DDP 116

PROGRAMMERS REFERENCE MANUAL

3C COMPUTER CONTROL COMPANY, INC.

Document No. 71-281A

PROGRAMMERS REFERENCE MANUAL

FOR THE

DDP-116

GENERAL PURPOSE COMPUTER

February 1966

Computer Control Company, Inc.
Old Connecticut Path
Framingham, Massachusetts

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

SECTION VI
SAMPLE PROGRAMS

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

TABLE OF ILLUSTRATIONS (Cont)

LIST OF TABLES

INTRODUCTION

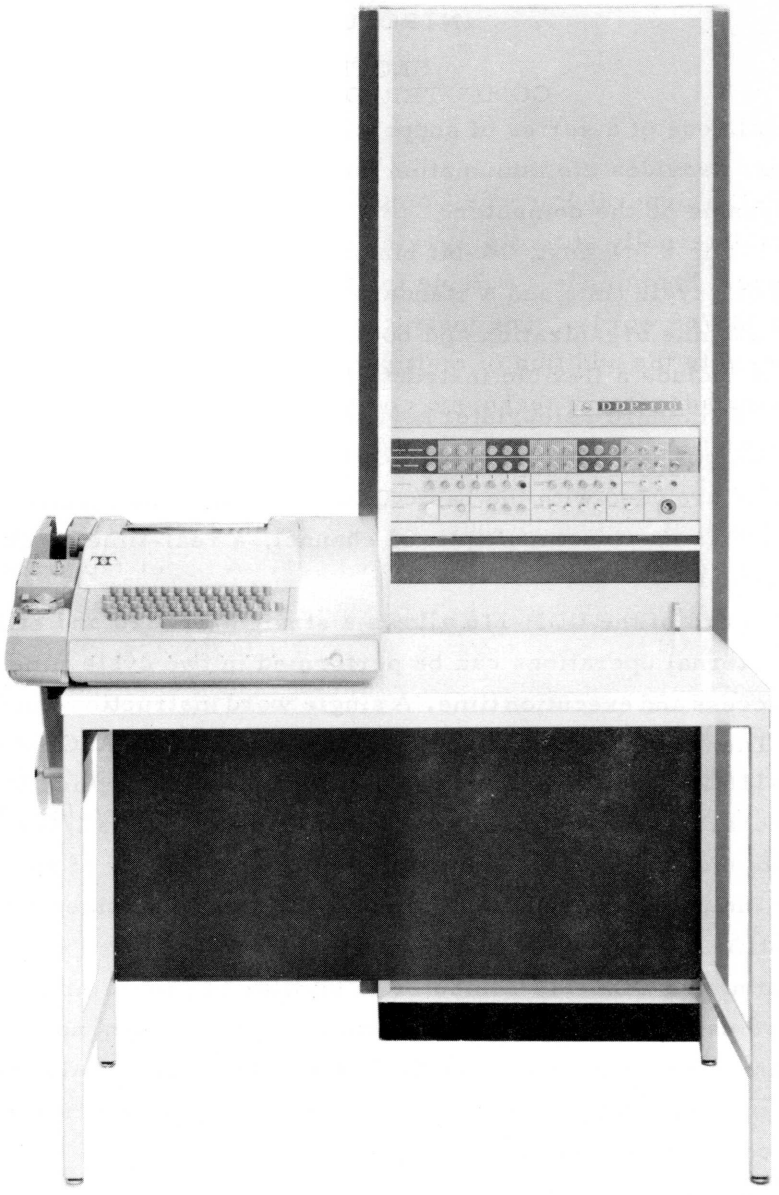
This manual, one of a series of support documents for the DDP-116 General Purpose Digital Computer, provides the information normally required by a programmer in preparing and running programs on the computer.

The DDP-116 is a low cost, 16-bit binary word general purpose digital computer with a 1.7-μsec memory cycle time and a standard 4096-word core memory. The DDP-116 has a fully parallel machine organization and both indexing and multilevel indirect addressing. Standard features include a flexible instruction repertoire of 62 commands, a powerful I/O bus structure, and standard Teleprinter keyboard and paper tape I/O unit. An extensive programming package, including a symbolic assembler, FORTRAN IV and diagnostic and utility routines, is provided with the basic DDP-116. Options include memory parity, a high-speed arithmetic option, direct multiplexed channel, a real-time clock and a full line of peripheral equipment.

The 16-bit word of the DDP-116 allows a straightforward and efficient addressing scheme. Most internal operations can be performed in two cycle times (3.4 μsec), 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.

The DDP-116 is designed for both open-shop scientific applications and real-time on-line data processing and control. Modular design, a flexible I/O structure, and command repertoire enable the DDP-116 to be tailored to a broad variety of applications both on and off line. These include data reduction, process control, instrumentation, simulation and open-shop scientific and engineering computation.

Programming the DDP-116 computer is similar to programming other single-address binary computers using two's complement notation. Therefore, no major differences confront the programmer who is new to the DDP-116. This manual also presents programming considerations in enough detail to satisfy the needs of a programmer who is new to single-address binary computers in general.



741

DDP-116 General Purpose Computer

SECTION I
COMPUTER ORGANIZATION

The DDP-116 General Purpose Computer is a solid-state, parallel, 16-bit binary machine with an internally stored program. The memory cycle time is conservatively rated at 1.7 μsec. Memory is available in 1024-, 4096-, and 8192-word modules and can be expanded to 32,768 words. The instruction repertoire includes 62 commands. An arithmetic option permits the addition of multiply, divide, and normalize hardware commands. A powerful sector addressing technique combined with large 512-word sectors permits the majority of instructions to be one word and maximizes memory efficiency. Multilevel indirect addressing and indexing are provided.

The input/output system includes a word parallel I/O bus system, an interrupt line, sense line, and 10 external function address lines. A paper tape reader, paper tape punch and a Teleprinter are provided as standard equipment. Software includes the DAP-116 assembler, FORTRAN IV, COP bugging system, I/O library, math library, and diagnostic programs. Support services include programming and maintenance training, logistic support program, maintenance services, and a user organization.

A full line of optional moderate and high-speed peripheral equipment is available. In addition, a line of compatible general-purpose logic modules (S-PACs) is available for system interface implementation.

SPECIFICATIONS

Type
Parallel binary, solid state

Addressing
Single address with indexing and indirect addressing

Word Length
16 bits

Machine Code
Two's complement

Memory Type
Coincident-current ferrite core

Memory Size
1024-, 4096-, or 8192-word basic modules expandable to 32,768 words

Memory Cycle Time
1.7 μsec

Speed

Add 3.4 μsec

Subtract 3.4 μsec

Multiply (subroutine) 243 μsec

Divide (subroutine) 336 μsec

Multiply (hardware option) 9.18 μsec maximum

Divide (hardware option) 16.67 μsec maximum

Standard Peripheral Equipment

ASR-33 Teletype Unit providing the following capabilities:

a. Read paper tape at 10 cps
b. Punch tape at 10 cps
c. Print 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

300-line-per-minute (120-character-per-line) high-speed printer

100-card-per-minute card reader

100-card-per-minute card punch

IBM compatible magnetic tape units:

Unit	Tape Speed (ips)	Density (bpi)
Low speed	45	200, 556
High speed	75	200, 556, 800

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

Four modes are available for data transfer between peripheral devices and the DDP-116.

a. Single word transfer
b. Single word transfer with priority interrupt
c. Direct multiplexed channel (DMC) (optional)
d. Direct data channel (DDC) (optional)

Interrupt

Single interrupt line standard. Optional priority interrupts are available in multiples of eight up to a maximum of 256 lines.

Circuits

Computer Control Company standard S-PAC Digital Modules

Signal Levels

Zero volt for logical ZERO; -6v for logical ONE

Dimensions

22-1/16 in. wide; 22-1/16 in. deep; 68-13/16 in. high

Weight

Approximately 500 lb (standard single-bay machine)

Cooling

Filtered, forced convection cooling provided within cabinet

Power

Single-phase, 115v ± 10v, 60 cycles

SYSTEM DESCRIPTION

Figure 1-1, a block diagram of the DDP-116, 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 one or more modules of 1024, 4096, or 8192 16-bit words. Data from the core memory is transferred to and from the DDP-116 registers through the M-register and the A- and B-buses. The DDP-116 memory can be expanded to include up to 16,384 16-bit words, or up to 32,768 words by means of the extended addressing option.

The functional units of the central processor and the input/output controls are as follows:

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 14-bit counter that contains the location of the next instruction to be executed. The program counter is incremented by one each time an instruction is performed and may be incremented additional times during the execution of conditional skip instruction. In the case of a jump, the program counter is loaded with the memory location to which the program is to jump.

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 Flip-Flop: The C-bit flip-flop, associated with the A- and B-registers, stores overflow indications occurring during the execution of arithmetic instructions, and can be tested by sense instructions in connection with shifting operations.

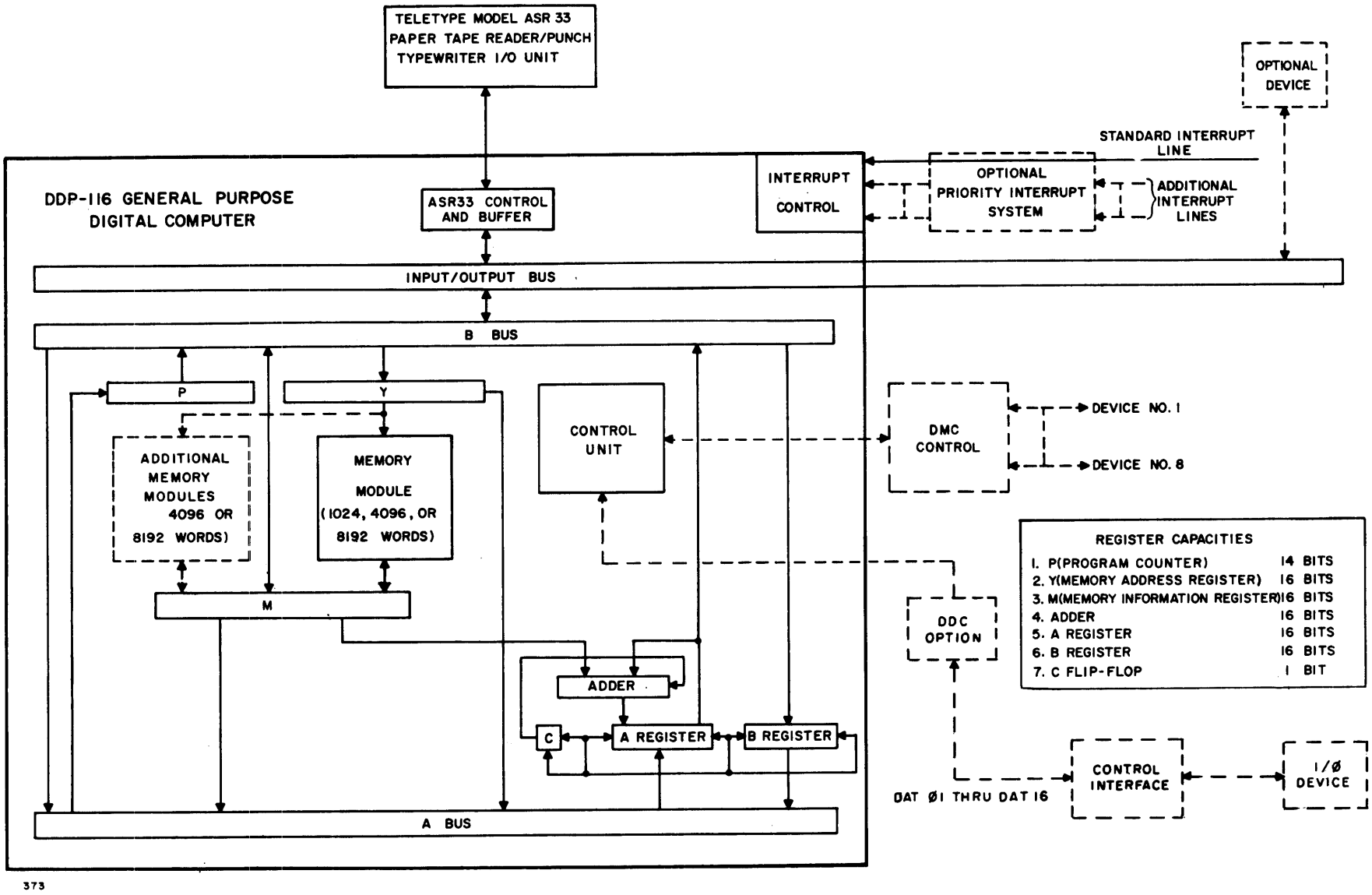


Figure 1-1. DDP-116 General Purpose Computer, Simplified Block Diagram

F-Register: A 6-bit register that contains the indirect address bit, index bit, and 4-bit operation code.

