
( )	Contents of a hardware register (e. g., (A) = contents of A-Register)
[ ]	Contents of core location specified by (e. g. [EA] = contents of core location specified by EA)
T	Tag Bit (bit 2 of instruction word)
MR	Memory Reference Instruction
G	Generic Instruction
SH	Shift Instruction
IO	Input-Output Instruction

Table 2-2. Instruction Repertoire

Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
Load and Store					
CRA	G	140040	Clear A	$O \rightarrow (A)$	1
IAB	G	000201	Interchange A and B	$(A) \leftrightarrows (B)$	1
IMA	MR	13	Interchange Memory and A	$(A) \leftrightarrows [EA]$	3
INK	G	000043	Input Keys	$(C) \rightarrow (A)_1$ $(DP\ Mode) \rightarrow (A)_2$ $(PMI) \rightarrow (A)_3$ $O \rightarrow (A)_{4-11}$ $Shift\ Count \rightarrow (A)_{12-16}$	1
LDA	MR	02	Load A	$[EA] \rightarrow (A)$	2
LDX	MR	15 T = 1	Load X	$[EA] \rightarrow (X)$ $[EA] \rightarrow [00000]$	3
			NOTE This instruction cannot be indexed. However, if indirect addressing is called for, the indirect address can be indexed in the usual manner.		
OTK	G	171020	Output Keys	$(A)_1 \rightarrow (C)$ $(A)_2 \rightarrow (DP\ Mode)$ $(A)_3 \rightarrow (PMI)$ $(A)_{12-16} \rightarrow Shift\ Count$	2
STA	MR	04	Store A	$(A) \rightarrow [EA]$	2
STX	MR	15 T = 0	Store X	$(X) \rightarrow [EA]$	2
			NOTE This instruction cannot be indexed. However, if indirect addressing is called for, the indirect address can be indexed in the usual manner.		
Arithmetic					
ACA	G	141216	Add C to A	$(A) + (C) \rightarrow (A)$ Overflow status → (C)	1
ADD	MR	06	Add	$(A) + [EA] \rightarrow (A)$ Overflow status →(C)	2
AOA	G	141206	Add One to A	$(A) + 1 \rightarrow (A)$ Overflow status → (C)	1

Table 2-2. Instruction Repertoire (cont)

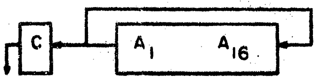
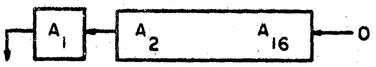
Mnemonic	Type	Op Code	Definition	Description	Time (µsec)
SUB	MR	07	Subtract	$(A) - [EA] \rightarrow (A)$ Overflow status → C	2
TCA	G	141407	Two's Complement A	$-(A) \rightarrow (A)$	1.5
Logical					
ANA	MR	03	AND to A	$(A) \wedge [EA] \rightarrow (A)$ EXAMPLE: (A) 0 1 0 1 [EA] 0 0 1 1 RESULT IN A 0 0 0 1	2
CSA	G	140320	Copy Sign and Set Sign Plus	$(A)_1 \rightarrow (C)$ $0 \rightarrow (A)_1$	1
CHS	G	140024	Complement A Sign	$\overline{(A)}_1 \rightarrow (A)_1$	1
CMA	G	140401	Complement A	$\overline{(A)} \rightarrow (A)$	1
ERA	MR	05	Exclusive OR to A	$(A) \veebar [EA] \rightarrow (A)$ EXAMPLE: (A) 0 0 1 1 [EA] 0 1 0 1 RESULT IN A 0 1 1 0	2
SSM	G	140500	Set Sign Minus	$1 \rightarrow (A)_1$	1
SSP	G	140100	Set Sign Plus	$0 \rightarrow (A)_1$	1
Shift					
ALR	SH	0416N	Logical Left Rotate	 The A register is shifted left, end-around (n) positions. A_1 is shifted out to A_{16} and the C bit. The C bit takes the state of the last bit shifted into A_{16}.	1 + n/2
ALS	SH	0415N	Arithmetic Left Shift	 Overflow status → (C) The A register is shifted left (n) positions. If shifting causes a change in the sign of A at any time during the instruction, the C bit is set. If the sign is not changed, the C bit is reset. After 16 or more shifts, the A register contains ZERO.	1 + n/2

Table 2-2. **Instruction Repertoire (cont)**

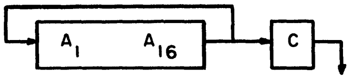
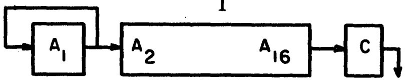
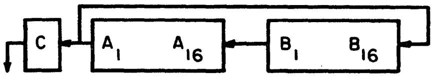
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
ARR	SH	0406N	Logical Right Rotate	 The A register is shifted right, end around (n) positions. Bits shifted out of A_{16} enter A_1 and the C bit. The C bit takes the state of the last bit shifted into A_1.	1 + n/2 1 + n/2
ARS	SH	0405N	Arithmetic Right Shift	 The A register is shifted right (n) positions. The sign bit (A_1) does not change; it is shifted into vacated positions of the register. Bits shifted out of A_{16} enter the C bit. The C bit takes the state of the last bit shifted out of the register. If 15 or more shifts are specified, all stages of the A register will be the same as the sign bit.	1 + n/2
LLR	SH	0412N	Long Left Rotate	 The A and B registers are treated as a single 32-bit register and shifted left, end around, (n) positions. Bits shifted out of B_1 enter A_{16}; bits shifted out of A_1 enter B_{16} and the C bit. Bits shifted out of C bit are discarded. The C bit takes the state of the last bit shifted into B_{16}.	1 + n/2

Table 2-2. Instruction Repertoire (cont)

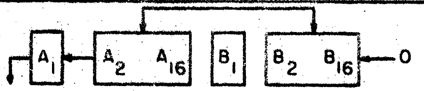
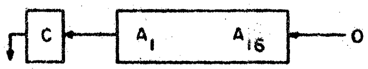
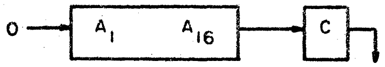
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
LLS	SH	0411N	Long Arithmetic Left Shift	 A 1 A 2 A 16 B 1 B 2 B 16 0 Overflow Status → (C) The A and B registers are treated as a single 31-bit register (B_1 is not changed) and shifted left n positions. ZEROs are shifted into vacated positions through B_{16}. Bits shifted out of B_2 enter A_{16}. If at any time during the instruction the sign of the A register $(A)_1$ is changed, the C bit is set. If at the end of the instruction the sign has not been changed, the C bit is reset. If 31 or more shifts are specified, the A and B registers contain ZERO (except for B_1, which is unchanged).	1 + n/2
LGL	SH	0414N	Logical Left Shift	 The A register is shifted left (n) positions. ZEROs fill in vacated bit positions. A_1 is shifted to the C bit. Bits shifted out of C are discarded. After 16 or more shifts, the A register contains ZERO. The C bit takes the state of the last bit shifted out of the register.	1 + n/2
LGR	SH	0404N	Logical Right Shift	 The A register is shifted right (n) positions. ZEROs fill in vacated bit positions. A_{16} is shifted to the C bit. Bits shifted out of C are discarded. After 16 or more shifts, the A register contains ZERO. The C bit takes the state of the last bit shifted out of the register.	1 + n/2

Table 2-2. Instruction Repertoire (cont)

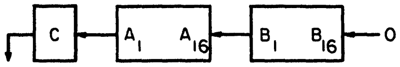
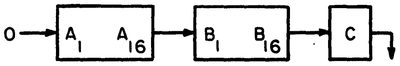
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
LLL	SH	0410N	Long Left Logical Shift	 The A and B registers are treated as a single 32-bit register (A being the most significant) and shifted left n positions. Zeros are shifted into vacated positions of B. Bits are shifted from B_1 to A_{16}. Each bit shifted out of A_1 enters the C bit. Bits shifted out of the C bit are discarded. If 32 or more shifts are specified, the A and B registers will contain ZERO. The C bit takes the state of the last bit shifted out of the register.	1 + n/2
LRL	SH	0400N	Long Right Logical Shift	 The A and B registers are treated as a single 32-bit register (A being the most significant) and shifted right n positions. Bits shifted out of A_1 enter B_1. Bits shifted out of B_{16} enter the C bit. Bits shifted out of C bit are discarded. ZEROs are shifted into vacated positions through A_1. The C bit takes the state of the last bit shifted out of the register. If 32 or more shifts are specified, the A and B registers will contain ZERO.	1 + n/2

Table 2-2. Instruction Repertoire (cont)

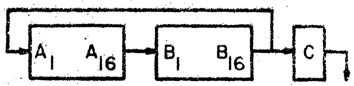
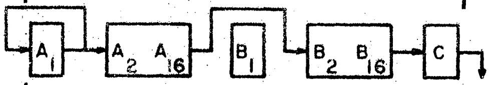
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
LRR	SH	0402N	Long Right Rotate	 The A and B registers are treated as a single 32-bit register (A being the most significant) and shifted right, end-around (n) positions. Bits shifted out of A_{16} enter B_1. Bits shifted out of B_{16} enter A_1 and the C bit. Bits shifted out of C are discarded. The C bit takes the state of the last bit shifted into A_1.	1 + n/2
LRS	SH	0401N	Long Arithmetic Right Shift	 A_1 A_2 A_{16} B_1 B_2 B_{16} C The A and B registers are treated as a single 31-bit register (B_1 is not changed) and shifted right (n) positions. The sign bit, in A_1, is not unchanged; it is propagated into vacated positions of the register. Bits shifted out of A_{16} enter B_2. Bits shifted out of B_{16} enter the C bit. (Bits shifted out of the C bit are discarded.) After 30 or more shifts, both registers are filled with the sign of the A register, except for B_1 which is unchanged. The C bit takes the state of the last bit shifted out of the register.	1 + n/2

Table 2-2. Instruction Repertoire (cont)

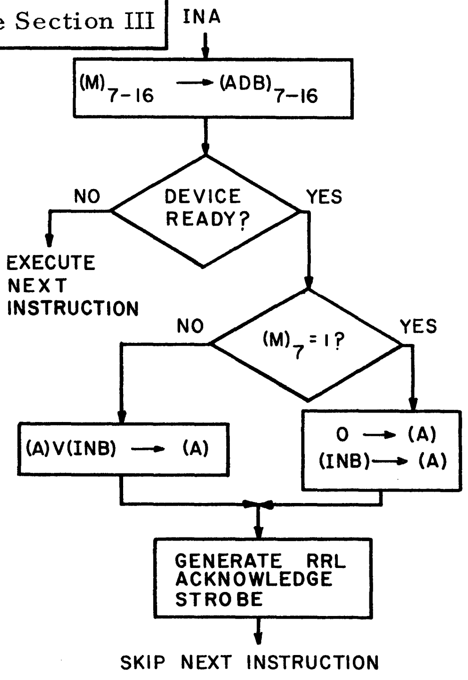
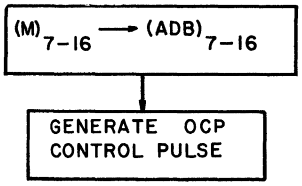
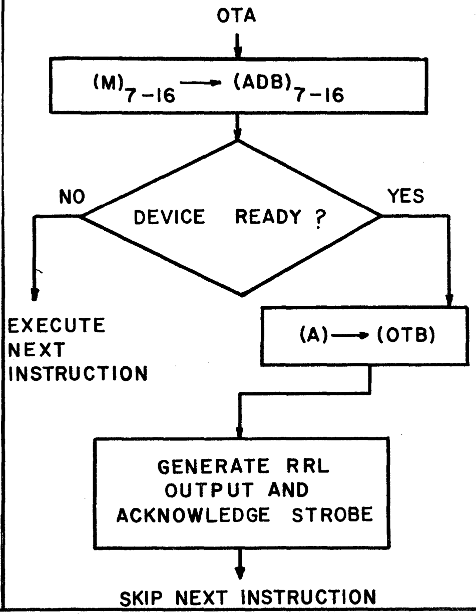
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
Input-Output		For I/O Discussion see Section III			
INA	IO	54 For INA Codes see Appendix	Input to A	 INA $(M)_{7-16} \rightarrow (ADB)_{7-16}$ DEVICE READY? NO YES EXECUTE NEXT INSTRUCTION $(M)_7 = 1$? NO YES (A)V(INB) → (A) 0 → (A) (INB) → (A) GENERATE RRL ACKNOWLEDGE STROBE SKIP NEXT INSTRUCTION	2
OCP	IO	14 For OCP codes see Appendix	Output Control Pulse	 $(M)_{7-16} \rightarrow (ADB)_{7-16}$ GENERATE OCP CONTROL PULSE	2
OTA	IO	74 For OTA codes see Appendix	Output from A	 OTA $(M)_{7-16} \rightarrow (ADB)_{7-16}$ DEVICE READY ? NO YES EXECUTE NEXT INSTRUCTION (A) → (OTB) GENERATE RRL OUTPUT AND ACKNOWLEDGE STROBE SKIP NEXT INSTRUCTION	2