A-Bus and B-Bus: Two gating structures to which the computer registers and counters are connected both as source and destination. Transfer of information from one register or counter to another is effected by the simultaneous gating of the proper paths from one register to a bus, and from the bus to the other register.

Output Bus (OTB): Sixteen lines that transmit data from the computer B-bus to an I/O device.

Input Bus (INB): Sixteen lines that transmit data from an I/O device to the computer B-bus.

Address Bus (ADB): Ten lines used in conjunction with I/O devices. Bits on lines 1 through 4 define the function to be performed by the I/O device. Bits on lines 5 through 10 designate the I/O device to be used.

Control Clock: The control clock generates timing pulses required for the operation of the computer.

WORD FORMATS

Data Words

The format for data words stored in the DDP-116 is shown in Figure 1-2.

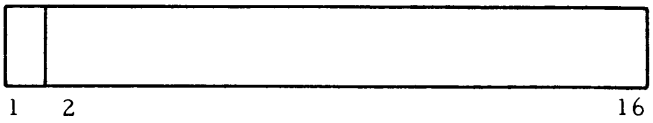


Figure 1-2. Data Word Format, Single Precision

Sixteen-bit data words are stored in two's complement form. The first bit of a data word represents the sign and is zero for positive data. Bits 2 to 16 represent the magnitude in true binary form. Positive values thus range from zero (which always has a positive sign) to 32,767 as follows:

0	000 000 000 000 000	Zero
0	000 000 000 000 001	+1
0	000 000 000 000 010	+2
	'	'
	'	'
	'	'
0	111 111 111 111 111	+32,767

Negative numbers are represented in two's complement form and always have a one in the sign bit position.

Two's Complement Arithmetic. -- The two's complement of a binary number is obtained by complementing (reversing) each bit and adding one. For example, the two's complement of +1, which represents -1, is obtained as follows:

+1	0	000 000 000	000 001
Complement	1	111 111 111	111 110
Add 1	0	000 000 000	000 001
Two's Complement (-1)	1	111 111 111	111 111

The number range for negative values is from -1 to -32,768 as follows:

1	111 111 111 111	111	(-1)
1	111 111 111 111	110	(-2)
1	111 111 111 111	101	(-3)
1	000 000 000 000	000	(-32,768)

While -1 has the largest magnitude of any negative number, if +1 is added to it, the result is zero. Thus:

1	111 111 111 111	111	-1
0	000 000 000 000	001	+1
0	000 000 000 000	000	Zero

Note that a carry bit from the most significant position has been ignored. In two's complement arithmetic, if numbers of unlike signs are added together, carries from the most significant bit are disregarded.

Overflow. -- Overflow is the condition that occurs when two number of like signs are added together to produce a sum of a different sign. For example, adding +1 to +32,767 would produce a result larger than the capacity of a single data word.

0	111 111 111 111	111	(+32,767)
0	000 000 000 000	001	(+1)
1	000 000 000 000	000	

The different sign of the result defines an overflow condition.

Addition on the DDP-116 is performed by adding a quantity in the memory to a quantity in the A-register. True signed arithmetic takes place. Overflow conditions automatically result in the setting of the C-bit indicator, even though no carry is propagated from the sign position. In the preceding example, the C-bit indicator would be set.

Double Precision. -- When further precision is required than that obtainable from a data word, 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 subroutines. An example of a double precision addition is shown in Section

Logical Data. -- Logical data, such as the condition of sixteen binary switches, 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 two types: non-memory reference and memory reference.

The basic instruction word format in the DDP-116 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, the instruction is identified as an add instruction; if they contain 1011, 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 DAP, the DDP-116 assembly language. The table below shows the op codes expressed in binary, octal, and mnemonically for representative instructions of each type.

			Operation Code	
Instruction	Type	Binary	Octal	Mnemonic
Subtract	Memory Reference	x x01 11x xxx xxx xxx	07	SUB
Input to A	Input/Output	1 011 00x 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

Certain configurations in bits 3 to 6 are interpreted to signify not a single instruction but a group of instructions. Further decoding of other bits is necessary to uniquely identify the instruction. Instead of using 4 bits to identify the operation code (as in the memory reference instruction) 6, 10, or all 16 bits are utilized for identification. These operation codes identify non-memory reference instructions.

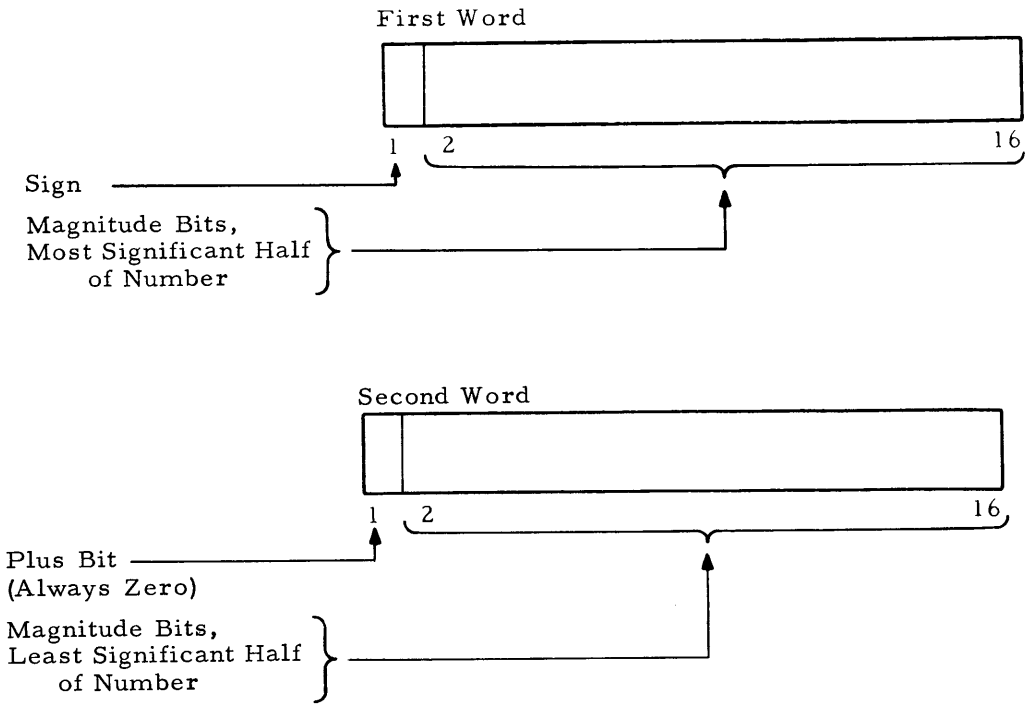


Figure 1-3. Data Word Format, Double Precision

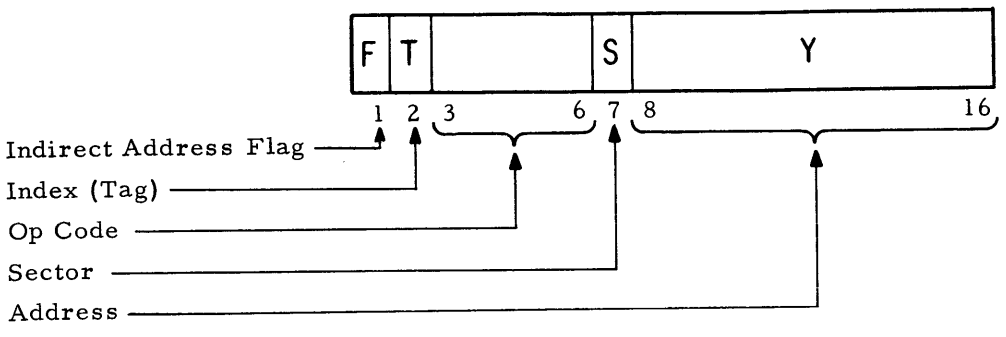


Figure 1-4. Memory Reference Instruction Format

Non-Memory Reference Instructions. -- The non-memory reference instructions are subdivided into three types, each type having its own unique format. These types are:

a. Input-output and test (6-bit op code)
b. Shift (10-bit op code)
c. Generic (16-bit op code)

Their formats are shown in Figure 1-5.

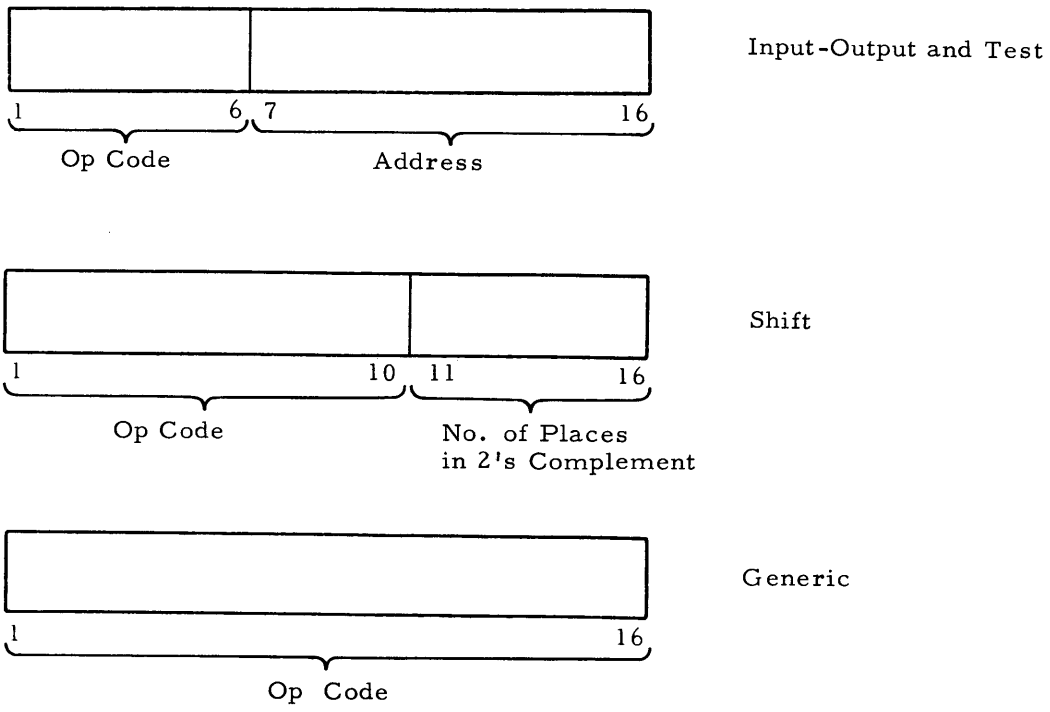


Figure 1-5. Formats of Non-Memory Reference Instructions

Memory Reference Instructions. -- The DDP-116 is a single address computer and can access one operand in memory with each instruction. For example, to add the contents of memory address X to the contents of memory address Y and put the sum in memory address Z requires three instructions, one to address each of the three factors. The program to do this is expressed mnemonically as

```
LDA  X
ADD  Y
STA  Z
```

The first instruction loads the A-register with the data in address X, the second adds the data in address Y to it, and the third stores the contents of the A-register (the sum) in location Z.

Memory reference instructions in the DDP-116 (see Figure 1-4) are extremely flexible in their addressing capability and can directly reference large portions of memory in a single cycle. All the bits in the instruction except for the op-code bits are utilized in various forms of addressing. These bits are:

a. Bit 1, indirect address bit (FLAG)
b. Bit 2, index bit (TAG)
c. Bit 7, sector bit
d. Bits 8-16, address bits

The use of these bits is described after the general description of memory addressing.

MEMORY ADDRESSING

Several techniques are used in the DDP-116 for memory addressing: sector addressing, indexing and indirect addressing.