Table 2-2. **Instruction Repertoire (cont)**

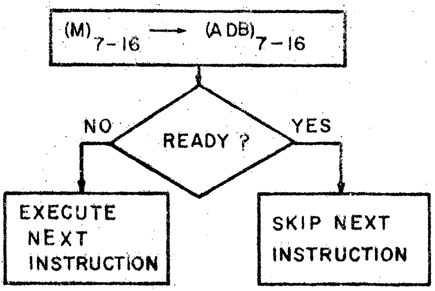
Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
SMK	IO	74 For SMK codes see Appendix	Set Mask (Special OTA)	(A) → (OTB) Generate SMK pulse to transfer output bus to external device mask flip-flops. This instruction does not skip.	2
SKS	IO	34 For SKS codes see Appendix	Skip if Ready Line Set	$(M)_{7-16} \rightarrow (ADB)_{7-16}$  $(M)_{7-16} \longrightarrow (ADB)_{7-16}$ READY ? NO YES EXECUTE NEXT INSTRUCTION SKIP NEXT INSTRUCTION	2
Control					
CAS	MR	11	Compare	Algebraically compare (A) and [EA] If (A) > [EA], execute next instruction If (A) = [EA], skip next instruction If (A) < [EA], skip two instructions	3
ENB	G	000401	Enable Program Interrupt	Set machine status to permit interrupt. The permit interrupt status will not take effect until the instruction immediately following ENB is completed. (PI indicator lights.)	1
HLT	G	000000	Halt	Sets machine to halt mode. No further instructions or interrupts will be serviced until the console START button is pressed, at which time normal execution resumes.	
INH	G	001001	Inhibit Program Interrupt	Resets "permit interrupt status" to prohibit standard or priority interrupts. (PI indicator is extinguished.)	1

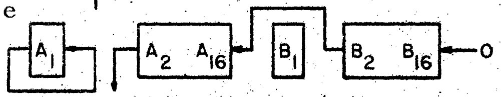
Table 2-2. Instruction Repertoire (cont)

Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
IRS	MR	12	Increment, replace and Skip	$[EA] + 1 \rightarrow [EA]$ If $[EA] + 1 = 0$, skip next instruction	3
JMP	MR	01	Unconditional Jump	$EA \rightarrow (P)$ Next instruction to be executed is at location EA.	1
JST	MR	10	Jump and Store Location	$(P_{3-16}) \rightarrow [EA_{3-16}]$ $[EA_{1,2}]$ not changed $EA_{3-16} + 1 \rightarrow (P_{3-16})$	3
NOP	G	101000	No Operation	Performs no operation; proceeds to next instruction.	1
RCB	G	140200	Reset C Bit	O → (C)	1
SCB	G	140600	Set C Bit	1 → (C)	1
SKP	G	100000	Unconditional Skip	Skip next instruction	1
SLN	G	101100	Skip if (A_{16}) One	If $(A_{16}) = 1$: skip next instruction	1
SLZ	G	100100	Skip if (A_{16}) Zero	If $(A_{16}) = 0$: skip next instruction	1
SMI	G	101400	Skip if A Minus	If $(A_1) = 1$: skip next instruction	1
SNZ	G	101040	Skip if A Not Zero	If $(A) \neq 0$: skip next instruction	1
SPL	G	100400	Skip if A Plus	If $(A_1) = 0$: skip next instruction	1
SR1	G	100020	Skip if Sense Switch 1 is Reset	If Sense Switch 1 is OFF: skip next instruction	1
SR2	G	100010	Skip if Sense Switch 2 is Reset	If Sense Switch 2 is OFF: skip next instruction	1
SR3	G	100004	Skip if Sense Switch 3 is Reset	If Sense Switch 3 is OFF: skip next instruction	1

Table 2-2. Instruction Repertoire (cont)

Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
SR4	G	100002	Skip if Sense Switch 4 is Reset	If Sense Switch 4 is OFF: skip next instruction	1
SRC	G	100001	Skip if C Reset	If (C) = 0: skip next instruction	1
SS1	G	101020	Skip if Sense Switch 1 is Set	If Sense Switch 1 is ON: skip next instruction	1
SS2	G	101010	Skip if Sense Switch 2 is Set	If Sense Switch 2 is ON: skip next instruction	1
SS3	G	101004	Skip if Sense Switch 3 is Set	If Sense Switch 3 is ON: skip next instruction	1
SS4	G	101002	Skip if Sense Switch 4 is Set	If Sense Switch 4 is ON: skip next instruction	1
SSC	G	101001	Skip if C Set	If (C) = 1: skip next instruction	1
SSR	G	100036	Skip if No Sense Switch Set	If no Sense Switches are ON: skip next instruction	1
SSS	G	101036	Skip if Any Sense Switch Set	If any Sense Switch is ON: skip next instruction	1
SZE	G	100040	Skip if A Zero	If (A) = 0: skip next instruction	1
Half-Word					
CAL	G	141050	Clear A, Left Half	$O \rightarrow (A_{1-8})$ (A_{9-16}) are unchanged	1
CAR	G	141044	Clear A, Right Half	$O \rightarrow (A_{9-16})$ (A_{1-8}) are unchanged	1
ICA	G	141340	Interchange Characters in A	$(A_{1-8}) \leftrightarrows (A_{9-16})$ A_1 is interchanged with A_9, A_2 with A_{10}, etc.	1

Table 2-2. Instruction Repertoire (cont)

Mnemonic	Type	Op Code	Definition	Description	Time (μsec)
ICL	G	141140	Interchange and Clear Left Half of A	$(A_{1-8}) \rightarrow (A_{9-16})$ $0 \rightarrow (A_{1-8})$ Bits 9-16 of A are replaced with bits 1-8; bits 1-8 are cleared.	1
ICR	G	141240	Interchange and Clear Right Half of A	$(A_{9-16}) \rightarrow (A_{1-8})$ $0 \rightarrow (A_{9-16})$ Bits 1-8 of A are replaced with bits 9-16; bits 9-16 are cleared.	1
Arithmetic and Double Precision					
MPY	MR	16	Multiply	(A) x [EA] → (A, B)	5.5
DIV	MR	17	Divide	(A, B) ÷ [EA] → (A) Remainder → (B) Overflow Status → (C) If initial magnitude of dividend is ≥ magnitude of divisor overflow occurs	10.0 or 10.5 or 11.0
NRM	G	000101	Normalize	 Shift until $(A)_2 \neq (A)_1$; number of shifts required stored as Shift Count	1 + n/2
SCA	G	000041	Shift Count to A	Shift Count → $(A)_{11-16}$ $0 \rightarrow (A)_{1-10}$	1
DBL	G	000007	Enter Double-Precision Mode	Execute LDA, STA, ADD, and SUB as DLD, DST, DAD and DSB, respectively, until SGL is executed or MASTER CLEAR is depressed	1
SGL	G	000005	Enter Single-Precision Mode	Execute LDA, STA, ADD, and SUB in normal single precision	1
DLD	MR	02	Double-Precision Load	[EA] → (A) [EA+ 1] → (B)	3
DST	MR	04	Double-Precision Store	(A) → [EA] (B) → [EA+ 1]	3
DAD	MR	06	Double-Precision Add	(A, B) + [EA, EA+1] → (A, B) Overflow Status → (C) If $(EA + 1)_1 \neq B1$, an invalid sum results	
DSB	MR	07	Double-Precision Subtract	(A, B) - [EA, EA+ 1] →(A, B) Overflow Status → (C) If $(EA + 1)_1 \neq B_1$, an invalid difference results	3 3

SECTION III

INPUT/OUTPUT

INPUT/OUTPUT CONTROL AND COMMUNICATION

The basic communication link between the PCU and peripheral (input/output) devices is an input/output bus This bus contains 16 input lines, 16 output lines, 10 address lines and a group of control lines. As many as 20 peripheral devices may be attached to the bus. These devices then all communicate with the central processor by time sharing the bus. Since all standard I/O devices are individually buffered, and the bus is only used by a particular device while the PCU is actually transferring information to or from the device, many devices can operate concurrently. The input/output bus lines are listed in Table 3-1.

Table 3-1. Input/Output Bus Lines

Lines Available for Input/Output	Designation	Bit Capacity	Function
Output bus	OTB$_{1-16}$	16	Transmit data from the PCU to an I/O device
Input bus	INB$_{1-16}$	16	Transmit data from an I/O device to the PCU
Address bus	ADB$_{7-10}$	4	Define the function to be performed by an I/O device
	ADB$_{11-16}$	6	Define the I/O device selected
Device ready line	DRLIN	1	Transmit a signal to the PCU indicating the status of the device addressed by the I/O command
Output control pulse	OCPLS	1	Transmit a pulse from the PCU that defines the fact that an OCP command is being executed

Table 3-1. Input/Output Bus Lines (cont)

Lines Available for Input/Output	Designation	Bit Capacity	Function
Master clear	MSTCL	1	Transmit a master reset to devices
Parity error	PARCK	1	Transmit a signal to the PCU indicating that a parity error has been detected in an I/O device
Program-interrupt line	PIL00	1	Transmit a signal to the PCU indicating that a standard interrupt is requested
Set Program interrupt mask	SMK01	1	Transmit a pulse from the PCU indicating that the OTB contains a new setting for the interrupt mask flip-flops
Clear mask	CMKXX	1	Transmit a pulse from the PCU used to clear the device mask control flip-flops
Set Mask (general)	SMKXX	1	Transmit a pulse from the PCU indicating that the OTB contains a new setting for option masks specified by ADB_{7-10} & ADB_{11-16}.
Reset ready line	RRLIN	1	Transmit a pulse from the PCU which is used to strobe the output bus during an OTA instruction and to "reset ready" during the OTA and INA instructions.

The central processor is responsible at all times for determining what information is on the bus. Thus, the typical sequence of operation is for the PCU to send out on the address bus lines a 6-bit device code that identifies the device with which the central processor is communicating and a 4-bit function code indicating which function the device is to perform. If the instruction is an input to A (INA), output from A (OTA), or sense status (SKS), the device next sends back to the central processor an indication as to its condition (ready, etc.). The central processor then performs the necessary functions (input, output, skip, etc.) on the basis of the reply.

The selection of a device, the testing for its status, and the actual input or output can often be performed with a single instruction. Once each device has been set up in its proper operating mode and been started, the only instructions necessary to perform data transfers are INA or OTA instructions.

Two basic modes of input/output are available with the PCU. The standard mode is single-word input/output transfer, with or without interrupt. The second mode is the DMC (direct multiplex control) which permits input/output to and from memory without program intervention.

Single-Word Transfer Mode

The single-word transfer under program control is the basic input/output mode of the PCU. In this mode, full words or character can be read from external devices into the A-register by utilizing INA instructions, and words or characters can be transferred from the A-register to an output device by using OTA instructions. During an input operation in the single-word transfer mode, the programmer has the option of clearing or not clearing the A-register before each input. If characters are being read into the PCU, this allows the programmer to pack the characters into words in the A-register as part of a basic input routine. The ability to test and skip on the ready status of an I/O device also is included in the basic input and output instructions to make the PCU extremely flexible. Thus, the PCU is not required to hold in an input or output instruction waiting for a ready signal. This permits maximum utilization of the central processor. It also makes it convenient to handle multiple input/output devices all running simultaneously under program control. Because of the high internal speed of the PCU, quite high data transfer rates can be accommodated in the single-word transfer mode. This mode is also convenient for slower devices such as paper-tape equipment and card equipment.

The instructions which are used to operate in the single-word transfer mode are as follows:

a. Input to A (INA)
b. Output from A (OTA)
c. Sense status (SKS)
d. Output control pulse (OCP)

On each of these instructions bits 11 through 16 identify the I/O device selected, and bits 7 through 10 define the function to be performed. With the exception of bit 7 in the INA command, these bits are completely ignored by the central processor. Their only function is to serve as a command to the peripheral device.

INA Instruction. -- The INA instruction is used to input data from a device into the A-register. All 16 bits of the data are ORed into the A-register by the instruction; however, data is not necessarily placed on all lines by every device. Thus, a character input device may place data only on the eight least significant bits of the input bus leaving the other bits as ZEROs. Since the content of the input bus is always logically ORed with the A-register, the effect is as though only eight bits had been transferred from the device to the A-register. The function code portion of the INA instruction is typically used by the device to determine the mode of input (for example, binary or ASCII).

The INA instruction sends out its device and function code on the I/O bus. It then looks for a ready signal on the DRLIN (device ready line). If a ready signal is received within a predetermined time interval, the content of the INB (input bus) is logically ORed with the contents of the A-register and the next instruction is skipped. A reset-ready signal is also sent out on the RRLIN (reset ready line) to tell the device that the data has been accepted by the PCU. If bit 7 is set in the instruction, the A-register is cleared before the INB is ORed with the A-register. If a ready signal is not received, no input is performed and the next instruction is not skipped.

OTA Instruction. -- The OTA instruction is utilized to send data from the A-register to an output device. All 16 bits of the A-register are sent out on the I/O bus; however, not all may be accepted by a particular device. Thus, a character device might receive only the eight least significant bits of the data. The function code portion of the instruction is typically used by the device to determine the mode of output (for example, binary or ASCII).

This instruction sends out its device and function code and the contents of the A-register on the I/O bus. It then looks for a ready signal on the DRLIN (device ready line). If a ready signal is received within a predetermined time interval, an output pulse is sent out on the RRLIN line indicating to the device that it may take data off the OTB (output bus). The next instruction is then skipped. If a ready signal is not received, no output function is performed, and the next instruction is not skipped.

OCP Instruction. -- The OCP instruction is used to set up the operating mode of a device, to start the device, etc. This instruction sends out its device and function code on the I/O bus. It also sends an output control pulse on the OCPLS line after the device has had time to receive and decode the address and function bits. The function bits in this instruction are used to determine the particular function that the OCPLS pulse is required to perform. The DRLIN line is not examined during this instruction, and the next instruction is never skipped.

SKS Instruction. -- The SKS instruction is used to test different conditions in the device. Thus, it might test for "power on," "tape moving," "device busy," "device ready," etc. It is also used to supplement the device-ready test included in the INA and OTA instructions. The function bits are used to determine the particular condition to be tested.

This instruction sends out its device and function code on the I/O bus. It then looks for a status signal on DRLIN. If an affirmative status signal is received within the prescribed time interval, the next instruction is skipped. If an affirma·ive status signal is not received, the next instruction is not skipped

DMC MODE

The Direct Multiplex Control (DMC) permits data transfer between peripheral devices and the computer memory concurrently with computation.

When a device has data to input, or is ready to accept data, it uses the DMC control lines to request service. Devices request service from the DMC on lines called DIL. DIL line 1 has highest priority, line 16 has lowest. The priority network will allow the highest priority line which has its DIL set to be serviced by the next DMC cyrle.

When a DMC cycle is required, the DMC will send a break request to the CPU When the CPU has completed the current instruction, a DMC cycle will be executed. During this cycle the appropriate transfer between the device and the memory will take place, using the standard I/O bus.

This process is repeated each time the I/O dev'ce indicates that it is ready until the required number of words has been transferred When the required number of words have been transferred, the DMC sends an End of Range (ERL) signal to the device. The device may use this signal to generate a program interrupt.

Each channel requires a starting and ending address for the block transfer. These addresses (a pair per channel) are stored in dedicated memory locations (listed in Table 3-2).

The remaining bits specify the address limits of the data block. In input mode, data from the device will be stored beginning at this address. In output mode, data beginning at this address will be sent to the device. Input or output mode is set by the device.

Table 3-2. DMC Start and Terminal Memory Address Locations

Channel Number	Starting Address	Ending Address
1	00020	00021
2	00022	00023

The DMC can effect a transfer following any instruction, provided a DMC request from a device is transmitted to the DMC 0.6 μsec before the end of that instruction. If a request occurs less than 0.6 μsec before the end of an instruction, the DMC cycle may not occur until after the next instruction.

The data transfer is completed 1.74 μsec into the DMC cycle for an input, 3.0 μsec for an output. Thus, the longest waiting time, from the time a request occurs to the time the data transfer is completed is:

$$T_{wc} = T_{li} + 3.84M + 1.2N + \begin{matrix} 2.34 & \text{(input)} \\ 3.60 & \text{(output)} \end{matrix}$$

where

T_{wc} = worst-case waiting time (μsec) from request to completion of data transfer.

T_{li} = execution time of longest* instruction (μsec).

*The longest useful instruction in the CPU repertoire is executed in 16.32 μsec. (Shifts of more than 32 places and memory reference instructions with more than six levels of indirect addressing are not considered "useful" in this context.) Lower values of T_{li} may be used to facilitate input-output buffer design, provided appropriate programming constraints are adopted.

M = number of higher priority DMC requests which may occur during T_{wc}.

N = number of DMC requests which may occur during T_{wc}.

Each DMC cycle requires four memory cycles, or 4 μsec, during which computation is suspended. At 0.6 μsec before the end of a DMC cycle the device request lines are inspected. If a device is requesting at this time, another DMC cycle will immediately follow the first. DMC cycles will continue as long as requests are waiting. During this time the CPU cannot resume control.

The maximum transfer rate of a single DMC channel is one word every four cycles or 250 KC. This rate can be attained if this channel is the only channel being used. If the DMC is operating at 250 KC, no computation can take place. In order to operate between 200 KC - 250 KC, T_{1i} must be 1 μsec.

DMC Sub-Channel

A device is connected to the DMC control unit through a DMC sub-channel. The DMC sub-channel, available as an option on a number of standard I/O devices, contains the necessary logic to permit the device to operate in the DMC mode.

Standard Interrupt

The basic interrupt system consists of a single interrupt line. All standard I/O devices are connected to this line. A total of 16 interrupt sources can be connected on this line. Each source also has an interrupt mask bit which can inhibit an interrupt signal from being gated onto the interrupt line. The mask can be set and reset by an SMK '0020 instruction, which transfers the contents of the A-register via the OTB to the mask bits of standard devices as listed in Table 3-3. Thus, the program has the ability to selectively inhibit interrupt sources. This selective inhibiting of interrupt sources permits a multilevel priority interrupt system to be programmed in which an interrupt subroutine can be interrupted in turn by a program of even higher priority.

Furthermore, because all interrupt sources connect with the computer via the I/O bus, the logic associated with all the interrupt sources does not have to be centralized in a priority interrupt unit; it can be located wherever it is most convenient to place it. In particular, I/O devices can be handled on a priority interrupt basis by merely adding the necessary logic to the device control unit.

When the interrupt line is activated by an external source, the PCU inhibits all further interrupts and generates a jump and store location instruction (JST) indirectly through location $(00063)_8$. If more than one interrupt source is connected to the interrupt line, the program proceeds to an interrupt service routine which tests the sources one by one with sense status commands (SKS). When the routine finds the source which caused the interrupt, it jumps to the appropriate subroutine. The program then sets up a new status for the interrupt mask bits for all of the interrupt sources. The new status determines the sources that have a higher priority than the one which actually interrupted. The program then enables interrupt and proceeds.

The signals in the I/O bus which are used for interrupt are as follows:

a. PIL00 - This ORs together interrupt request signals from all standard interrupt sources and sends them to the CPU.

b. DRLIN - This line is used by the SKS instruction to test each individual interrupt source in order to check whether it is requesting an interrupt. The device address is sent out which selects the device, and a particular function code is sent out which places the status of the priority interrupt request logic on DRLIN.

c. SMK01 - This line from the CPU is used in place of a device address and a function code to indicate that a new status for the interrupt mask bits in the system is on the OTB.

Table 3-3. Standard Interrupt Mask Assignments

OTB Bit No.	Device	OTB Bit No.	Device
1	TCU	9	(Unassigned)
2	Print Head Control	10	(Unassigned)
3	On-Line Interface Controller	11	ASR-33
4	(Unassigned)	12	(Unassigned)
5	I/O Channel No. 1	13	(Unassigned)
6	I/O Channel No. 2	14	(Unassigned)
7	(Unassigned)	15	(Unassigned)
8	(Unassigned)	16	(Unassigned)

Power Failure Interrupt (PFI)

The PCU contains a PFI circuit which acts as a memory protection feature. If the primary PCU ac input power fails or is turned off at the control console while the PCU is in the ''RUN'' mode, the PFI circuit either halts the PCU or forces an interrupt to a pre-assigned memory location. The operation performed by the PFI on the detection of a power failure is dependent on the position of a console PFI/PFH control switch.

If the control switch is in the PFI position, the detection of a power failure will cause the PFI to initiate an interrupt during which the PCU is forced to perform an indirect JST to memory location $(00060)_8$. The PFI interrupt will occur at least one millisecond before the dc power drops below the guaranteed operating limits of the circuits.

If the control switch is in the PFH position, the detection of a power failure causes the PFI to place the PCU in a halt state. No information in memory will be altered when power fails.

SECTION IV

MAGNETIC TAPE OPERATION

Functional Characteristics

The functional characteristics of the tape control unit (TCU) are as follows:

Tape Units	- IBM 729 Mod II, IV, V, or VI - IBM 2400 Mod 1, 2, or 3 either 7 or 9 track - Provision is made for interfacing either 729, 2400, or both 729 and 2400 tape units.
Number of tape units	- 8 maximum
Tape Densities	- 200, 556, or 800 bits per inch (bpi) Any two densities may be selected on the PCU control panel and placed under program control.
Basic Operations	- Space forward 1 record - Read forward 1 record - Write forward 1 record - Space backward 1 record - Read backward 1 record - Rewind - Rewind Unload
Computer Interface	- Data transferred by Data Multiplex Channel 1. - This is a buffered I/O channel which permits simultaneous computation with read or write operations. Interrupts are used to give the TCU priority over other simultaneous computer programs.

Error Detection

- Lateral parity, Odd or Even (VRC)
- Longitudinal Parity, (LRC)
- Cyclic Redundancy Code (CRC)
- Lost Character
- Echo Error

Error Correction

- Programmed clip level and a special correct read command permit rereading of records at different clip levels and the transfer to memory of correct (correct VRC) data only.
- The parity bit for each tape character is stored in memory during a tape read. Consequently 9 track records are correctable by CRC correction procedure (in software) providing that errors are in one track only.

Special Operations

- Correct Read (described above)
- Repeat - The repeat function permits software to control the skipping of an arbitrary number of records or files without stopping tape motion at intermediate record gaps.
- Delay - This function causes a load point space to be made before reading or writing commences.

Tape units utilize Direct Multiplex Control (DMC) channel 1 of the PCU. This permits buffered transfers to be made from the core address stored in location '20 to the core address stored in location '21. At the completion of a transfer, location '20 holds the final address from/to which data was transferred, plus 1. The address words stored locations '20 and '21 must not exceed '17777.

Only one tape character is written from or read into each word of memory. The least significant 6 or 8 bits of the 16 bit word will contain the data that is transferred to or from tape. Table 4-1 shows the organization of data bits and tape tracks for 7 and 9 track tapes. The eight high order bits are supplied by the TCU during the read operation. These may be used for error checking.

The time required by the DMC to transfer one word to or from core is 4 microseconds. The nominal period between characters at a density of 800 bpi and a tape speed of

112.5 inches per second is 11.1 microseconds (90kc). Under worst case conditions, however, the period between characters may be as little as 6.5 microseconds. Since the DMC can acquire control of the CPU only at the end of an instruction, it is important when using 90kc tape drives to minimize the duration of CPU instructions executed simultaneously with tape read or write operations. It is recommended that a "jump here" loop (1 microsecond instructions) or a "skip if ready and jump back" loop (2 and 1 microsecond instructions) be employed during such conditions.

Table 4-1. Bit Significance in Read or Write Words

Digit	Bit	Write*		Read	
No.	No	729	2400	729	2400
1	1	X	X	VRC CHARACTER ERROR	
2	2	X	X	CLIP LEVEL CODE	
	3	X	X	"	
	4	X	X	"	
3	5	X	X	CHECK CHARACTER	
	6	X	X	LOST CHARACTER	
	7	X	X	END OF RECORD	
4	8	X	X	Track C	Track P - PARITY
	9	X	Track 0	0	" 0
	10	X	" 1	0	" 1
5	11	Track B	" 2	Track B	" 2
	12	" A	" 3	" A	" 3
	13	" 8	" 4	" 8	" 4
6	14	" 4	" 5	" 4	" 5
	15	" 2	" 6	" 2	" 6
	16	" 1	" 7	" 1	" 7
PARITY**		" C	" P		

* Bits marked X may have any value

** Write parity bit is generated in the TCU

TCU Interrupts

All tape operations except rewind cause the TCU to generate interrupt signals. If the TCU is to be allowed to interrupt the PCU, and interrupts from all other devices are to be excluded, bit 1 of the A register must be loaded with a "1" and the interrupt mask instruction issued. For example:

```
LDA     ='100000
OTA     '0020
```

Interrupts from tape operations occur at the following times:

First Interrupt - Prior to the end of tape motion (start) delay or the end of read skip.

Second Interrupt - At the earlier of the DMC end of range signal (the specified number of data characters have been transferred) or at the end of record.

Third Interrupt - Prior to operation complete.

The timing of the first and third interrupts is controlled by a clock whose speed is proportional to tape speed. At least 64 microseconds is available between the interrupt and the beginning of data transfers for the highest speed tape drives. The interrupts may be used to minimize loss of computation time due to tape motion delays. If the Repeat function is used, for example to skip records without stopping at the record gaps, it must be set after interrupt 1 and before interrupt 2. The timing of the different interrupts are given in Table 4-2.

Tape Function Code

Tape operations are controlled by the tape function code which is a 16 bit word transferred to the TCU (see OTA 1210). Table 4-3 shows the structure of the tape function code.

Table 4-2. Interrupts During Tape Operations

Operation	Interrupt Number	Interrupt Timing
SPACE READ WITHOUT SKIP WRITE	1	400_8 speed clock counts before the opening of the read or write gate. This interval corresponds to 64 μs at 112.5 ips 96 μs at 75 ips 192 μs at 37.5 ips
	2	The earlier of ERL - (DMC end of range) or END RECORD
	3	Operation complete*
READ WITH SKIP	1	32 characters before the first character to be stored in memory.
	2	The earlier of ERL or END RECORD
	3	Operation complete*
REWIND	1	At deselection of tape unit.
FAULT	1	Any fault sets interrupt and stops the tape unit.

*In repeat operations interrupts 1 and 2 are repeated for each record. The operation complete interrupt does not occur until the tape unit is given a normal stop.