Sector Addressing

The memory of the DDP-116 is considered to be divided into sectors of 512 words each. A 4096-word computer will have 8 sectors; and 8192-word computer, 16. 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 000-777).
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 in two sectors of memory, half of which are in sector 0 and half of which are determined by the location of the instruction. Figure 1-6 represents the memory that can be directly addressed by an instruction in sector 2 and an instruction in sector 6.

Indexing

Further addressing flexibility is implemented on the DDP-116 through the use of indexing. The index "register" is a 16-bit word in memory location 0 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).

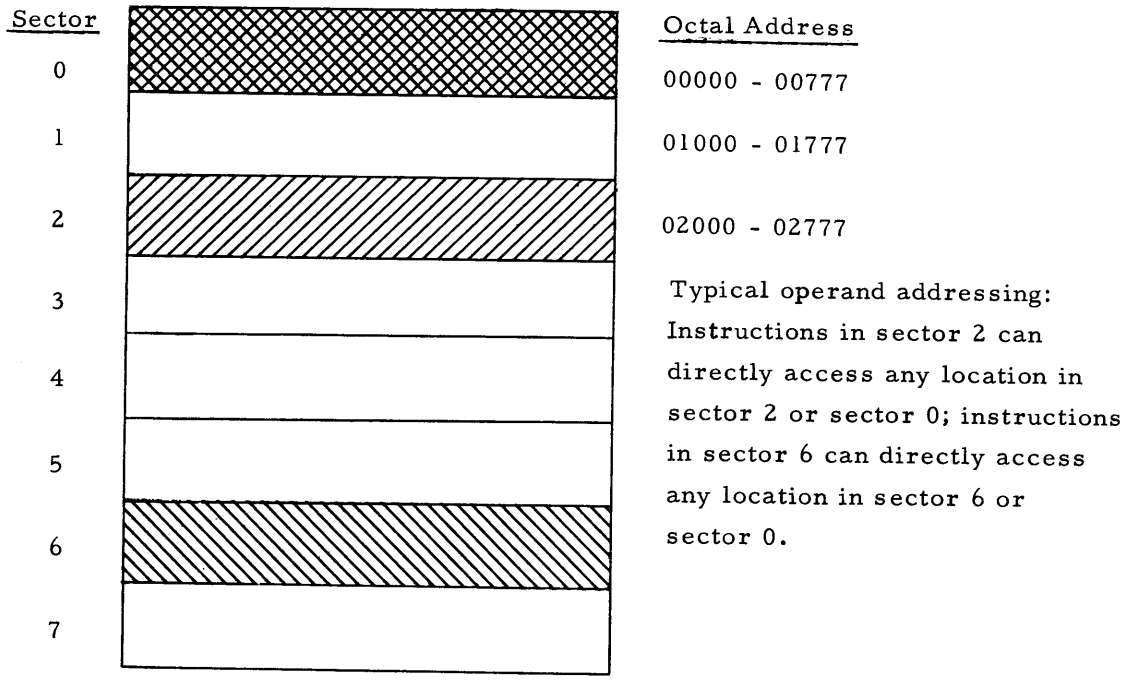


Figure 1-6. Memory Sectors in 4096-Word DDP-116

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 1-7.

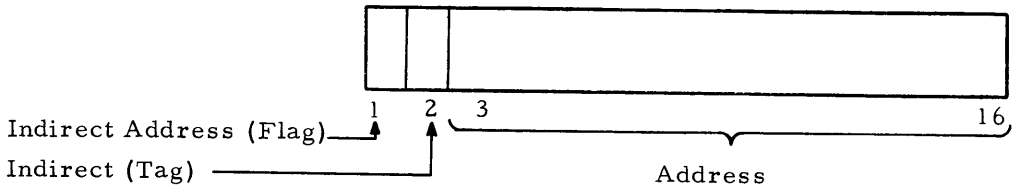


Figure 1-7. Indirect Address Format

To illustrate indirect addressing, consider that an add command in sector 2 is flagged for indirect addressing (this is specified in DAP 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. Its contents 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.

Indirect addressing thus enables an instruction to address 16K of memory. With the memory extension option 116-15, indirect addressing of 32K memory is possible.

INSTRUCTION LOGIC AND TIMING

Figure 1-8 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 first 6 bits are transferred from M into the F-register.

Assuming that the instruction is a memory reference instruction such as an add, the sector bit is next examined. If set, the first 5 bits of the P-register (the sector the instruction came from) are put into the Y-register together with the least significant 9 bits from M (the address portion of the instruction). If the sector bit is not set, zeroes are put in the first 5 bits of the Y-register.

If the instruction is neither indexed nor indirect addressed, the Y-register now contains the effective address of the operand, and the processor proceeds to the execution phase of the command. One memory cycle, or 1.7 μsec, has been utilized thus far.

If indexing is called for, the contents of the index register are added to the address in Y. This adds 2.04 μsec to the instruction timing.

Following indexing, the indirect bit is examined. If set, the contents of the address specified by Y are read into the M-register and the 14 least significant bits transferred to Y. The processor then executes further indexing and indirect cycles if necessary. Every level of indirect addressing adds 1.7 μsec to the execution time.

After all levels of indexing and indirect addressing have been completed, the processor proceeds to the execution phase of the instruction. The execution phase of each instruction is described in the following section. During the execution of most instructions, the P-register is incremented by one, and its contents are transferred to Y to set up the fetch of the next instruction.

All data transfers in the DDP-116 are effected in parallel. Thus the execution of most instructions such as load, store, add, or subtract are accomplished in one cycle. The times listed for each instruction include fetching the instruction and executing it but do not include indexing and indirect addressing time.

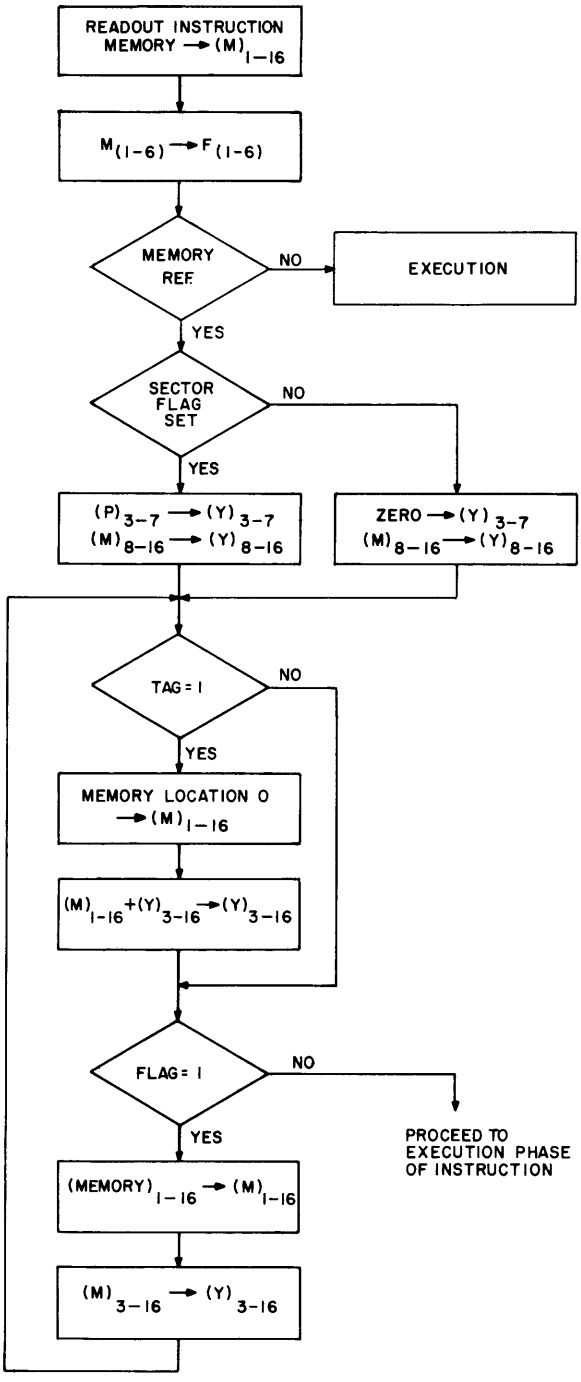


Figure 1-8. Fetch, Indexing and Indirect Addressing, Logic Flow Diagram

SECTION II
STANDARD INSTRUCTIONS

Contained in this section are descriptions of all standard DDP-116 instructions, grouped within the following functional categories: load and store, arithmetic, logical, shift, control, input/output, and transfer of control. The descriptions include the following information:

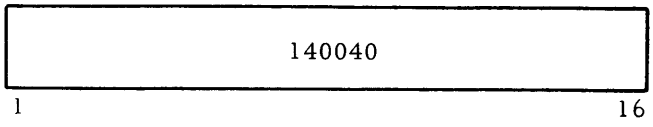
a. The 3-letter mnemonic assigned to the instruction
b. The time required to execute the instruction, including memory access of the instruction and operand, as applicable. Add 1.7 μsec to the listed time for each level of indirect addressing, and add 2.04 μsec for each use of indexing.
c. The effect of the instruction
d. Indicators affected by the instruction, where applicable
e. The type of instruction (memory reference, IOT, shift, or generic)

The type of the instruction determines the number of bits that must be used to specify the instruction (4 for memory reference instructions, 6 for IOT instructions, 10 for shift instructions, and 16 for generic instructions). Additionally, if the instruction is a memory reference type, the flag, tag, and sector bits (bits 1, 2, and 7) will function as described in Section I.

Flow diagrams showing the instruction execution process are provided where necessary. Table 2-1 at the end of this section contains summary descriptions of the instructions, listed in alphabetical order. Table 2-2 contains definitions of symbols and abbreviations.

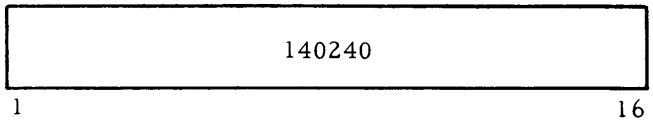
LOAD AND STORE INSTRUCTIONS

CRA — Clear A — 1.7 μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, are cleared to ZEROs.
TYPE: Generic

CCA — Clear C and A — 1.7 μsec



DESCRIPTION: The contents of the C-bit and the A-register, bits 1 through 16, are cleared to ZEROs.
TYPE: Generic