Table 4-3. Tape Function Code

Tape Function Code - Octal Decoding

Digit No	Bit No	Value Octal	Significance	Value Octal	Significance
1	1	1	High Density		
2	2	4	Even Parity		
	3	2	Delay (LD PT SPACE)		
	4	1	Correct Record		
3	5		Operation Code		
	6		Operation Code		
	7		Operation Code		
		0	Space Forward	4	Space Backward
		1	Read Forward	5	Read Backward
		2	Write With CRC	6	Rewind
		3	Write Without CRC	7	Rewind Unload
4	8		Clip Level Code		
	9		Clip Level Code		
	10		Clip Level Code		
		0	1 volt P-P Clip Level	4	3.7 volt P-P Clip Level
		1	1.7	5	4.5
		2	2.2	6	5
		3	3.0	7	6
5	11		Read Skip Character Count Code		
	12		Read Skip Character Count Code		
	13		Read Skip Character Count Code		
		0	No Skip	4	Skip 2560
		1	Skip 1024	5	Skip 3072
		2	1536	6	3584
		3	2048	7	4096
6	14		Select Tape Code		
	15		Select Tape Code		
	16		Select Tape Code		
		0	Tape Unit 0	4	Tape Unit 4
		1	Tape Unit 1	5	Tape Unit 5
		2	Tape Unit 2	6	Tape Unit 6
		3	Tape Unit 3	7	Tape Unit 7

If the special tape interrupts are to be allowed, the following instructions should be given in the order listed below:

	Operation	Coding
1.	Set tape function	LDA (tape function code) OTA '1210 JMP *-1
2.	Clear TCU	OCP '1710
3.	Load and store interrupt jump	LDA (location of interrupt service routine) STA '0063
4.	Set tape interrupt mask (disable other interrupts)	LDA ='100000 OTA '0020
5.	Enable interrupt	ENB

After an interrupt occurs, it is necessary to reset the present interrupt condition by executing

SKS '1310

before re-enabling for subsequent interrupts

Basic Tape Operations

The following instructions are used to perform the basic tape operations:

SKS '1010 Skip if TCU Ready

This instruction causes a skip of one instruction if the TCU is ready. This instruction may be given at any time.

INA '1110 Sense Tape Status

This instruction causes a 16-bit sense word to be transferred from the TCU to the A-register. This instruction may be given at any time but it is normally done following the setting of a tape function and after each interrupt to access the progress of the tape operation. The significance of the bits in the sense word is listed in Table 4-4.

No conditions are placed on the acceptance of the instruction; thus the instruction will always elicit a response.

INA '0110 Sense Tape Status

Same as for INA '1110 except that the status word is ORed with the contents of the A-register.

SKS '1310 Skip if TCU is Not Interrupting and Reset TCU Interrupt

This instruction causes a skip of one instruction if the TCU is not interrupting. It also resets the interrupt. To enable the PCU to receive the next TCU interrupt, it is necessary to set both the tape unit interrupt mask (LDA = '100000 and OTA '0020) and to enable the PCU reception of interrupts by giving the ENB instruction.

OTA '1210 Set Tape Function

This instruction transfers to the TCU a 16-bit function word from the A-register (see Table 4-3). Since this instruction is accepted by the TCU only when the TCU is ready (not busy), the instruction may be placed in a waiting loop to send a new function word as soon as the last tape operation is complete.

OCP '1510 Enable Tape DMC Channel

This instruction starts the transfer of data to/from the TCU through DMC channel 1.

OCP '1610 Set TCU to Repeat Status

This instruction causes the TCU to repeat the current operation without stopping the tape at record gaps. This instruction must be given immediately following the first TCU interrupt (see Table 4-2).

OCP '1710 Clear TCU

Clearing the TCU may be performed any time that the TCU is ready without changing subsequent operations. It will not clear error detection flip-flops or change the function held in the function register.

OCP '1410 Start Tape Action

This instruction starts the tape function specified in the function register providing the TCU is ready (not busy) and there are no TCU fault conditions present.

Table 4-4. Bit Significance In The Sense Word

Digit No.	Bit No.	Significance
1	1	VRC ERROR
2	2	MARK RECORD
	3	LRC ERROR
	4	NOT COUNTER OVERFLOW (No data has been read for a distance of approximately 48 inches on the tape)
3	5	ECHO ERROR
	6	CRC ERROR
	7	LOST CHARACTER (One or more characters have been detected with spacing more than 1.4 and less than 2.25 times the normal character spacing.)
4	8	7 TRACK
	9	SELECTED TAPE READY
	10	FAULT*
	11	REPEAT (The repeat function is set. When this is true immediately following interrupt 2, the tape drive will not stop in the record gap but will pass over and repeat the same operation on the next record.)
	12	NOT FILE PROTECT (applies to 2400 tape units only)
	13	READ STATUS
6	14	LOAD POINT
	15	END OF TAPE
	16	INCOMPLETE TRANSFER (The DMC end of range was reached before the end of record. Applicable to READ operations only.)

* FAULT CONDITIONS

Selected Tape Unit Not Ready
Backward Status at Load Point
Blank Tape
Write and Read Status
Read and Write Status

SECTION V

ASR-33 TELETYPE

ASR-33 TELETYPE UNIT

The ASR-33 Teletype unit is available as a basic I/O device with the PCU. The ASR-33 is a versatile device providing a capability to read paper tape at 10 characters/second, and punch paper tape at the same rate. The ASR-33 may also print out data from the PCU at 10 characters/second and transfer data to the PCU from the keyboard. In the local mode, the unit may be used for off-line paper tape preparation, reproduction and listing.

Keyboard and Carriage Features

The ASR-33 keyboard is similar to that of a standard typewriter. The keyboard includes four rows of keys and generates an eight-level code. Letters and numerals are transmitted without a shift, similar to lower-case transmission on a typewriter. Printing characters (?, =, *, etc.) are typed by using the shift key, similar to upper-case positions on certain typewriter keys. Control functions, generated using the control (CTRL) key, are X-OFF (S-key), X-ON (Q-key), EOM (C-key) and BELL (G-key). The LINE FEED and RETURN codes are transmitted without the CTRL key being depressed.

The ASR-33 is capable of printing a 72-character line. If a programmer wishes to print 72 or fewer characters, he must perform a carriage return and line feed (in that order) after the last character desired in each line.

Keyboard Interlock. -- The ASR-33 keyboard is interlocked for all keys except the SHIFT, CTRL and REPT keys, preventing more than one key from being depressed at a time. The keyboard does not lock in the upper-case position. Therefore, the operator must hold the SHIFT key depressed to produce upper-case characters.

Operating Modes - ASR-33 Unit

Tape Reader

Starting. -- The reader is started under program control as follows:

a. Enable the ASR-33 in the output mode using OCP 104.
b. Output an X-ON character $(221)_8$ using OTA 004.
c. Delay while the ASR-33 is busy (test with SKS 104)
d. Enable the ASR-33 in the input mode using OCP 004.

To use this method, the ASR-33 must first be set up in the output mode by an OCP. After the X-ON character to the ASR-33 buffer is outputted by an OTA, the SKS not busy test must follow. When the not busy indication is obtained, the ASR-33 must then be OCP'd in Input Mode; whereupon INAs can then be executed. Manual starting is controlled by the START/STOP switch. After the reader is started, the first character to be read is the one initially positioned over the read pins.

Stopping. -- The reader stops automatically only when an "X-OFF" code $(223)_8$ or $(023)_8$, is read from paper tape. The X-OFF character will be transmitted into the device's buffer and the character following will be transmitted into the buffer before the reader stops. Manual stops are controlled by the START/STOP switch. (The reader also stops automatically when it runs out of paper tape.)

Overriding Stop Code. -- A stop code will stop the reader while tape is being duplicated off line. To continue duplicating, manually restart the reader with the START switch.

Tape Punch

The punch is controlled by manual operation of the punch ON-OFF switch located on the ASR-33. When the punch is on, any input from or output to the ASR-33 will cause tape to be punched. Tape leader may be generated in bursts of 20 sprockets with each depression of the HERE-IS key.

Off-Line Operation

Off-line operation of the ASR-33 includes the following data transmission.

a. Keyboard to printer
b. Keyboard to printer and punch
c. Reader to printer
d. Reader to printer and punch

ASR-33 On-Line Operating Modes

There are two basic modes of operation for the ASR-33 when on line: input mode and output mode. These are set up by the appropriate OCP instruction. Once set up, the ASR-33 remains in a given mode until it is changed by another OCP.

Input Mode. -- The input mode is used to transmit information from the ASR-33 keyboard to the PCU or from the reader to the PCU. In either case, printed copy is produced if the 8-bit character is printable, and a control function is performed if the 8-bit character is a control character (see Appendix C). If characters are being read from the reader, any of the 256 possible 8-bit characters appearing on the tape will be transmitted to the PCU. When an X-OFF, $(223)_8$, or $(023)_8$ is read, the reader will stop after reading the character following the X-OFF, unless that following character is an X-ON $(221)_8$ or $(021)_8$.

Output Mode. -- The output mode is used to transmit information from the PCU to the ASR-33 printer or the printer and the punch. In either case, printed copy is produced if the 8-bit character is printable, and a control function is performed if the 8-bit character is a control character. When punching, any 8-bit code (of the possible 256) transmitted from the PCU will be punched whether it is printable or not. Certain 8-bit codes -- $(221)_8$, $(021)_8$, $(005)_8$ -- when transmitted from the PCU will also cause a control action by the ASR-33/35 and prevent proper transmission of further characters. X-ON, $(221)_8$ or $(021)_8$ will start the paper tape reader, and WRU, $(205)_8$ or $(005)_8$ will trigger the answer-back drum.

Character Modes

Within either the input or output modes, either of two character modes, ASCII or binary, may be used. Code type is selected by inidividual INA or OTA instructions and may be intermixed in any manner (though this is not normally done).

ASCII Mode. -- In the ASCII mode a full 8-bit character is transmitted to or from the least significant 8 bits of the A-register and the ASR-33. This permits transmission of any standard character or control character from the reader or keyboard of the ASR-33 to the PCU or from the PCU to the printer or punch of the ASR-33.

Binary Mode. -- In the binary mode a 6-bit character is transmitted to or from the least significant 6 bits of the A-register and the ASR-33. In the case of output in the binary mode, an additional 2 bits are automatically added in the high-order position to the 6-bit character to form an 8-bit character acceptable to the ASR-33. The 2 bits added are chosen so that the resulting 8-bit character is an alphanumeric character, not a control character. On input, the two high-order bits of the 8-bit character transmitted by the ASR-33 are stripped and ignored.

Instructions

The following instructions are used to control the ASR-33 and to transfer data to and from it.

OCP '0004 Enable ASR-33 in Input Mode

This instruction sets up the device interface to accept characters from the ASR-33. It should be given any time it is desired to switch the ASR-33 from the output mode to the input mode. This instruction must not be given while the ASR-33 is busy. Thus, an SKS "not busy" test should precede the instruction.

OCP '0104 Enable ASR-33 in Output Mode

This instruction sets up the device interface to transmit characters to the ASR-33. The instruction must be given any time it is desired to switch from the input to the output mode. The instruction must not be given while the ASR-33 is busy. Thus, an SKS "not busy" test should precede the instruction.

SKS '0404 Skip if ASR-33 Is Not Interrupting

This instruction tests whether the ASR-33 has caused an interrupt on the standard interrupt line.

SKS '0004 Skip if ASR-33 Is Ready in ASCII Mode

This instruction tests whether the ASR-33 device interface is ready to accept another character from the PCU in ASCII output mode or to present another character to the PCU in ASCII input mode.

SKS '0104 Skip If ASR-33 Is Not Busy

The ASR-33 busy signal is defined as follows:

a. In the output mode the ASR-33 is busy from the time a character is transmitted from the PCU to the ASR-33 device interface until it has been serially shifted out to the ASR-33. This time is approximately 105 ms.

b. In the input mode the ASR-33 is busy from the time the ASR-33 starts to serially transfer a character to the device interface until the transfer is complete and the ASR-33 ready condition is present. This time is approximately 100 ms.

SKS '0204 Skip If ASR-33 Is Ready in Binary Mode

This instruction tests whether the ASR-33 device interface is ready to accept another character in binary output mode or to present another character to the PCU in binary input mode.

SKS '0504 Skip If Stop Code Was Not Read on ASR-33

This instruction tests whether a stop code $(223)_8$ or $(023)_8$ has been read on the ASR-33. The stop code indication can be tested as soon as the stop code has been read from the ASR-33 into the device buffer and is ready for input to the PCU. When a stop code is read by an ASR-33, the stop code and ONE/TWO following characters will be transferred to the device buffer before the reader stops. The stop code indication will remain present until the character following the stop code is ready for input to the PCU (approximately 100 ms).

INA '0004 Input in ASCII Mode If Ready*

This instruction transmits the full 8-bit character from the ASR-33 to the 8 least significant bits of the A-register. The A-register is not cleared.

INA '0204 Input in Binary Mode If Ready*

This instruction transmits the 6 least significant bits of the 8-bit ASR-33 character to the 6 least significant bits of the A-register. The A-register is not cleared.

INA '1004 Clear A and Input in ASCII Mode If Ready*

INA '1204 Clear A and Input in Binary Mode If Ready*

OTA '0004 Output in ASCII Mode If Ready

This instruction transmits the 8 least significant bits of the A-register to the ASR-33. If the ASR-33 is punching, it will punch all 8 bits of the code that is transmitted. However, in printing, it will determine the character to be printed or the control function to be performed from the 7 least significant bits.

OTA 0204 Output in Binary Mode If Ready

This instruction transmits the 8 least significant bits of the A-register to the ASR-33 and then modifies channel 7 (normally A10) to form a valid ASCII alphanumeric character. To do this, bit 7 is made the inverse of A11. Thus, if the 8 least significant bits in the A-register were $(XXX1XXXXX)_2$, they would be transmitted to the ASR-33 as $(X01XXXXX)_2$. If they were $(XX0XXXXX)_2$, they would be transmitted as $(X10XXXXX)_2$

*READY must be honored within one millisecond to ensure taking the character. If another input mode is to follow an input from reader during which a stop code is encountered, the READY signal for the buffer character following the "X-OFF" must be horored or a new input OCP instruction must be issued by the PCU

Standard Interrupt

The OTB mask bit assignment for standard interrupt is OTB 11.

Input Mode. -- An interrupt request will occur when data is in the buffer and Ready is set. When the interrupt is honored by executing an INA and data is transferred, the controller will not be busy; interrupt request will be reset and can accept another OCP command if desired.

Output Mode. -- An interrupt request will occur whenever the Ready Flip-Flop is set (controller ready to accept data from the CPU). The request can be reset by executing an OTA or OCP input command.

Paper Tape Format and ASR Codes

The format of the ASR-33 paper tape is shown in Figure 5-1. The codes for the ASR-33 characters and symbols are shown in Table 5-1.

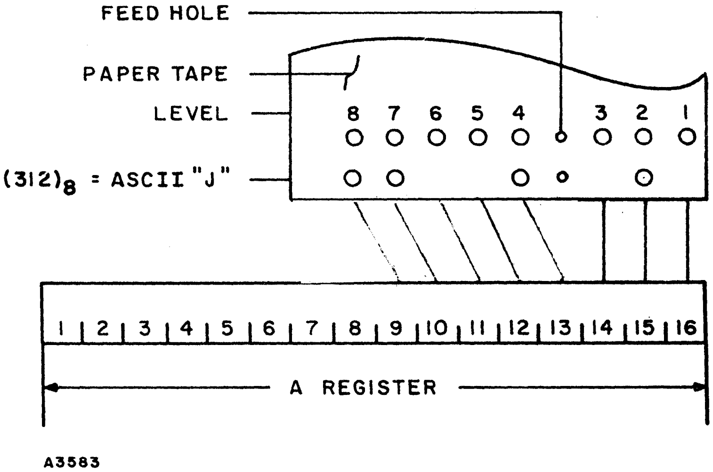


Figure 5-1. ASR-33 Paper Tape Format

Table 5-1. ASR-33 Characters and Symbol Codes

KEY	Lower Case Code	SYM	Shift Code	SYM	Control Code	SYM
0	260	0	LO		260	0
1-!	261	1	241	!	261	1
2-"	262	2	242	"	262	2
3-#	263	3	243	#	263	3
4-$	264	4	244	$	264	4
5-%	265	5	245	%	265	5
6-&	266	6	246	&	266	6
7-'	267	7	247	'	267	7
8-(	270	8	250	(	270	8
9-)	271	9	251	)	271	9
A	301	A	LO		201	
B	302	B	LO		202	
C	303	C	LO		203	
D-EOT	304	D	LO		204	
E-WRU	305	E	LO		205	
F-RU	306	F	LO		206	
G-BELL	307	G	LO		207	
H	310	H	LO		210	
I-TAB	311	I	LO		211	
J	312	J	LO		212	LF
K-VT	313	K	333	[	213	
L-FORM	314	L	334	\	214	
M	315	M	335	]	215	CR
N-↑	316	N	336	↑	216	↑
O-←	317	O	337	←	217	
P-@	320	P	300	@	220	
Q-XON	321	Q	LO		221	
R-TAPE	322	R	LO		222	
S-XOFF	323	S	LO		223	
T-TAPE	324	T	LO		224	
U	325	U	LO		225	
V	326	V	LO		226	
W	327	W	LO		227	
X	330	X	LO		230	
Y	331	Y	LO		231	
Z	332	Z	LO		232	
:-*	272	:	252	*	272	:
-(minus)- =	255		275	=	255	-
HERE IS	000		020		000	
ALT MODE	375		LO		275	=
LF	212		LO		212	
CR	215		LO		215	
;-+	273	;	253	+	273	;
RUB OUT	377		LO		277	?
BREAK	NOTE					
, -<	254		274	<	254	,
. ->	256	.	276	>	256	.
/-?	257	/	277	?	257	/
SPACE	240		LO		240	

Note: While BREAK is depressed, a 000 code will be generated.
When BREAK is released, an indeterminate character will be produced.

SECTION VI

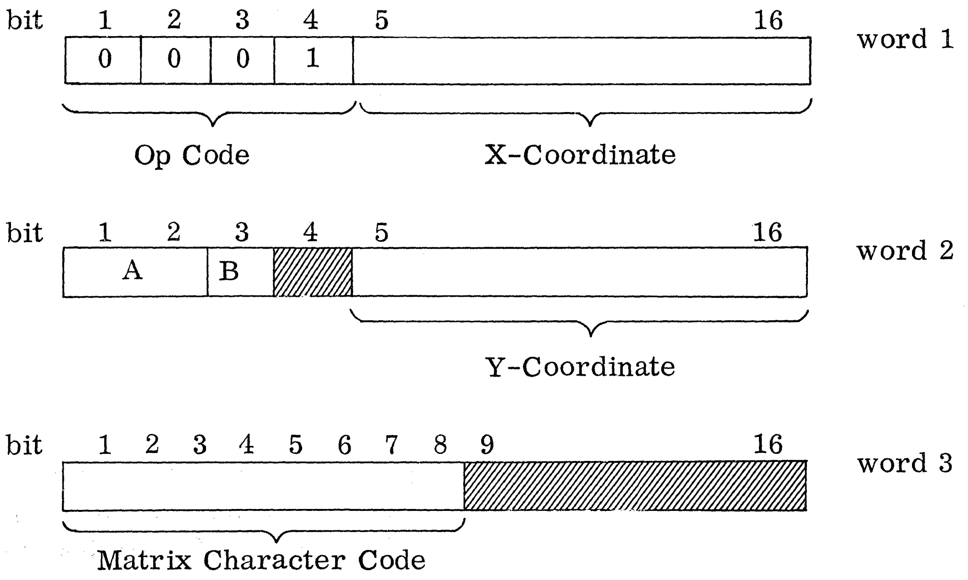
S-C 4060 GRAPHICS INSTRUCTIONS

PRINT HEAD WORD FORMATS

The following is a detailed description of the S-C 4060 print head formats. Table 6-1 summarizes these formats. Appendix A, "CRT Programming Specifications", discusses the details of line and character plotting.

Plot Specified Point

This instruction will plot the character specified by the character code in word 3 (see Table 6-2 for character Matrix Codes) at the point (location) specified in words 1 and 2 with the size and orientation specified in word 2.

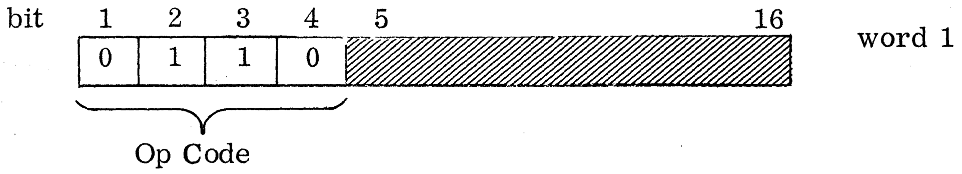


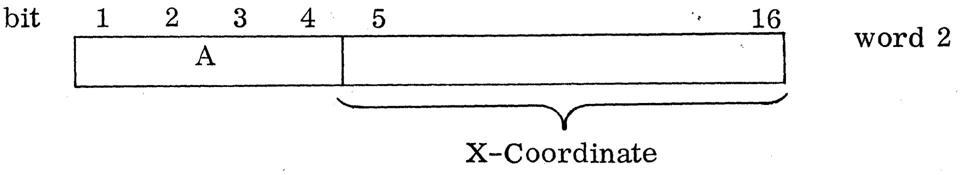
A =	00	Normal size character
=	01	Small size character
=	10	Medium size character
=	11	Large size character
B =	0	Vertical Character (Normal)
=	1	Horizontal Character (Rotated)

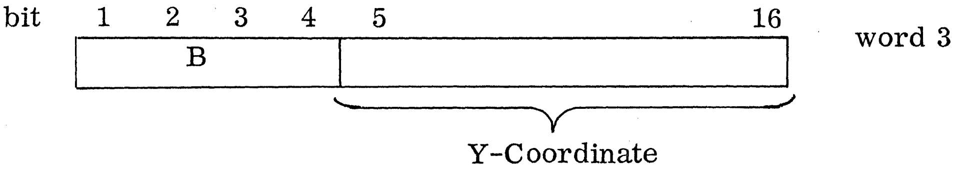
Bit position 4 of word 2 and bit positions 9 thru 16 of word 3 are not used.

Fast Plot

This instruction sets the fast plot mode in word 1. The high order 4 bits of the character code is in word 2 and the low order 4 bits is in word 3. The plotting location is also given in words 2 and 3. Subsequent characters and locations must be structured according to words 2 and 3. The code '17 in bit positions 1 thru 4 of word 2 will terminate the fast plot mode. At least one character must be specified when entering the fast plot mode.





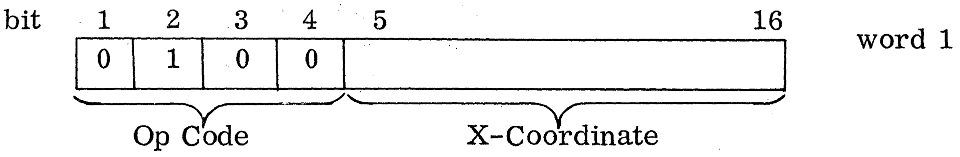


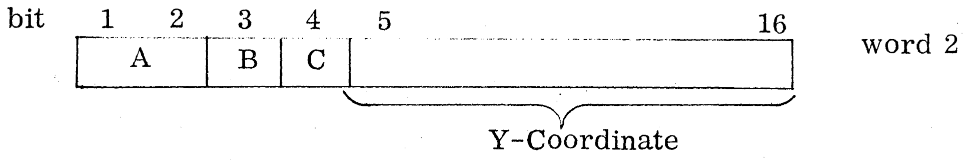
A = High Order 4 Bits of Matrix Character Code
B = Low Order 4 Bits of Matrix Character Code

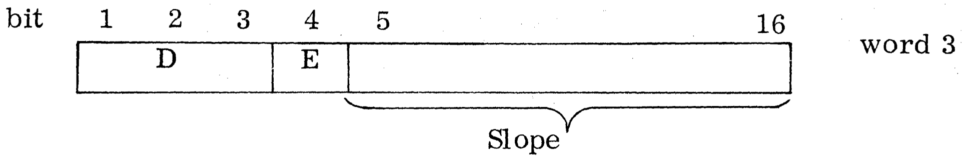
Bit Positions 5 thru 16 of word 1 are not used.

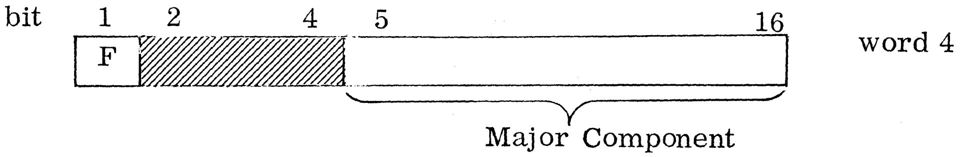
Draw Vector

This instruction will draw a line starting at the location specified in words 1 and 2. The horizontal component is specified by ΔX and the vertical as ΔY. If $\Delta X > \Delta y$, the slope required by word 3 must be $(\Delta Y \cdot 4095) / \Delta X$. If $\Delta Y > \Delta X$, then the slope must be $(\Delta X \cdot 4095)/\Delta Y$.









A	=	00	Normal Line Width
	=	01	Small Line Width
	=	10	Medium Line Width
	=	11	Large Line Width
B	=	0	for ΔX > ΔY (ΔX in word 4; bit positions 5-16)
	=	1	for ΔY > ΔX (ΔY in word 4; bit positions 5-16)
C	=	0	for + ΔX
	=	1	for - ΔX
D	=	000	Solid Line
	=	001	Short Dashed Line
	=	011	Normal Dashed Line
	=	101	Medium Dashed Line
	=	111	Large Dashed Line
E	=	0	for + ΔY
	=	1	for - ΔY
F	=	0	Standard Line Density
	=	1	Light Line Density

Bit positions 2 thru 4 of word 4 are not used.

Stroke Write (Optional)

This instruction will draw a stroke character starting at the location specified in words 1 and 2. The height adjust bit is used to increase the height of certain characters which otherwise would be out of alignment. The size bit determines the overall size of the character. A short settle time may be selected if the stroke character is located within 80 plotting positions from the previous stroke character. Any number of strokes may be used as required. The stroke designator must be marked to indicate the last stroke word.

bit	1	2	3	4	5 … 16	
	0	1	0	1		word 1
	Op Code				X-Coordinate	

bit	1	2	3	4	5 … 16	
	▨	A	B	C		word 2
					Y-Coordinate	

bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	▨	U	S			S			H	U	S			S			word 3 and subsequent
				ΔX			ΔY					ΔX			ΔY		

A	=	0	Normal Settle Time
	=	1	Fast Settle Time
B	=	0	Normal Size
	=	1	Large Size
C	=	0	Normal Height
	=	1	Increased Height
U	=	0	Blank (Stroke not recorded)
	=	1	Unblank (Record stroke)
S	=	0	+ ΔX or + ΔY
	=	1	- ΔX or + ΔY
H	=	0	Continue
	=	1	Terminate Stroke Mode

ΔX = 00
= 01 X-Component of Stroke
= 10

ΔY = 00
= 01 Y-Component of Stroke
= 10

Bit position 1 of words 2 and 3 are not used

Control Functions

This single word instruction provides for the selection of combinations of print head control commands. Generally, a control function may be given alone or in combination with other control functions. The exceptions are:

1. VOID FRAME must be accompanied with a FRAME ADVANCE.
2. GENERATE LEADER and DELAY must be given alone.

bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	0	0	0	A		B	▨	▨	C	D	E				

Op Code

A = 00 No Change in FORM FLASH Mode
= 10 Single FORM FLASH

B = 0 No Action
= 1 64 millisecond DELAY

C = 0 No Action
= 1 VOID FRAME Strobe

D = 0 No Action
= 1 GENERATE LEADER

E	=	10000	2 pole (4 perf)	FRAME ADVANCE
	=	10001	3 pole (6 perf)	FRAME ADVANCE
	=	10010	4 pole (8 perf)	FRAME ADVANCE
	=	10100	5 pole (10 perf)	FRAME ADVANCE
	=	11000	6 pole (12 perf)	FRAME ADVANCE

Bit positions 8 and 9 are not used

Special Functions

bit	1	2	3	4	5	6	7	8	9	16
	Op Code				A				▨	

Op Code

'13	A	=	0100	CLEAR BLOCK MODE
		=	1000	Start FILM PROCESSOR
		=	0010	NOTCH FILM
'14	A	=	0000	Set BLOCK MODE
'15	A	=	0000	6 perf. film advance - no hard copy
		=	1101	4 perf. film advance - strip chart
		=	0110	6 perf. film advance - 8 1/2 x 11 H.C.
		=	0100	6 perf. film advance - 11 x 14 H. C.