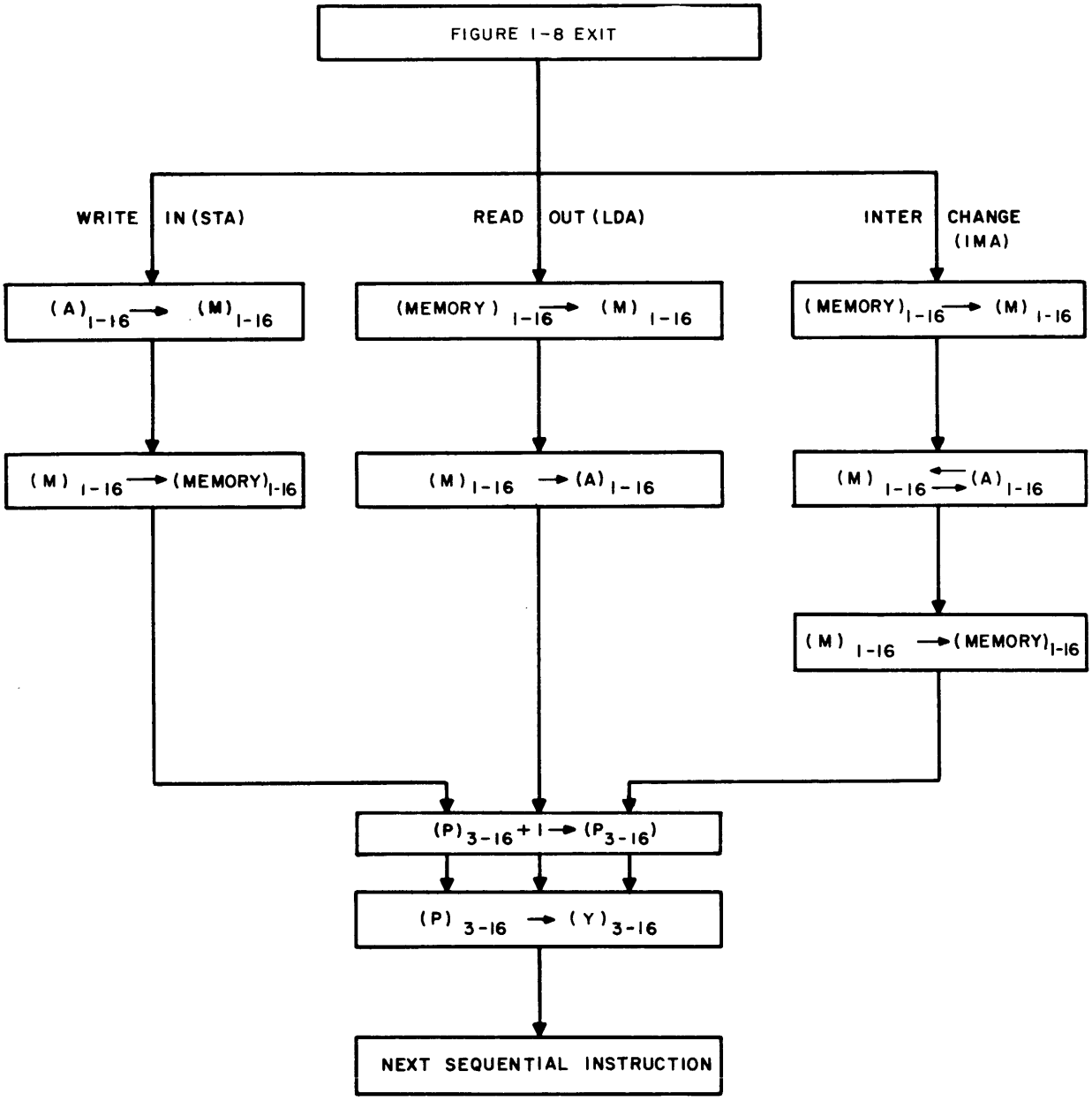
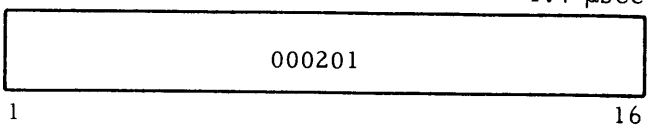


Figure 2-1. STA, LDA, and IMA Instructions, Flow Diagram

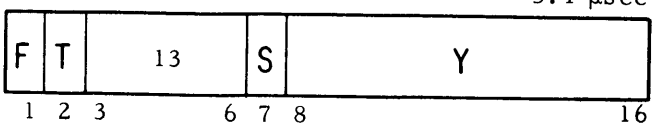
IAB Interchange A and B 1.7 μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 1 through 16, are interchanged.

TYPE: Generic

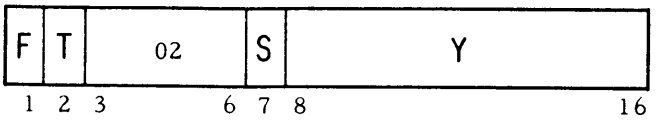
IMA Interchange A and Memory 5.1 μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the effective operand address in memory, bits 1 through 16, are interchanged. (See Figure 2-1.)

TYPE: Memory reference

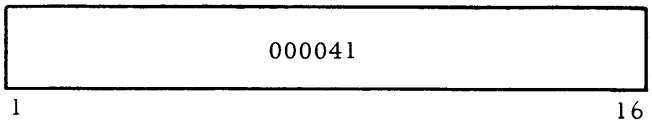
LDA Load A 3.4 μsec



DESCRIPTION: The contents of the effective operand address in memory, bits 1 through 16, replace the contents of the A-register, bits 1 through 16. The contents of the effective operand address in memory, bits 1 through 16, remain unchanged. (See Figure 2-1.)

TYPE: Memory reference

SCA* Shift Count to A 1.7 μsec

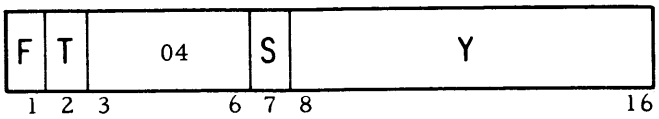


DESCRIPTION: The SCA instruction is used to store the contents of the shift counter in the A-register, bits 12 through 16. Bits 1 through 11 of the A-register are cleared. The shift counter is utilized by the multiply-divide and normalize instructions. Therefore, the contents of the shift counter must be extracted, if pertinent, prior to any subsequent use of the MPY, DIV, or NRM instructions.

TYPE: Generic

*Optional

STA — Store A — 3.4 μsec

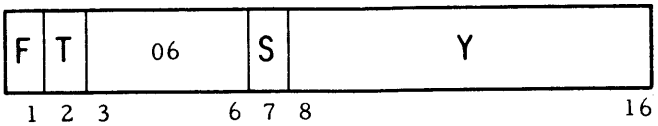


DESCRIPTION: The contents of the A-register, bits 1 through 16, replace the contents of the effective operand address in memory, bits 1 through 16. The contents of the A-register are unchanged. (See Figure 2-1.)

TYPE: Memory reference

ARITHMETIC INSTRUCTIONS

ADD — Add — 3.4 μsec

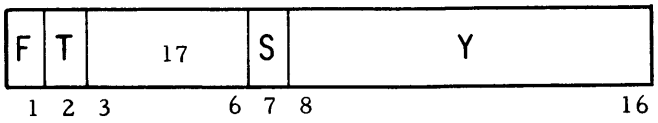


DESCRIPTION: The contents of the A-register, bits 1 through 16, are algebraically added to the contents of the effective operand address in memory, bits 1 through 16. The resultant sum replaces the contents of the A-register, bits 1 through 16. If overflow occurs, the C-bit is set and the computer proceeds to the next sequential instruction. The C-bit is reset if overflow does not occur. (See Figure 2-2.)

TYPE: Memory reference

INDICATORS: C-STATUS indicator lights to indicate overflow.

DIV* — Divide — 15.5 μsec avg, 16.67 μsec max



DESCRIPTION: The contents of the effective operand address in memory, bits 1 through 16, are divided into the contents of the combined A- and B-registers. The sign and the most significant half of the dividend is contained in the A-register, bits 1 through 16. The least significant half of the dividend is contained in the B-register, bits 2 through 16. The 16-bit quotient replaces the contents of the A-register, bits 1 through 16. The remainder replaces the contents of the B-register, bits 2 through 16. Bit 1 of the B-register is set to ZERO. With one exception, if the initial magnitude of the A-register is equal to or greater than the contents of the effective operand address in memory, the C-bit is set and the computer proceeds to the next sequential instruction. The exception is the case where the contents of the A-register and operand address are equal and opposite in sign, the A-register having the negative sign. For this case division is valid and the C-bit is not set. The C-bit is reset if an illegal divide does not occur.

*Optional

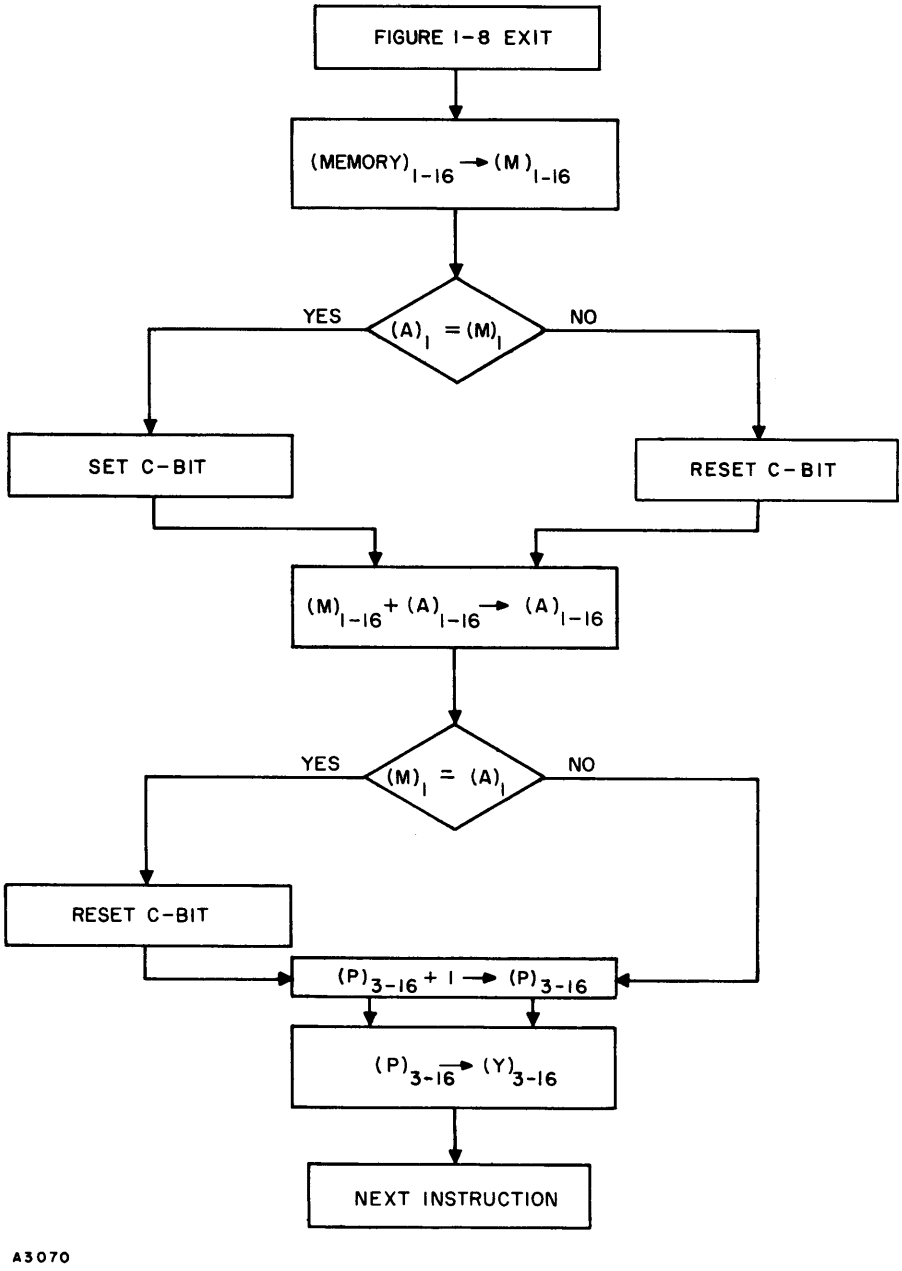
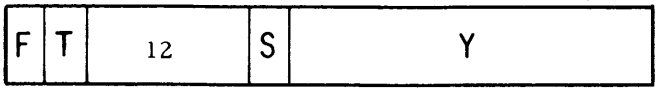


Figure 2-2. Add Instruction, Flow Diagram

TYPE: Memory reference

INDICATORS: C-STATUS indicator lights to indicate overflow.

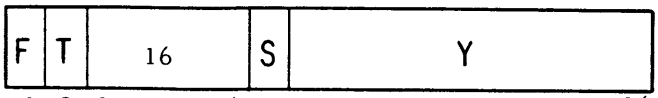
IRS Increment, Replace, and Skip 5.1 μsec



DESCRIPTION: The contents of the effective operand address in memory, bits 1 through 16, are incremented by 1 and the resulting sum replaces the contents of the effective operand address in memory, bits 1 through 16. If the result of the incrementation is ZERO, the next sequential instruction is skipped. Otherwise, the next sequential instruction is executed. (See Figure 2-3.)

TYPE: Memory reference

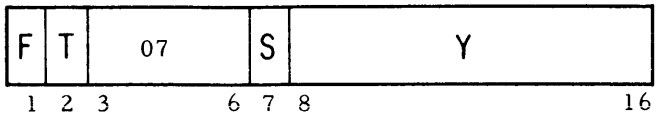
MPY* Multiply 9.18 μsec



DESCRIPTION: The contents of the A-register, sign and bits 2 through 16, are multiplied by the contents of the effective operand address in memory, sign and bits 2 through 16, forming a 30-bit product. The A-register, sign and bits 2 through 16, contain the most significant part of the product. The B-register, bits 2 through 16, contains the least significant part of the product. Bit 1 of the B-register is set to ZERO. There is no overflow indication on this instruction.

TYPE: Memory reference

SUB Subtract 3.4 μsec



DESCRIPTION: The contents of the effective operand address in memory, bits 1 through 16, are algebraically subtracted from the A-register, bits 1 through 16. The resultant difference replaces the contents of the A-register, bits 1 through 16. If overflow occurs, the C-bit is set and the computer proceeds to the next sequential instruction. The C-bit is reset if overflow does not occur.

TYPE: Memory reference

INDICATORS: C-STATUS indicator lights to indicate overflow.

*Optional

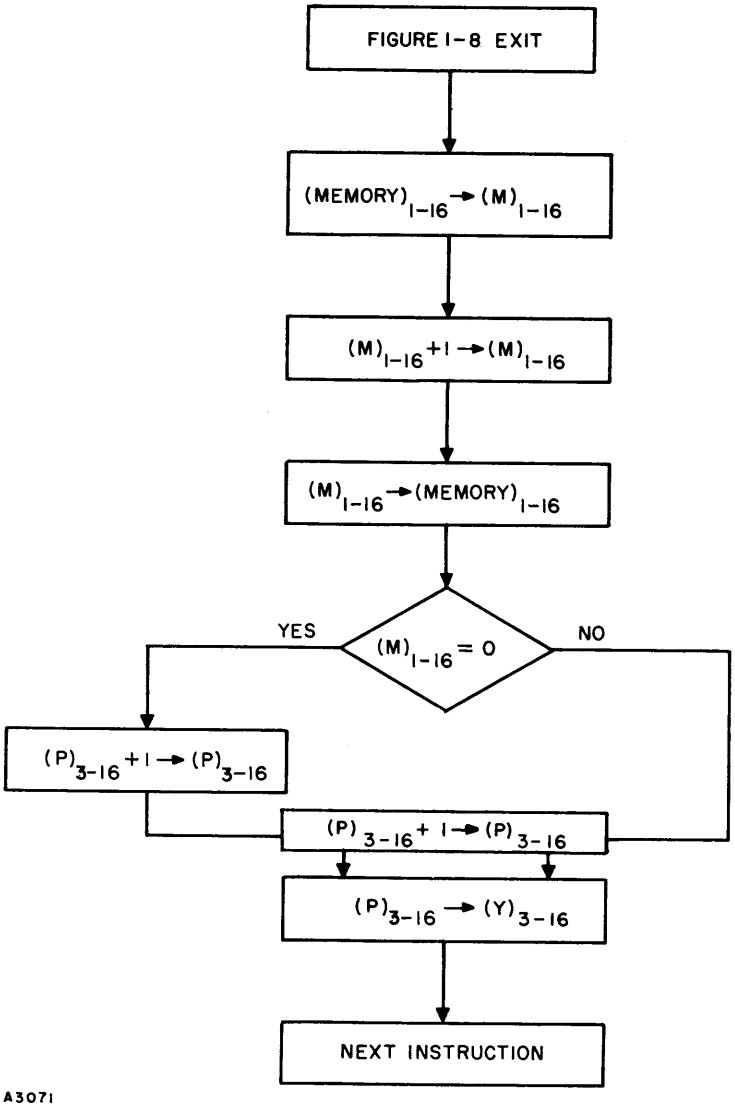
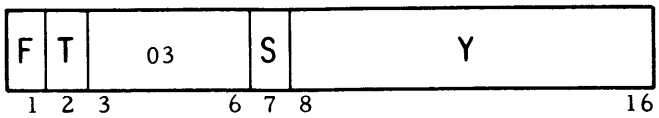


Figure 2-3. IRS Instruction, Flow Diagram

LOGICAL INSTRUCTIONS

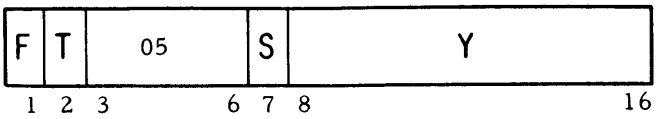
ANA — Logical AND — 3.4 μsec



DESCRIPTION: The logical product of the contents of the A-register, bits 1 through 16, and the contents of the effective operand address in memory is formed and the result replaces the contents of the A-register, bits 1 through 16. A ZERO is placed into the corresponding bit position in the A-register for each ZERO in the contents of the effective operand address in memory. The contents of the corresponding bit positions in the A-register are unchanged for each ONE in the contents of the addressed memory location. The contents of the addressed memory location are unchanged.

TYPE: Memory reference

ERA — Exclusive OR — 3.4 μsec



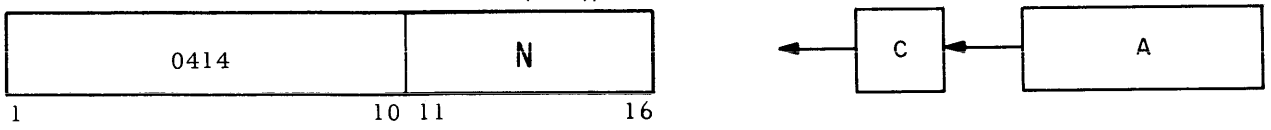
DESCRIPTION: The logical difference of the contents of the A-register, bits 1 through 16, and the contents of the effective operand address in memory, bits 1 through 16, is formed and the result replaces the contents of the A-register. The contents of the corresponding bit position of the A-register is complemented for each ONE in the contents of the addressed memory location. The corresponding bit of the A-register is unchanged for each ZERO in the contents of the addressed memory location. The contents of the addressed memory location, bits 1 through 16, are unchanged.

TYPE: Memory reference

SHIFT INSTRUCTIONS
(See Figure 2-4.)

Logical Left Shift

LGL — 1.7+0.34N(N>0) 2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, are shifted left through the C-bit the number of positions specified in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of the A-register are lost and ZEROs are shifted into the vacated positions of the register. Each bit shifted out of the A-register enters the C-bit. A maximum shift of 63-bit positions may be specified. If a shift greater than 15 is specified, bits 1 through 16

of the A-register will be ZERO. However, the number of actual shifts performed equals the number specified by the instruction.

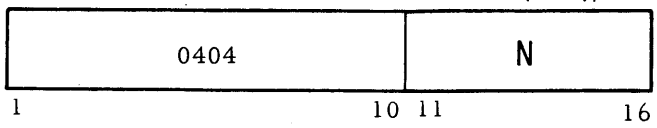
TYPE: Shift

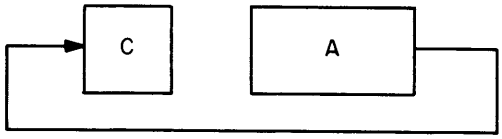
INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Logical Right Shift

LGR

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, are shifted right through the C-bit the number of positions specified in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of the A-register are lost and ZEROs are shifted into the vacated positions of the register. Each bit shifted out of the A-register enters the C-bit. A maximum shift of 63 bit positions may be specified. If a shift greater than 15 is specified, bits 1 through 16 of the A-register will be ZERO. However, the number of actual shifts performed equals the number specified by the instruction.

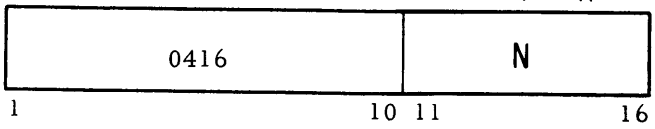
TYPE: Shift

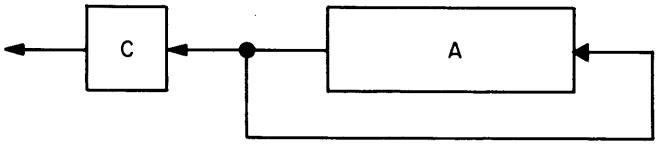
INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction .

Logical Left Rotate

ALR

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, are rotated to the left (end around carry) the number of positions specified in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of bit position 1 of the A-register enter bit position 16 of the A-register. Each bit shifted out of bit position 1 of the A-register enters the C-bit. A maximum shift of 63-bit positions may be specified. If a shift of greater than 15-bit positions is specified, the net result of the shift is modulo 15. However, the number of actual shifts performed equals the number specified by the instruction.

TYPE: Shift

INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

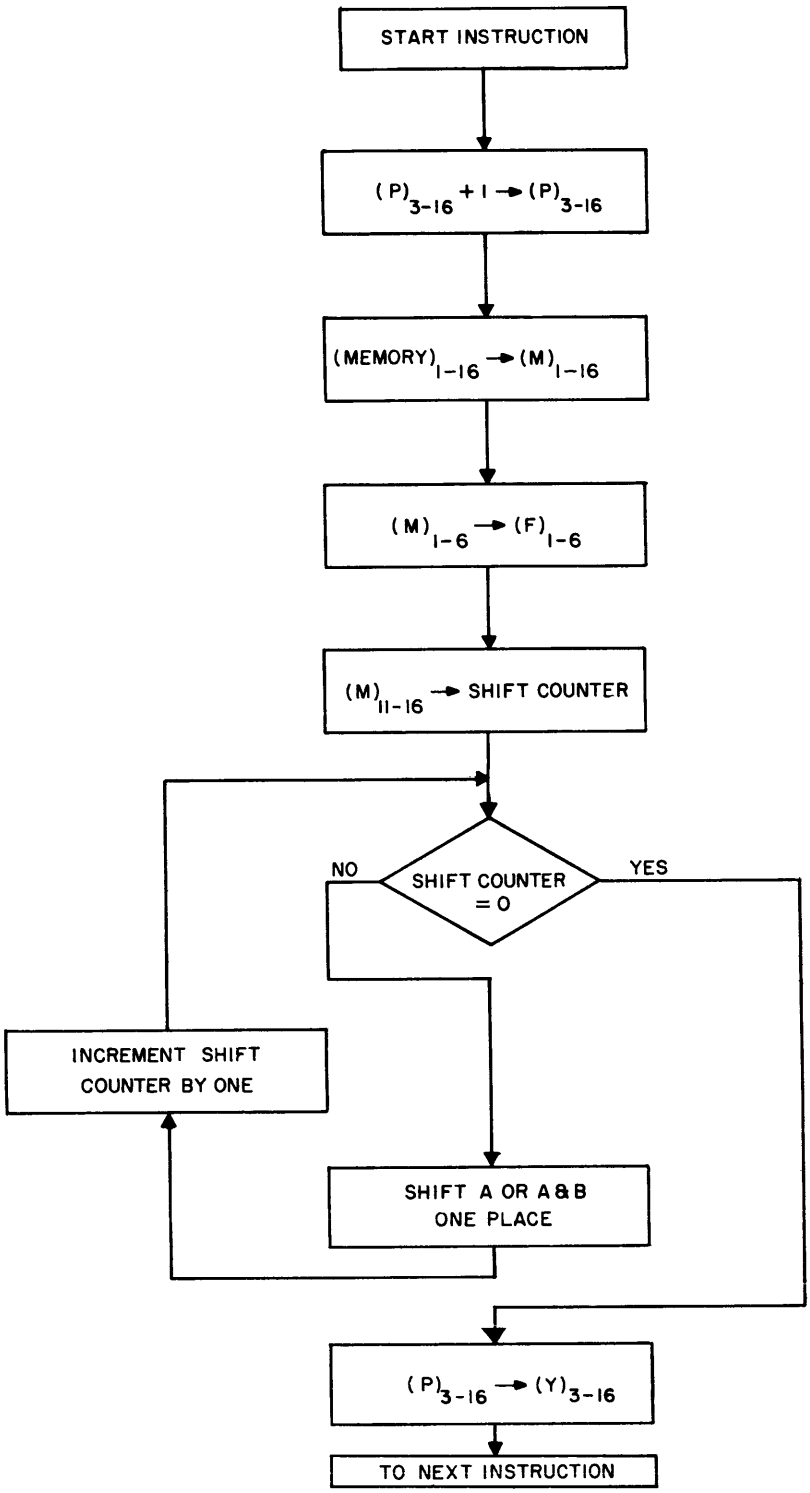
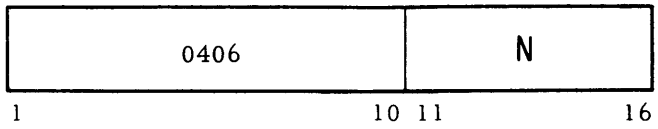


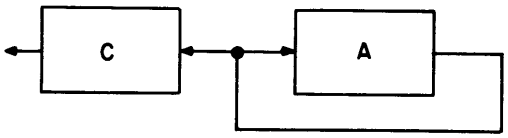
Figure 2-4. Shift Instructions, General Flow Diagram

Logical Right Rotate

ARR

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, are rotated to the right (end-around carry) the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of bit position 16 of the A-register enter bit position 1 of the A-register. Each bit shifted out of bit position 16 of the A-register enters the C-bit. A maximum shift of 63-bit positions may be specified. If a shift greater than 15 is specified, the net result of the shift is modulo 15. However, the number of actual shifts performed equals the number specified by the instruction.