Op Code '15 represents the EXPOSE HARDCOPY command
It must be followed by the DELAY command.

'16	A	=	0000	TEST MODE

Op Code '16 will reset a previous TEST MODE command

Bit positions 9 thru 16 are not used

Table 6-1. Summary of Print Head Commands

Operation	Function
PLOT SPECIFIED POINT	Plots any matrix character at the raster coordinates (point) specified, in any of four sizes and with either of two orientations.
FAST PLOT	Plots any matrix character at the point specified The characteristics of this command are: 1. Uses the two smallest character sizes only. 2. Any character must be plotted within 80 raster units of the previous plotted character. The first character plotted with this command should be a blank. 3. Character size and orientation for this mode is established by first plotting a blank with the Plot Specified Point Command.
DRAW VECTOR	Draws a line from any point on the raster, with a given slope either solid or dashed in any of four line widths and in either of two line densities. The slope is found by multiplying the smallest vector component by 4095 and dividing by the largest component. Light density is not recommended for the smallest line width. A zero length vector will result in a plotted blank.
STROKE WRITE (Optional)	Draws a stroke generated character at the point specified. A full discussion of stroke write is given in Stromberg-Carlson document #9500209. "S-C 4060 Stored Program Recording System - Description and Specifications "
FORM FLASH*	Causes a form slide to be projected and recorded on film either singly or concurrent with each frame advance.

*May be included in same command

Table 6-1. Summary of Print Head Commands (cont)

Operation	Function
VOID FRAME*	Exposes the film frame to a light source which causes a circular spot near the top of the frame. This command must be accompanied by a frame advance.
GENERATE LEADER	Advances leader through camera adequate for last exposed frame to reach hard copy station. Recording is delayed until the leader is generated.
FRAME ADVANCE*	Causes the film to be advanced through the camera. The distance advanced may be 2, 3, 4, 5, or 6 poles. For 35mm perforated film, this is 4, 6, 8, 10, or 12 perforations.
DELAY	Delays input to the print head for a period of 64 milliseconds.
BLOCK MODE	Conditions how much film the film processor will do once started. If a film notcher is installed, the processor will develop film continuously until a notch is encountered in the edge of the film. If a notcher is not installed, the processor will develop 242 perfs of film. In either case the processor will stop on input loop empty if that occurs first.
RESET BLOCK MODE	Clears a previously set Block Mode. Film Processor when started will run to input loop empty.
START PROCESSOR	Starts the film processor. How much film will be processed is determined by whether the machine is in Block Mode or not.
NOTCH FILM	Activates an electromechanical film notcher required for Block Mode synchronization. The notch may also be used for hard copy sync. The notch will be placed on the upper edge of the film.
EXPOSE HARDCOPY	Exposes the upper edge of the film to a coded dot pattern required by the hardcopy processing station. The pattern is interpreted to produce strip chart, 11 x 14 or 8 1/2 x 11 paper feed, or, if no pattern is exposed, no hard copy. This command must be followed by the delay command to allow for the code light strobe time.
TEST MODE	Bypasses the print head interlocks to enable S-C 4060 operation in a non-monitored condition.

*May be included in same command

SECTION VII

PRINT HEAD OPERATIONS

Functional Characteristics

The S-C 4060 print head logic utilizes Direct Multiplex Control (DMC) channel 2 of the PCU. Print head commands, as described in Section VI, consist of a series of one or more 16-bit words and are delivered to the print head in block form. Block transfers to the print head are made from the address stored in location '22 at the address stored in location '23. At the completion of a transfer, location '22 holds the final address from which data was transferred, plus 1. Addresses stored in locations '22 and '23 define the DMC transfer limits and must not exceed '17777.

The time required to transfer a 16-bit word to the print head is a function of DMC transfer time. The time required to execute a function on the print head is a function of print head response and may vary from 2 microseconds to 112 milliseconds depending upon the graphic function or control function to be performed.

Basic Print Head Instructions

The following instructions are used to control the print head operations:

SKS '060 Skip if Print Head Ready

This instruction causes a skip of one instruction of the print head is ready to receive the next word. This instruction may be given at any time.

OCP '060 Enable Print Head DMC Channel

This instruction starts the transfer of data to the print head through DMC channel 2.

OCP '0260 Clear Print Head

This instruction will perform the following:

1. Stops the generation of leader (if in progress)
2. Clears form flash and frame advance failures.
3. Resets DMC Flip-Flop.

This instruction performs the same function as depressing the MASTER CLEAR button on the PCU console except that it will not clear block mode or automatic form flash.

INA '1160 Access Print Head Status Word

This instruction transfers the PHSW to the A-register, and resets the processor and frame advance interrupt flip-flops.

Print Head Status Word (PHSW)

A 16-bit print head status word is available to the programmer (INA '1160). Table 7-1 shows an analysis of the PHSW.

The PHSW may be accessed at any time but the examination if it is recommended at the beginning of a run and prior to and succeeding certain instructions. The conditions sensed by the PHSW are as follows:

A. Print Head Ready. This condition results from:

1. Power supply on and operative
2. Mirror in position
3. Normal mode (not test mode)
4. Film available
5. Deflection enabled
6. Ready button d epressed

B. Tube Orientation. This condition indicates the orientation of the CHARACTRON tube (see appendix A)

C. Film Low. This condition warns that the length of unexposed film available is critically short.

D. Film Processor Ready. This condition results from:

1. Film drive on
2. Chemical pumps on
3. Manual switch off
4. Chemicals at operating temperature

This condition is indicated at "not ready" if the film processing option is not included.

E. Film Processor in Position. This condition indicates the availability of the film processor. Film processor 'not in position' will be indicated if the film processing option is not included.

F. Hardcopy Ready. This condition results from:

1. Paper available
2. Hardcopy processor power on
3. Film threaded through the hardcopy exposure station

This condition will be indicated as 'not ready' if the hardcopy option is not included.

G. Paper Low. This condition warns that the quantity of unexposed paper available is critically low 'Paper Low' will be indicated if the hardcopy option is not included.

H. Input Loop Empty. This condition indicates a minimum length of exposed film between the camera and the film processor. This condition will be indicated if the film processing option is not included

J. Output Loop Empty. This condition indicates a minimum length of processed film between the film processor and the hardcopy exposure station. This condition will be indicated if the hardcopy option is not included

K. Film Processor Advance Interrupt. A 4 perforation film advance through the film processor will cause an interrupt signal to be sent to the PCU. This condition may be examined whether or not the PCU has been enabled for the interrupt. The condition is reset when the PHSW is accessed (INA '1160).

To enable the PCU for this and a camera advance interrupt, the following instruction sequence may be used:

```
LDA = '20000
OTA '0020
ENB
```

If the PCU is not enabled for this interrupt, a test for this condition may be made within approximately one-half second after the last film processor advance is executed.

L. Camera Advance Interrupt. A single film pull through the camera will cause an interrupt signal to be sent to the PCU. This condition is reset and enabled simultaneously with the Film Processor Advance Interrupt. If the PCU is not enabled for this interrupt a test for this condition may be made with 16 milliseconds after the frame advance is executed.

M. Test Mode. This condition indicates the operational mode of the S-C 4060.

Table 7-1. Print Head Status Word

Bit	Value	Condition	Bit	Value	Condition
1	0	Print head ready	9	1	Form flash failure
	1	Print head not ready		0	Form flash normal
2	1	CRT in normal position	10	1	Input loop empty
	0	CRT in rotated position		0	Input loop not empty
3	0	Film low	11	1	Output loop empty
	1	Film not low		0	Output loop not empty
4	0	Film processor ready	12	1	Film processor advance interrupt
	1	Film processor not ready		0	No film processor advance interrupt
5	1	Film processor in position	13	1	Camera advance interrupt
	0	Film processor not in position		0	No camera advance interrupt
6	0	Hardcopy ready	14	-	Not used-always = 0
	1	Hardcopy not ready			
7	1	Paper low	15	-	Not used-always = 0
	0	Paper not low			
8	-	Not used-always = 0	16	1	S-C 4060 in test mode
				0	Not in test mode

SECTION VIII

SAMPLE PROGRAMS

Fixed Point, Double Precision Add Subroutine

```
*
* THE DOUBLE PRECISION AUGEND MUST BE IN THE COMBINED A AND B REGISTERS
* ON ENTRY TO THIS SUBROUTINE.  THE LOCATION OF THE MOST SIGNIFICANT
* WORD OF THE TWO-WORD ADDEND IS SPECIFIED IN THE LOCATION FOLLOWING THE
* CALL TO THIS SUBROUTINE.  THE DOUBLE PRECISION SUM WILL BE LEFT IN THE
* COMBINED A AND B REGISTERS ON RETURN.
*
     SUBR  DADD
     REL                      RELOCATABLE PROGRAM
DADD DAC   **                 RETURN ADDRESS
     STA   COMM               SAVE HIGH ORDER A
     LDA*  DADD               ENTER ADDRESSES OF YH AND YL
     STA   COMM+1             ADDRESS OF YH
     AOA                      ADDRESS OF YL
     STA   COMM+2
     LDA   COMM               YH
     IAB                      YL
     ADD*  COMM+2             YL+XL
     CSA                      COPY SIGN TO CBIT SSP
     IAB                      YH TO A REGISTER
     ACA                      ADD C TO A
     SRC                      CHECK FOR OVERFLOW
     JMP   DADO               YES
     ADD*  COMM+1             IF NOT XH YH
     SSC                      CHECK OVERFLOW
DADX IRS   DADD               INCREMENT FOR NORMAL RETURN
DADZ IRS   DADD               ERROR RETURN
     JMP*  DADD
DADO ADD*  COMM+1             ADD YH
     SSC
     JMP   DADZ               ERROR RETURN
     JMP   DADX               NORMAL RETURN
COMM BSS   3                  STORAGE
     END   DADD
```

Fixed Point, Double Precision Subtract Subroutine

```
* THE DOUBLE PRECISION MINUEND MUST BE IN THE COMBINED A AND B REGISTERS
* ON ENTRY TO THIS SUBROUTINE.  THE LOCATION OF THE MOST SIGNIFICANT
* WORD OF THE TWO-WORD SUBTRAHEND IS SPECIFIED IN THE LOCATION FOLLOWING
* THE CALL TO THIS SUBROUTINE.  THE DOUBLE PRECISION DIFFERENCE WILL BE
* LEFT IN THE COMBINED A AND B REGISTERS ON RETURN.
*
     SUBR  DSUB
     REL                       RELOCATABLE PROGRAM
DSUB DAC   **
     STA   COMM                SAVE A
     LDA*  DSUB                ENTER ADDRESS OF YH AND YL
     STA   COMM+1
     AOA
     STA   COMM+2
     LDA   COMM                RESTORE A
     IAB
     SUB*  COMM+2              XL-YL
     CSA                       SIGN TO C.0 TO A1
     IAB
     SRC                       CHECK CARRY
     SUB   =1                  SUBTRACT 1 IF CARRY
     SRC                       CHECK FOR OVERFLOW
     JMP   DSUA
     SUB*  COMM+1              XH YH
     SSC                       CHECK OVERFLOW
DSUN IRS   DSUB                NORMAL RETURN
DSUE IRS   DSUB                ERROR RETURN
     JMP*  DSUB                RETURN
DSUA SUB*  COMM+1              OVERFLOW ON HI-ORDER DUE
     SSC                       TO CARRY PRIOR TO ADD
     JMP   DSUE
     JMP   DSUN                OVERFLOW CORRECT RESULT
*    STORAGE AREAS
COMM BSS   3
     END   DSUB
```