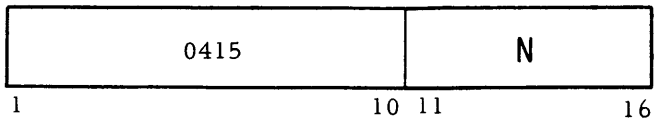
TYPE: Shift

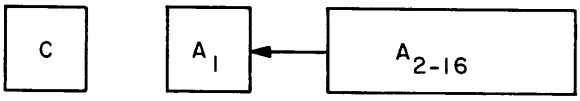
INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Arithmetic Left Shift

ALS

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, are shifted left the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. The C-bit indicates an overflow in this instruction. It is set if the sign bit (bit 1) is altered during execution of the instruction and reset if the sign bit is not altered. Bits shifted out of the A-register are lost and ZEROs are shifted into the vacated positions of the register. A maximum shift of 63 bit positions may be specified. If a shift greater than 15 is specified, bits 1 through 16 of the A-register will be ZERO. The number of actual shifts performed equals the number specified by the instruction.

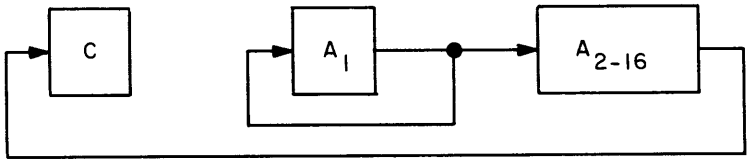
TYPE: Shift

INDICATORS: The C-STATUS indicator lights to indicate overflow.

Arithmetic Right Shift

ARS

1.7+0.34N(N>0)
2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, are shifted right through the C-bit the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. The sign bit, bit 1, of the A-register does not change. The sign bit (bit position 1) is shifted into the vacated positions of the A-register. Each bit shifted out of the A-register enters the C-bit. A maximum shift of 63 bit positions may be specified. If a shift greater than 14 is specified, bits 2 through 16 of the A-register will be the same as the sign. However, the number of actual shifts performed equals the number specified by the instruction. The sign bit remains as it was prior to the execution of the instruction.

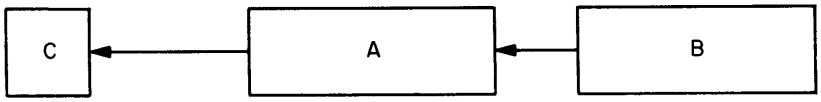
TYPE: Shift

INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Long Left Logical Shift

LLL

1.7+0.34N(N>0)
2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 1 through 16, are treated as a single 32-bit register. These bits are shifted to the left through the C-bit the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of the bit position 1 of the B-register enter bit position 16 of the A-register. Bits shifted out of bit position 1 of the A-register, through the C-bit, are lost. Each bit shifted out of bit position 1 of the A-register enters the C-bit. ZEROs are shifted into the vacated positions of the A- and B-registers through bit position 16 of the B-register.

A maximum shift of 63 bit positions may be specified. If a shift greater than 31 is specified, bits 1 through 16 of the A- and B-registers will be ZERO. However, the number of actual shifts performed equals the number specified by the instruction.

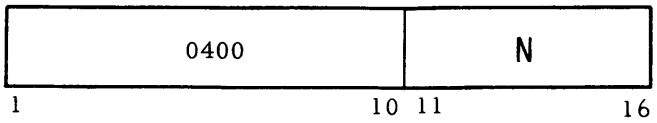
TYPE: Shift

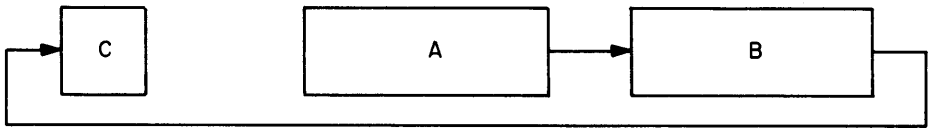
INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Long Right Logical Shift

LRL

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 1 through 16, are treated as a single 32-bit register. These bits are shifted to the right through the C-bit the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of bit position 16 of the A-register enter bit position 1 of the B-register. Bits shifted out of bit position 16 of the B-register, through the C-bit,are lost. Each bit shifted out of the B-register enters the C-bit. ZEROs are shifted into the vacated positions of the A- and B-registers through bit position 1 of the A-register. A maximum shift of 63-bit positions may be specified. However, the number of actual shifts performed equals the number specified by the instruction. If a shift greater than 31 occurs, bits 1 through 16 of the A- and B-registers will be ZERO.

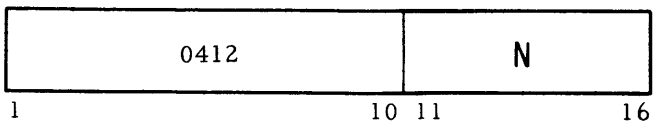
TYPE: Shift

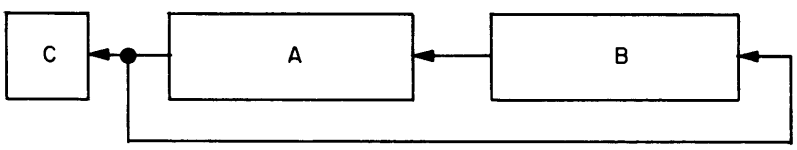
INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Long Left Rotate

LLR

1.7+0.34N(N>0)
2.04(N=0)μsec





DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 1 through 16, are treated as a 32-bit register. These bits are rotated

to the left (end-around carry) the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of bit position 1 of the B-register enter bit position 16 of the A-register. Bits shifted out of bit position 1 of the A-register enter bit position 16 of the B-register. Each bit shifted out of bit position 1 of the A-register enters the C-bit. A maximum shift of 63-bit positions may be specified. If a shift greater than 31 is specified, the net result of the shift is modulo 32. However, the number of actual shifts performed equals the number specified by the instruction.

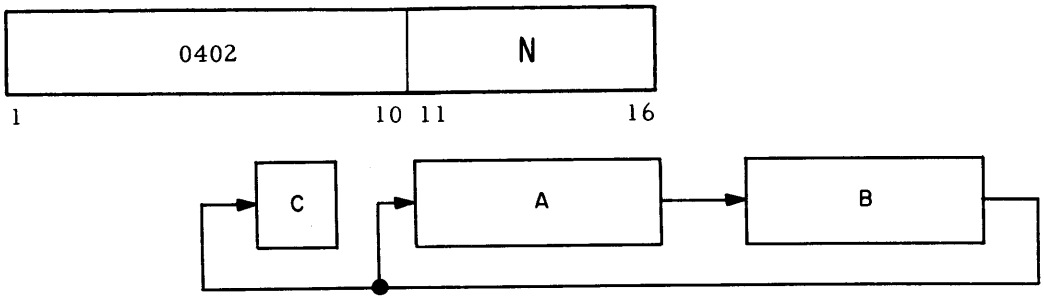
TYPE: Shift

INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Long Right Rotate

LRR

1.7+0.34N(N>0)
2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 1 through 16, are treated as a single 32-bit register. These bits are rotated to the right (end-around carry) the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bits shifted out of bit position 16 of the B-register enter bit position 1 of the A-register. Bits shifted out of bit position 16 of the A-register enter bit position 1 of the B-register. Each bit shifted out of bit position 16 of the B-register enters the C-bit. A maximum shift of 63-bit positions may be specified. If a shift of greater than 31 is specified, the net result of the shift is modulo 32. However, the number of actual shifts performed equals the number specified by the instruction.

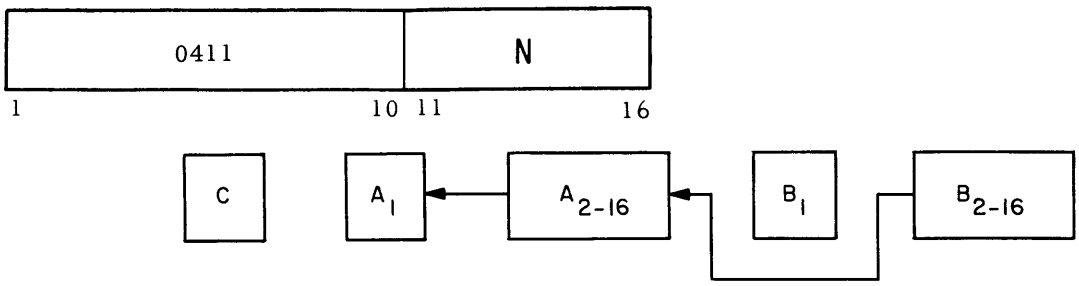
TYPE: Shift

INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Long Arithmetic Left Shift

LLS

1.7+0.34N(N>0)
2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 2 through 16, are treated as a single 31-bit register. These bits are shifted to the left the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. The sign bit, bit 1 of the B-register is not changed. The C-bit is set if the sign, bit 1 of the A-register, is altered during execution of the instruction. It is reset if the sign bit is not altered. Bits shifted out of bit position 2 of the B-register enter bit position 16 of the A-register. Bits shifted out of bit position 1 of the A-register are lost. ZEROs are shifted into the vacated positions of the A- and B-registers through bit position 16 of the B-register. A maximum shift of 63 bit positions may be specified. However, the number of actual shifts performed equals the number specified by the instruction. If a shift greater than 31 is specified, bits 1 through 16 of the A-register and bits 2 through 16 of the B-register will be ZERO.

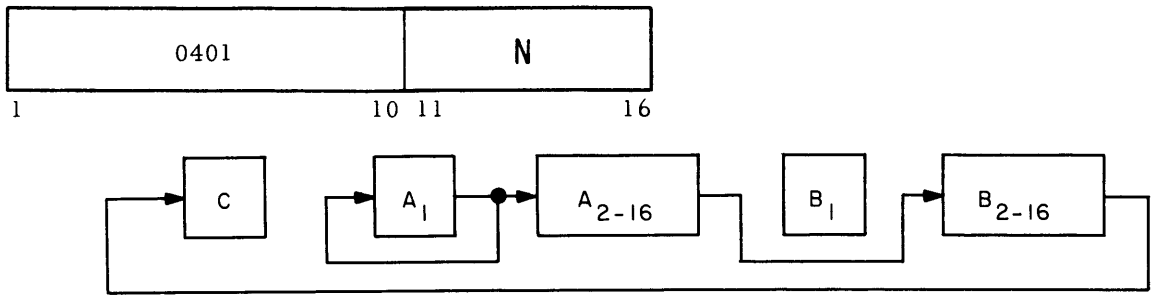
TYPE: Shift

INDICATORS: The C-STATUS indicator lights to indicate overflow.

Long Arithmetic Right Shift

LRS

1.7+0.34N(N>0)
2.04(N=0)μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, and the contents of the B-register, bits 2 through 16, are treated as a single 31-bit register. These bits are shifted to the right through the C-bit the number of positions specified, in two's complement notation, by the six least significant bits, bits 11 through 16, of the instruction. Bit 1 of the A- and B-registers do not change. Bits shifted out of bit position 16 of the

B-register are lost. Bits shifted out of bit position 16 of the A-register enter bit position 2 of the B-register. Each bit shifted out of the B-register enters the C-bit. The sign is copied into the vacated positions of the A- and B-registers through bit position 1 of the A-register. A maximum shift of 63 bit positions may be specified. However, the number of shifts performed equals the number specified by the instruction. If a shift greater than 30 is specified, bits 2 through 16 of the A- and B-register will be the same as the sign of the A-register.

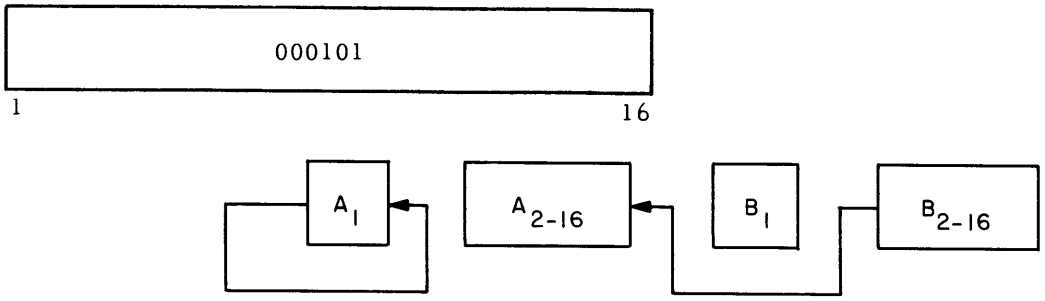
TYPE: Shift

INDICATORS: The C-STATUS indicator lights if the C-bit is set at the end of the instruction.

Normalize

NRM*

2.38+
0.34 Nμsec



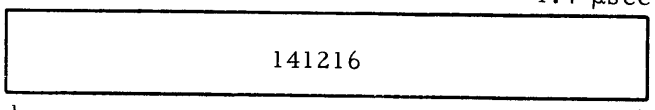
DESCRIPTION: The normalize instruction considers the 15 magnitude bits of the B-register and the A-register to be one 30-bit register. The A-register comprises the most significant half of the number. When the number to be normalized is positive, bits 2 through 16 of both the A- and B-registers are shifted until bit position 2 of the A-register contains a ONE. When the number is negative, bits 2 through 16 of both the A- and B-registers are shifted left until bit position 2 of the A-register becomes a ZERO. The number of positions shifted (or the exponent) is stored in the shift counter. This count (or exponent) is made available by the SCA instruction (shift count to A-register). If the contents of both registers are ZERO, 32 shifts are performed before the instruction is terminated, leaving ZERO in the shift counter. Bits shifted out of bit position 2 of the A-register are lost. Bits shifted out of bit position 2 of the B-register enter bit position 16 of the A-register. ZEROs are shifted into vacated bit positions of the B-register through bit position 16 of the B-register.

TYPE: Generic

*Optional

Add C to A

ACA 1.7 μsec



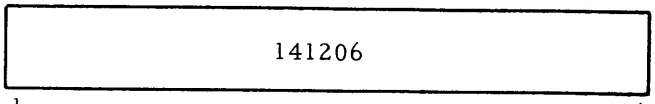
DESCRIPTION: If the contents of the C-bit is a ONE, it is added to the contents of the A-register, bits 1 through 16. The resulting sum replaces the contents of the A-register, bits 1 through 16. If the content of the C-bit is ZERO, no operation occurs. Overflow is possible with this instruction. If overflow occurs, the C-bit is set. If overflow does not occur, the C-bit is reset.

TYPE: Generic

INDICATORS: C-STATUS indicator lights to indicate overflow.

Add One to A

AOA 1.7 μsec



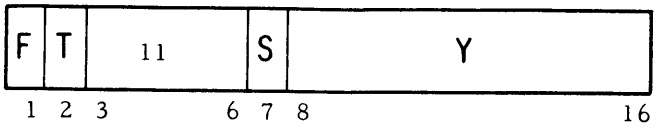
DESCRIPTION: Add ONE to the contents of the A-register, bits 1 through 16. The resulting sum replaces the contents of the A-register, bits 1 through 16. Overflow is possible with this instruction. If overflow occurs, the C-bit is set. If overflow does not occur, the C-bit is reset.

TYPE: Generic

INDICATORS: C-STATUS indicator lights to indicate overflow.

Compare

CAS 5.1 μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, are algebraically compared to the contents of the effective operand address in memory, bits 1 through 16. If the value in the A-register is greater than the value in memory, the computer proceeds to the next sequential instruction. If the value in the A-register is equal to the value in memory, the next sequential instruction is skipped. If the value in the A-register is less than the value in memory, the next two sequential instructions are skipped. The contents of the A-register, bits 1 through 16, and the contents of the memory, bits 1 through 16, are unchanged. (See Figure 2-5.)

TYPE: Memory reference

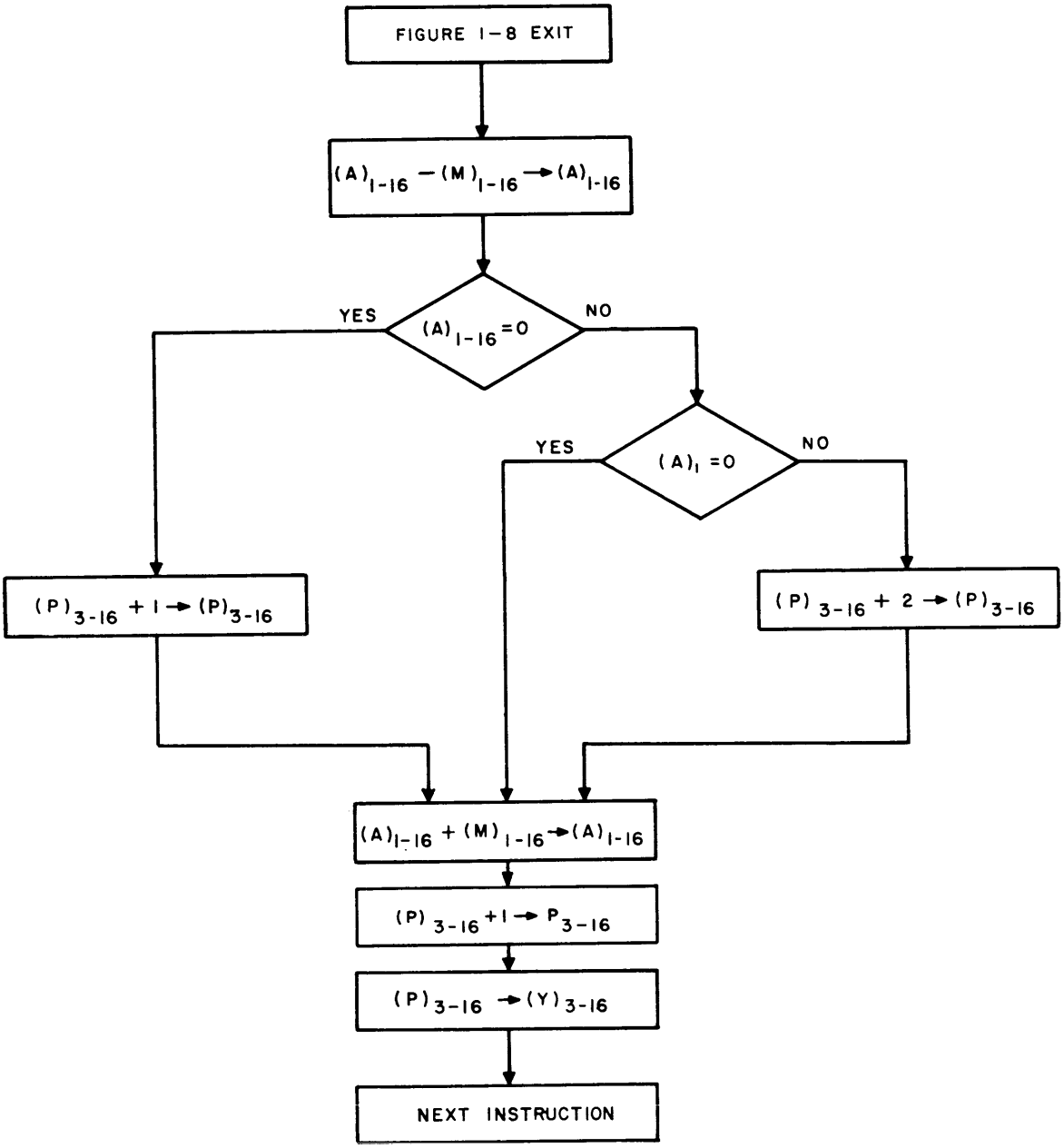
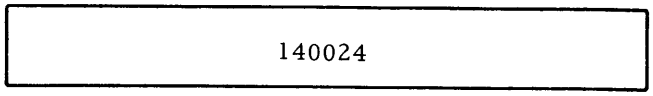


Figure 2-5. CAS Instruction, Flow Diagram

Complement A Sign

CHS 1.7 μsec

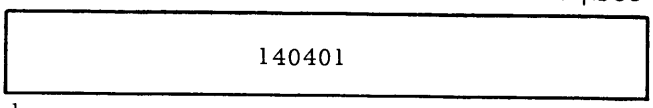


DESCRIPTION: The sign of the A-register, bit position 1, is complemented. Bits 2 through 16 of the A-register do not change.

TYPE: Generic

Complement A Register

CMA 1.7 μsec

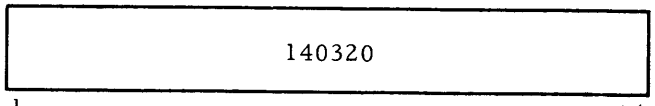


DESCRIPTION: The contents of the A-register, bits 1 through 16, are complemented (one's complement) and the complement replaces the contents of the A-register, bits 1 through 16. Bit positions that are ZERO are changed to ONE. Bit positions that are ONE are changed to ZERO.

TYPE: Generic

Copy Sign and Set Sign Plus

CSA 1.7 μsec



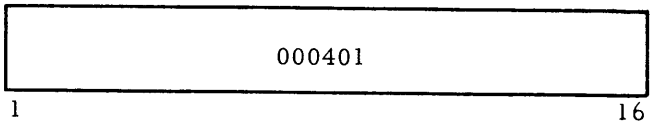
DESCRIPTION: The sign bit, bit 1, of the A-register, minus or plus, is copied into the C-bit and the sign bit position, bit 1, of the A-register is cleared to ZERO. Bits 2 through 16 of the A-register do not change.

TYPE: Generic

INDICATORS: C-STATUS indicator lights if C-bit is set.

Enable Program Interrupt

ENB 1.7 μsec



DESCRIPTION: This instruction enables program interrupt. It sets the interrupt control flip-flop so as to permit interrupts. If an interrupt is present on the basic interrupt line or any optional interrupt lines at the end of the next instruction, control will be

transferred through the appropriate standard location.

TYPE: Generic

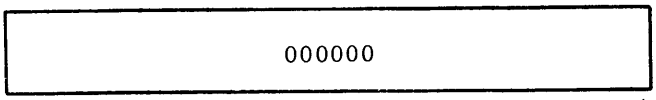
INDICATORS: PI indicator lights to indicate that interrupt is enabled.

NOTE

Do not give the ENB instruction before the desired interrupt masks are set up by an OTA '0020.

Halt

HLT μsec



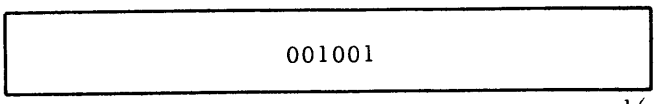
DESCRIPTION: The computer will halt until the START button on the operator's console is depressed at which time execution will be resumed at the next sequential instruction.

TYPE: Generic

INDICATORS: HALT indicator (in START button) is lighted to indicate machine is in halt condition.

Inhibit Program Interrupt

INH 1.7 μsec



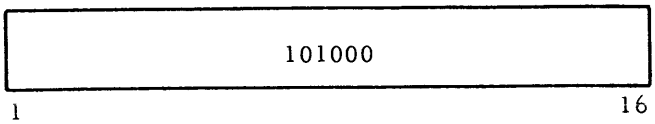
DESCRIPTION: This instruction inhibits interrupt. The computer cannot be interrupted until another ENB (enable program interrupt) instruction is executed.

TYPE: Generic

INDICATOR: PI indicator is extinguished indicating that program interrupt is inhibited.