Fixed Point, Single Precision Multiply Subroutine

```
* THE SINGLE PRECISION MULTIPLICAND MUST BE IN THE B REGISTER ON ENTRY
* TO THIS SUBROUTINE.  THE LOCATION OF THE MULTIPLIER IS SPECIFIED IN
* THE LOCATION FOLLOWING THE CALL TO THIS SUBROUTINE.  THE DOUBLE LENGTH
* PRODUCT IS LEFT IN THE COMBINED A AND B REGISTERS ON RETURN.
*
     SUBR  MPY
     REL                     RELOCATABLE PROGRAM
MPY  DAC*
     SMI                     CHECK SIGN OF MULTIPLIER
     JMP   MPYB              POSTIVE
     TCA                     NEGATIVE TWO,S COMPLEMENT
     IAB                     MULTIPLIER IN B REGISTER
     LDA*  MPY               LOAD MULTIPLICAND
     SMI                     CHECK SIGN
     JMP   MPYC              POSTIVE
     TCA                     NEGATIVE TWO,S COMPLEMENT
MPYA JST   MPYS              EXIT TO PERFORM MULTIPLICATION
     IRS   MPY               INCREMENT FOR RETURN
MPAA JMP   MEXT              EXIT
MPYB IAB                     PLACE MULTIPLIER IN B REG
     LDA*  MPY
     SMI                     CHECK SIGN OF MULTIPLICAND
     JMP   MPYA              MULTIPLICAND PLUS, GO TO MULTIPLY
     TCA
MPYC JST   MPYS              NEGATIVE-2'S COMPLEMENT RESULT
     RCB                     RESET C BIT
     CMA                     ONE'S COMPLEMENT HI-ORDER
     IAB
     TCA                     TWO,S COMPLEMENT LOW ORDER
     SNZ                     IS RESULT ZERO
     SCB                     INSERT 1 FOR CARRY IN
     SSP                     RESET MSB TO ZERO
     IAB                     HI-ORDER TO A, LOW ORDER TO B
     ACA
     IRS   MPY               INCREMENT FOR RETURN
MPCC JMP   MEXT              EXIT
```

Fixed Point, Single Precision Multiply Subroutine (Cont)

```
*    MULTIPLICATION SUBROUTINE FOR POSTIVE VALUES
MPYS DAC    **                  ENTRANCE
     STA    COMM                STORE MULTIPLICAND
     IAB                        MULTIPLIER IN A MULTIPLICAND IN B.
     STA    COMM+1              STORE MULTIPLIER
     ERA    COMM                IF NEG. MINUS LARGEST NEGATIVE NUMBER
     SPL                        IF ZERO NORMAL OR BOTH ARE LARGEST NEG NO.
     JMP    MLNN                LARGEST NEG NO. IS PRESENT. MULT. IN B
     IAB                        TEST MULTIPLICAND FOR LNN. IF NEG
     SMI                        BOTH OPERANDS ARE LNN CK FOR BOTH 1.00000.
     JMP    *+2
     JMP    MPYN                RESULT WILL BE ZERO LNN*LNN
*                               IN A
     LDA    COMM+1              NORMAL. MULTIPLY OPERANDS LOAD MULTIPLIER
     IAB                        MULTIPLIER IN B
MPYR LDA    =-15                LOAD COUNTER
     STA    COMM+1              LOOP COUNTER
     CRA                        CLEAR A REGISTER
     LRL    1                   SHIFT TO RIGHT
     SRC                        EXAMINE RIGHT MOST BIT
     ADD    COMM                ADD MULTIPLICAND
     IRS    COMM+1              CONTROL LOOP
     JMP    *-4                 RECYCLE
     LRL    1
     LGL    1                   PUT 0 IN SIGN ORDER OF A REGISTER
     LRL    1                   MOVE 0 TO SIGN OF B REGISTER
     JMP*   MPYS
MPYN CRA                        RETURN ZERO RESULT
     IAB                        ZERO IN B
     CRA                        ZERO IN A
     JMP    MPYA+1              RESULT CANNOT BE REPRESENTED ABORT WITH -1.
*    MULTIPLY BY LARGEST NEGATIVE NUMBER
*    IF RESULT IS NEGATIVE AND THE
*       1. MULTIPLICAND IS LNN. RESULT IS THE TWO'S
*          COMPLEMENT OF THE MULTIPLIER
*       2. MULTIPLIER IS LNN. RESULT IS THE TWO'S
*          COMPLEMENT OF THE MULTIPLICAND
*    IF RESULT IS POSITIVE AND THE
*       1. MULTIPLICAND IS LNN. RESULT IS THE
*          MULTIPLIER
*       2. MULTIPLIER IS LNN. RESULT IS THE MULTIPLICAND
MLNN CRA                        CLEAR B
     IAB
     SMI                        CHECK MULTIPLICAND = MINUS LARGEST NEG. NO.
     JMP*   MPYS                EXIT WITH MULTIPLICAND IN A MULT. IS LNN
     LDA    COMM+1              MULTIPLICAND IS LNN. LOAD MULTIPLIER IN A
     JMP*   MPYS                EXIT. MULTIPLICAND IS LNN
MEXT STA    COMM                SAVE RESULT EXIT ROUTINE
     LDA    MPY                 LOAD ENTRY
     ANA    ='37777             GET RID OF INDIRECT BIT
     STA    COMM+1              STORE RETURN
     LDA    COMM                RESTORE RESULT
     JMP*   COMM+1              RETURN
*    STORAGE USED
COMM BSS    2
     END    MPY
```

Fixed Point, Single Precision Divide Subroutine

```
* THE DOUBLE LENGTH DIVIDEND MUST BE IN THE COMBINED A AND B REGISTERS
* ON ENTRY TO THIS SUBROUTINE.  THE LOCATION OF THE DIVISOR IS SPECI-
* FIED IN THE LOCATION FOLLOWING THE CALL TO THIS SUBROUTINE.  IF THE
* DIVIDEND IS GREATER THAN OR EQUAL TO THE DIVISOR, DIVISION IS NOT
* ATTEMPTED AND CONTROL IS RETURNED TO THE LOCATION OF THE CALL PLUS
* TWOO OTHERWISE, THE 16 BIT QUOTIENT WILL BE FORMED IN THE A REGISTER
* AND THE 16 BIT REMAINDER IN THE B REGISTER.  THE QUOTIENT AND THE
* REMAINDER WILL HAVE THE SAME SIGN.  CONTROL WILL BE RETURNED TO THE
* LOCATION OF THE CALL PLUS THREE.
*
      SUBR  DIV
      REL                     RELOCATABLE PROGRAM
DIV   DAC*
      SMI                     CHECK SIGN OF DIVIDEND
      JMP   DIVB
      RCB                     RESET C BIT
      CMA                     ONE'S COMPLEMENT HI-ORDER
      IAB
      TCA                     TWO'S COMPLEMENT LOW ORDER
      SNZ                     IF LOWER ORDER=0
      SCB                     MUST TWO'S COMPLEMENT HI-ORDER
      SSP                     SET B SIGN PLUS
      IAB
      ACA
      SMI                     IS NUMERATOR LARGEST NEGATIVE NUMBER
      JMP   *+2               NO CONTINUE , NORMAL
      JMP   DIVZ              ERROR EXIT ILLEGAL DIVIDE
      STA   COMM+1
      LDA*  DIV               CHECK SIGN OF DIVISOR
      SMI
      JMP   DIVC
      TCA                     TWO'S COMPLEMENT DIVISOR
      SMI                     TEST FOR LARGEST NEGATIVE NUMBER
      JMP   DIVA              DIVIDE DIVIDEND , DIVISOR SAME SIGN
      LDA   COMM+1            RESULT POSITIVE LOAD NUMERATOR ALREADY COMP
SWAP  IAB

      JMP   DIVX
DIVA  JST   DIVS              DIVIDE IF DIVISOR AND DVD SAME SIGN
DIVX  IRS   DIV               INCREMENT FOR NORMAL RETURN
      IAB                     QUOTIENT IN A REG, REMAINDER IN B
DIVZ  IRS   DIV               ERROR EXIT
DEEX  JMP   DEXT              GO TO EXIT
DIVB  STA   COMM+1            STORE MOST SIGNIFICANT HALF OF DIVIDEND
      LDA*  DIV               CHECK SIGN OF DIVISOR
      SMI                     COMPLEMENT ON NEGATIVE
      JMP   DIVA              DIVIDE OPERANDS ARE OF SAME SIGN
      TCA                     TWOS COMPLEMENT
DIVC  SMI                     CHECK DENOMINATOR FOR LARGEST NEGATIVE NO.
      JMP   DIVN              NOT LNN
      LDA   COMM+1            RESULT SIGN WILL BE NEGATIVE
      IAB                     RESULT IS NUMERATOR
      JMP   *+2               BY PASS DIVIDE COMPLEMENT RESULT
DIVN  JST   DIVS              DIVIDE
*     RETURN HERE ON NEGATIVE RESULT
      IAB                     COMPLEMENT QUOTIENT AND REMAINDER
      TCA
      IAB
      TCA
      JMP   DIVX              EXIT
```

```
*     DIVIDE ROUTINE WITH HIGH ORDER DIVIDEND IN
*     COMMON +1,LOW ORDER DIVIDEND IN B REG AND
*     DIVISOR IN A REGISTER
DIVS HLT
     STA   COMM                SAVE DIVISOR
     LDA   =-15
     STA   COMM+2              LOOP COUNTER
     LLL   1                   MOVE LOW ORDER LEFT 1
     LDA   COMM+1              LOAD HI ORDER
     CAS   COMM                COMPARE DIVIDEND AND DIVISOR
     JMP   DIVZ                DIVIDEND GTR THAN OR
     JMP   DIVZ                EQUAL TO GO TO ERROR RETURN
*       LOOP FOR DIVISION
DIVT LLR   1                   MOVE DIVIDEND 1 POSITION TO LEFT
     SPL                       TEST SIGN POSITION FOR SPILL
     JMP   DIVU                MINUS SUBTRACT DIVIDEND IS GREATER
     CAS   COMM                COMPARE DIVIDEND AND DIVISOR
     JMP   DIVU                IF GREATER SUBTRACT
     JMP   DIVU                EQUAL SUBTRACT
     JMP   DIVH                INDEX AND LOOP
DIVU SUB   COMM
     SSM                       PUT-1-BIT IN QUOTIENT LOOP
DIVH IRS   COMM+2              LOOP
     JMP   DIVT                LOOP
     LLR   1                   RESET SIGN BITS
     LGR   1
     JMP*  DIVS
DEXT STA   COMM                SAVE RESULT EXIT ROUTINE
     LDA   DIV                 LOAD ENTRY
     ANA   ='37777             GET RID OF INDIRECT BIT
     STA   COMM+1              STORE RETURN
     LDA   COMM                RESTORE RESULT
     JMP*  COMM+1              RETURN
*     STORAGE AREAS
COMM BSS   3
*     END OF SINGLE PRECISION DIVISION
     END   DIV
```

Output on ASR-33

```
*    THIS SUBROUTINE OUTPUTS ONE CHARACTER TO THE ASR-33.  IF THE
*    CHARACTER IS PRINTABLE, THE CHARACTER WILL BE PRINTED.  IF THE
*    ASR-33 PAPER TAPE PUNCH IS ON, THE CHARACTER WILL BE PUNCHED
*    (WHETHER PRINTABLE OR NOT).  THE SUBROUTINE IS ENTERED WITH THE
*    CHARACTER TO BE OUTPUT IN THE A REGISTER.
*
     REL
     SUBR  ASRTYP,STRT        ESTABLISH SUBROUTINE NAME 'ASRTYP'
*                             HAVING THE ENTRY POINT AT 'STRT'
*
STRT DAC   **                 SUBROUTINE ENTRY POINT
     SKS   '104               DELAY IF ASR-33 BUSY
     JMP   *-1
     OCP   '104               ENABLE ASR-33 IN OUTPUT MODE
     OTA   4                  OUTPUT CHARACTER IN ASCII MODE
     JMP   *-1                (DELAY IF ASR-33 NOT READY)
     JMP*  STRT               RETURN TO CALLING PROGRAM
     END
```

APPENDIX A

CRT PROGRAMMING SPECIFICATIONS

Standard Raster

The S-C 4060 standard raster consists of a rectangular array of 12,582,906 address able points. The number of addressable points in the vertical direction is 3072 and in the horizontal direction, 4096. It is first quadrant with the origin at (0, 512) and bounded by the corner points (0, 3583), (4095, 3583), and (4095, 512).

Points addressed in the vertical direction which lie below 512 or above 3583 will be masked off by the camera aperture and will not be recorded on film.

The S-C 4060 CRT may be physically rotated. In this case, the origin will lie at (512, 0) and the raster will be bounded by the corner points (512, 4095), (3583, 4095), and (3583, 0). See Figure A-1.

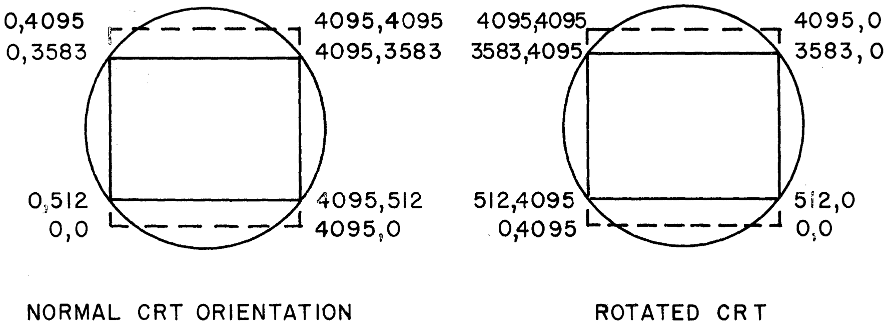


Figure A-1. CRT Orientation

Character Specifications

The standard character matrix contains 116 characters as shown in Table A-1. The physical sizes of the characters in raster units varies with the individual character and the selected size. For example, the maximum width of a normal size character is normally set to allow 31 raster units to be used for normal character spacing (10 characters per inch with 100mm lens on hardcopy).

Any character may be plotted vertically (normal) or horizontally (rotated) under programmed control.

Table A-1. Character Conversion Codes

Matrix Symbol	Octal Value	Hexadecimal Value	Matrix Symbol	Octal Value	Hexadecimal Value
A	125	55	Z	211	89
B	105	45	a	126	56
C	065	35	b	106	46
D	145	65	c	066	36
E	165	75	d	146	66
F	205	85	e	166	76
G	047	27	f	206	86
H	147	67	g	050	28
I	265	B5	h	150	68
J	245	A5	i	266	B6
K	171	79	j	246	A6
L	067	37	k	072	7A
M	107	47	l	170	38
N	127	57	m	110	48
O	305	C5	n	130	58
P	227	97	o	306	C6
Q	247	A7	p	230	98
R	267	B7	q	250	A8
S	327	D7	r	270	B8
T	307	C7	s	330	D8
U	071	39	t	310	C8
V	111	49	u	072	3A
W	131	59	v	112	4A
X	151	69	w	132	5A
Y	225	95	x	152	6A

Table A-1. Character Conversion Codes (cont)

Matrix Symbol	Octal Value	Hexadecimal Value	Matrix Symbol	Octal Value	Hexadecimal Value
y	226	96	-	263	B3
z	212	8A	\	064	34
0	232	9A	→	104	44
1	252	AA	&	124	54
2	272	BA	"	144	64
3	312	CA	(	164	74
4	113	4B	)	204	84
5	133	5B	*	224	94
6	153	6B	/	244	A4
7	173	7B	¬	264	B4
8	213	8B	—	304	C4
9	233	9B	γ	045	25
δ	122	52	}	325	D5
Π	142	62	BLANK	052	2A
?	162	72	~	326	D6
#	202	82	·	167	77
±	222	92	•	207	87
{	103	43	•	170	78
←	123	53	●	210	88
@	143	63	⌷	051	29
!	163	73	;	231	99
%	203	83	,	251	A9
,	223	93	.	271	B9
—	243	A3	:	311	C9

Table A-1. Character Conversion Codes (cont)

Matrix Symbol	Octal Value	Hexadecimal Value
β	331	D9
°	332	DA
Δ	073	3B
+	253	AB
-	273	BB
○	313	CB
α	114	4C
|	134	5C
>	154	6C
<	174	7C
□	214	8C
[	234	9C
]	254	AC
∫	274	BC
∧	115	4D
∂	135	5D
$	155	6D
¢	175	7D
=	215	8D
Σ	235	9D
‘	255	AD

Recommended character and line spacing is shown in Table A-2. A decrease in the raster values specified may result in buttered or overlapped characters.

Table A-2. Character And Line Spacing

Size	Spacing* Char.	Line	Char/Line 11 x 14	8 1/2 x 11	Lines/Page	Size Ratio To Normal
Small	24	38	171	92	81	.75
Normal	31	52	132	72	60	1.00
Medium	40	64	103	56	49	1.25
Large	48	77	86	46	40	1.50

* In raster units

Line Specifications

Lines (or vectors) may be drawn with a single beam sweep between any two points on the raster. They may be either solid or dashed.

Dashed lines will be drawn such that the solid portions and the spaces have the same length. Table A-3 specifies the dash (and space) lengths.

Four line widths (weights) are available as shown in Table A-4. The selected line width will be maintained for both solid and dashed lines.

All lines may be drawn in either of two densities; light or heavy. Heavy lines are considered normal density and should be used for all general line drawing. Light lines (or fast lines) are drawn at the rate of 250 nanoseconds per raster unit. This relatively short exposure time causes these lines to be recorded on film in a low density.

Table A-3. Standard Dashes In Raster Units

Dash Size	Fast Line	Normal Line
Short	32	8
Normal	64	16
Medium	128	32
Long	256	64

Table A-4. Standard Line Widths In Raster Units

Size	Nominal Width
Small	2*
Normal	4
Medium	8
Large	16

*Not recommended for fast lines

APPENDIX B

ASCII CODE*

Standard Code

b7 →					0	0	0	0	1	1	1	1
b6 →					0	0	1	1	0	0	1	1
b5 →					0	1	0	1	0	1	0	1
Bits b4 ↓	b3 ↓	b2 ↓	b1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	`	P	@	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0.	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SS	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	~	l	¬
1	1	0	1	13	CR	GS	–	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	|
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Character Representation

The standard 7-bit character representation, with b_7 the high-order bit and b_1 the low-order bit, is shown below.

Example. The bit representation for the character "K", positioned in column 4, row 11, is:

b_7	b_6	b_5	b_4	b_3	b_2	b_1
1	0	0	1	0	1	1

The code table position for the character "K" may also be represented by the notation "column 4, row 11" or alternately as "4/11." The decimal equivalent of the binary number formed by bits b_7, b_6 and b_5, collectively, forms the column number, and decimal equivalent of the binary number formed by bits b_4, b_3, b_2 and b_1, collectively, forms the row number.

Legend

Control Characters

NUL	Null
SOH	Start of Heading (CC)
STX	Start of Text (CC)
ETX	End of Text (CC)
EOT	End of Transmission (CC)
ENQ	Enquiry (CC)
ACK	Acknowledge (CC)
BEL	Bell (audible or attention signal)
BS	Backspace (FE)
HT	Horizontal Tabulation (punched card skip) (FE)
LF	Line Feed (FE)
VT	Vertical Tabulation (FE)
FF	Form Feed (FE)
CR	Carriage Return (FE)
SO	Shift Out
SI	Shift In
DLE	Data Link Escape (CC)
DC1	Device Control 1
DC2	Device Control 2
DC3	Device Control 3
DC4	Device Control (stop)
NAK	Negative Acknowledge (CC)
SYN	Synchronous Idle (CC)
ETB	End of Transmission Block (CC)
CAN	Cancel
EM	End of Medium
SS	Start of Special Sequence
ESC	Escape
FS	File Separator (IS)
GS	Group Separator (IS)
RS	Record Separator (IS)
US	Unit Separator (IS)
DEL	Delete

(CC) Communication Control. (FE) Format Effector. (IS) Information Separator.

Graphic Characters

Column/Row	*Symbol*	*Name*
2/0	SP	Space (normally nonprinting)
2/1	!	Exclamation Point
2/2	"	Quotation Marks (diaeresis)
2/3	#	Number Sign
2/4	$	Dollar Sign
2/5	%	Percent
2/6	&	Ampersand
2/7	'	Apostrophe (closing single quotation mark; acute accent)
2/8	(	Opening Parenthesis
2/9	)	Closing Parenthesis
2/10	*	Asterisk
2/11	+	Plus
2/12	,	Comma (cedilla)
2/13	-	Hyphen (minus)
2/14	.	Period (decimal point)
2/15	/	Slant
3/10	:	Colon
3/11	;	Semicolon
3/12	<	Less Than
3/13	=	Equals
3/14	>	Greater Than
3/15	?	Question Mark
4/0	`	Grave Accent (opening single quotation mark)
5/11	[	Opening bracket
5/12	~	Tilde
5/13	]	Closing bracket
5/14	^	Circumflex
5/15	—	Underline
6/0	@	Commercial at
7/11	{	Opening brace
7/12	¬	Overline
7/13	}	Closing brace
7/14	|	Vertical line

*The information presented is an excerpt from the proposed revised American Standard Code for Information Interchange.

APPENDIX C

SUMMARY OF STANDARD INSTRUCTIONS

(Listed in Alphabetical Order)

monic	Octal Code	Instruction	Type	Execution Time (μsec)	Page
CA	141216	Add C to A	G	1	2-3
DD	06	Add	MR	2	2-3
LR	0416	Logical Left Rotate	SH	1 + .5n	2-4
LS	0415	Arithmetic Left Shift	SH	1 + .5n	2-4
NA	03	AND to A	MR	2	2-4
OA	141206	Add One to A	G	1	2-3
RR	0406	Logical Right Rotate	SH	1 + .5n	2-5
RS	0405	Arithmetic Right Shift	SH	1 + .5n	2-5
CAL	141050	Clear A, Left Half	G	1	2-12
CAR	141044	Clear A, Right Half	G	1	2-12
CAS	11	Compare	MR	3	2-10
CHS	140024	Complement A Sign	G	1	2-4
CMA	140401	Complement A	G	1	2-4
CRA	140040	Clear A	G	1	2-3
CSA	140320	Copy Sign and Set Sign Plus	G	1	2-4
ENB	000401	Enable Program Interrupt	G	1	2-10
ERA	05	Exclusive OR to A	MR	2	2-4
HLT	000000	Halt	G		2-10
IAB	00201	Interchange A and B	G	1	2-3
ICA	141340	Interchange Characters in A	G	1	2-12
ICL	141140	Interchange and Clear Left Half of A	G	1	2-13
ICR	141240	Interchange and Clear Right Half of A	G	1	2-13

APPENDIX (Cont)

SUMMARY OF STANDARD INSTRUCTIONS
(Listed in Alphabetical Order)

Mnemonic	Octal Code	Instruction	Type	Execution Time (μsec)	P
IMA	13	Interchange Memory and A	MR	3	2
INA	54	Input to A	IO	2	2
INH	001001	Inhibit Program Interrupt	G	1	2
INK	000043	Input Keys	G	1	2
IRS	12	Increment, Replace and Skip	MR	3	2
JMP	01	Unconditional Jump	MR	1	2-
JST	10	Jump and Store Location	MR	3	2-
LDA	02	Load A	MR	2	2-
LDX	15	Load X	MR	3	2-
LGL	0414	Logical Left Shift	SH	1 + .5n	2-
LGR	0404	Logical Right Shift	SH	1 + .5n	2-
LLL	0410	Long Left Logical Shift	SH	1 + .5n	2-
LLR	0412	Long Left Rotate	SH	1 + .5n	2-
LLS	0411	Long Arithmetic Left Shift	SH	1 + .5n	2-
LRL	0400	Long Right Logical Shift	SH	1 + .5n	2-
LRR	0402	Long Right Rotate	SH	1 + .5n	2-
LRS	0401	Long Arithmetic Right Shift	SH	1 + .5n	2-
NOP	101000	No Operation	G	1	2-
OCP	14	Output Control Pulse	IO	2	2-
OTA	74	Output From A	IO	2	2-
OTK	171020	Output Keys	IO	2	2-
RCB	140200	Reset C Bit	G	1	2-1
SCB	140600	Set C Bit	G	1	2-1
SKP	100000	Unconditional Skip	G	1	2-1
SKS	34	Skip if Ready Line Set	IO	2	2-1
SLN	101100	Skip if (A_{16}) is ONE	G	1	2-1
SLZ	100100	Skip if (A_{16}) is ZERO	G	1	2-1

APPENDIX C (Cont)

SUMMARY OF STANDARD INSTRUCTIONS

(Listed in Alphabetical Order)

monic	Octal Code	Instruction	Type	Execution Time (μsec)	Page
MI	101400	Skip if A Minus	G	1	2-11
MK	74	Set Mask	IO	2	2-10
NZ	101040	Skip if A Not ZERO	G	1	2-11
PL	100400	Skip if A Plus	G	1	2-11
RC	100001	Skip if C Reset	G	1	2-12
R1	100020	Skip if Sense Switch 1 is Reset	G	1	2-12
R2	100010	Skip if Sense Switch 2 is Reset	G	1	2-12
R3	100004	Skip if Sense Switch 3 is Reset	G	1	2-12
R4	100002	Skip if Sense Switch 4 is Reset	G	1	2-12
SC	101001	Skip if C Set	G	1	2-12
SM	140500	Set Sign Minus	G	1	2-4
SSP	140100	Set Sign Plus	G	1	2-4
SSR	100036	Skip if no Sense Switch Set	G	1	2-12
SSS	10136	Skip if any Sense Switch is Set	G	1	2-12
SS1	101020	Skip if Sense Switch 1 is Set	G	1	2-11
SS2	101010	Skip if Sense Switch 2 is Set	G	1	2-11
SS3	101004	Skip if Sense Switch 3 is Set	G	1	2-11
SS4	101002	Skip if Sense Switch 4 is Set	G	1	2-12
STA	04	Store A	MR	2	2-3
STX	15	Store X	MR	2	2-3
SUB	07	Subtract	MR	2	2-4
SZE	100040	Skip if A ZERO	G	1	2-12
TCA	140407	Two's Complement A	G	1.5	2-4

High Speed Arithmetic

DAD	06	Double Precision Add	MR	3	
DBL	000007	Enter Double Precision Mode	G	1	

APPENDIX C (Cont)

SUMMARY OF STANDARD INSTRUCTIONS
(Listed in Alphabetical Order)

Mnemonic	Octal Code	Instruction	Type	Execution Time (μsec)	Pa
DIV	17	Divide	MR	11	2-
DLD	02	Double Precision Load	MR	3	2-
DSB	07	Double Precision Subtract	MR	3	2-1
DST	04	Double Precision Store	MR	3	2-1
MPY	16	Multiply	MR	5.5	2-1
NRM	000101	Normalize	G	1 + .5n	2-1
SCA	000041	Shift Count to A	G	1	2-1
SGL	000005	Enter Single Precision Mode	G	1	2-1

APPENDIX D

DEDICATED LOCATIONS

Octal Address	Assignment	
00000	Index Register	
00001 thru 00017	Protected Fill Program	
00020	Starting	Addresses for DMC Channel 1
00021	Final	
0022	Starting	Addresses for DMC Channel 2
0023	Final	
00060	Power Failure Interrupt Link	
00063	Standard Interrupt Link	