No Operation

NOP 1.7 μsec

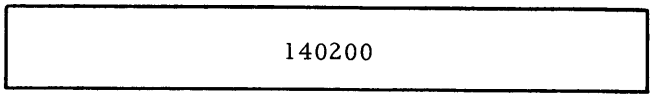


DESCRIPTION: No operation is performed by this instruction. The computer proceeds to and executes the next sequential instruction.

TYPE: Generic

Reset C to Zero

RCB 1.7 μsec



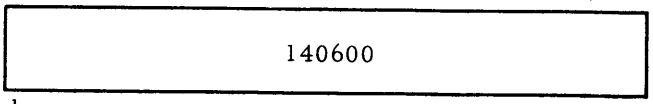
DESCRIPTION: The content of the C-bit is reset to ZERO. If the content of the C-bit is ZERO, no change occurs.

TYPE: Generic

INDICATORS: C-STATUS indicator is extinguished.

Set C to One

SCB 1.7 μsec



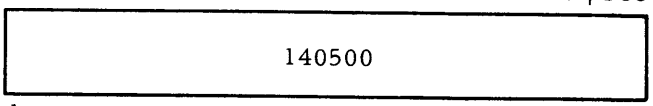
DESCRIPTION: The content of the C-bit is set to ONE. If the content of the C-bit is ONE, no change occurs.

TYPE: Generic

INDICATORS: C-STATUS indicator lights.

Set A Sign Minus

SSM 1.7 μsec

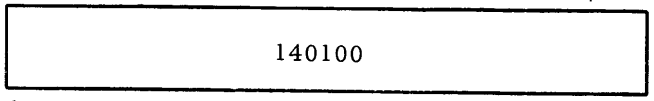


DESCRIPTION: If the sign of the contents of the A-register is plus (A_1=0), the sign is changed to minus (A_1=1). If the sign is minus, no change occurs. Bits 2 through 16 of the A-register do not change.

TYPE: Generic

Set A Sign Plus

SSP 1.7 μsec

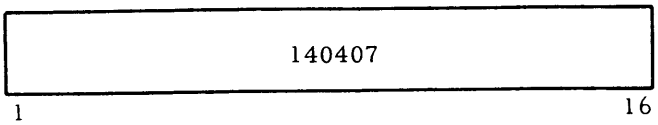


DESCRIPTION: If the sign of the contents of the A-register is minus (A_1=1), the sign is changed to plus (A_1=0). If the sign is plus, no change occurs. Bits 2 through 16 of the A-register do not change.

TYPE: Generic

Two's Complement A

TCA 1.7 μsec



DESCRIPTION: The contents of the A-register, bits 1 through 16, are two's complemented and the result replaces the contents of the A-register, bits 1 through 16.

TYPE: Generic

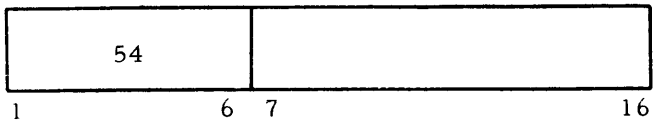
INPUT/OUTPUT INSTRUCTIONS

General

The basic information flow for the four input/output instructions is shown in Figure 2-6. Bits 11 through 16 are generally used to specify the device to which the instruction refers and bits 7 through 10 are used to specify the function to be performed by the I/O device.

Input to A

INA 5.1 μsec

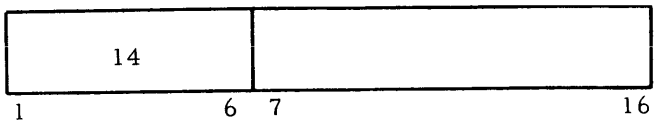


DESCRIPTION: Bits 7 through 16 of the instruction word are taken to the I/O bus address lines and decoded; however, bit 7 is normally not used as part of a function code. If the ready flip-flop for the device specified is not set, the instruction is treated as a NOP and the next sequential instruction is executed. If the device ready flip-flop is set (device ready) bit 7 of the instruction word is tested. If bit 7 is set, the A-register is cleared and the contents of the input bus, bits 1 through 16, are transferred to the A-register, bits 1 through 16. If bit 7 is not set, the contents of the input bus, bits 1 through 16, are ORed with the contents of the A-register, bits 1 through 16, and the result replaces the contents of the A-register, bits 1 through 16.

TYPE: IOT

Output Control Pulse

OCP 3.4 μsec



DESCRIPTION: Bits 7 through 16 of the instruction word are taken to the I/O bus address lines. The 10 bits are decoded within the device interface enabling generation of an output control pulse (OCP) used to set up the operating mode or perform a control function in a device. Bits 11 through 16 select the device interface. Bits 7 through 10 specify the function to be performed by the I/O device. The OCP may cause the I/O device to perform a mechanical motion, set up the mode of a device, etc.

TYPE: IOT

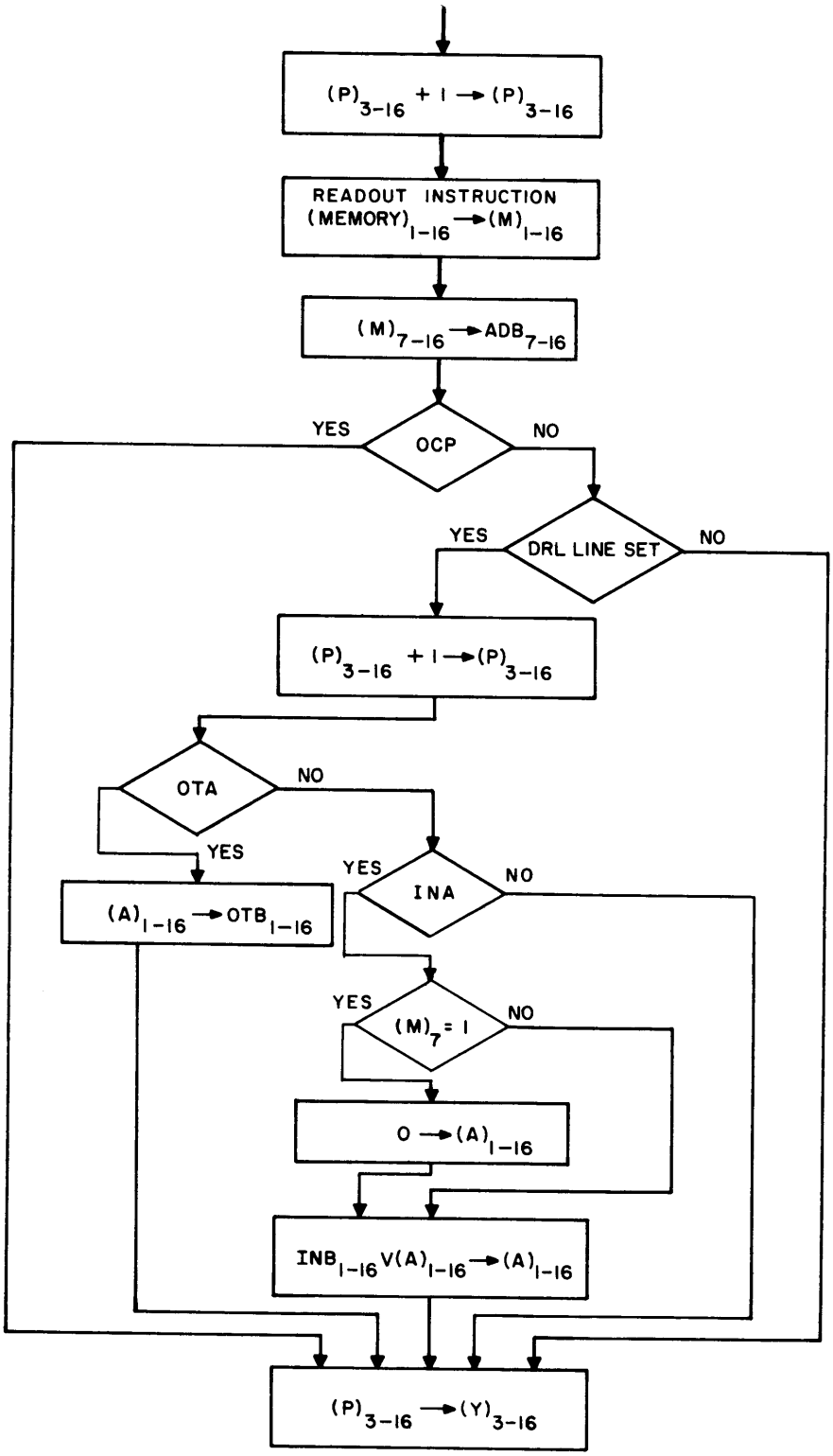


Figure 2-6. IOT Instructions, General Flow Diagram

Output from A

OTA 3.4 μsec

DESCRIPTION: Bits 7 through 16 of the instruction word are taken to the address lines of the I/O bus. Bits 11 through 16 are decoded and select the proper device interface. Bits 7 through 10 specify the function to be performed. If the device is not ready, the next sequential instruction is executed. If the device is ready, the contents of the A-register, bits 1 through 16, are transferred to the output bus, bits 1 through 16, and the next sequential instruction is skipped. The number of bits accepted by the device interface is contingent upon the size of the device buffer. Example: Only the least significan 8 bits, bits 9 through 16, are transmitted to the ASR-33 buffer.

TYPE: IOT

Skip If Ready Line Set

SKS 3.4 μsec

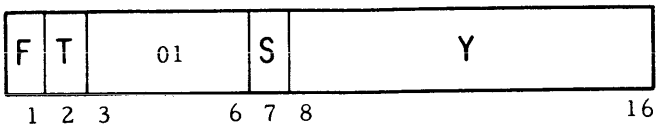
DESCRIPTION: Bits 11 through 16 of the instruction specify the device to be interrogated; bits 7 through 10 of the instruction specify the condition to be sensed. If the selected condition causes the ready line to be set, the computer skips the next sequential instruction and continues with the program. If the ready line is not set, the computer proceeds to the next sequential instruction.

TYPE: IOT

TRANSFER OF CONTROL INSTRUCTIONS

Unconditional Jump

JMP 1.7 μsec

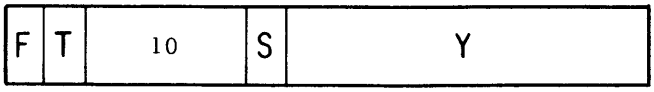


DESCRIPTION: The effective operand address of the instruction replaces the contents of the program counter (P), bits 3 through 16.

TYPE: Memory reference

Jump and Store Location

JST 5.1 μsec



DESCRIPTION: The contents of the program counter (P), bits 3 through 16, replace the 14 least significant bits, bits 3 through 16, of the memory location addressed by the effective operand address. The two most significant bits of the memory location, bits 1 and 2, are left unchanged. The effective operand address plus one is loaded into the program counter, P, bits 3 through 16. (See Figure 2-7.)

TYPE: Memory reference

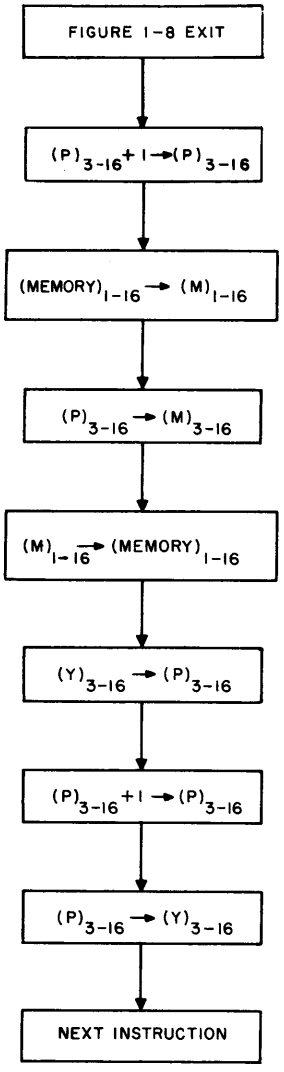


Figure 2-7. JST Instruction, Flow Diagram

Skip If C Reset

SRC 1.7 μsec

100001

1 16

DESCRIPTION: Skip the next sequential instruction if the C-bit is reset. Execute the next sequential instruction if the C-bit is set. (See Figure 2-8 for a general flow diagram of the skip instructions.) The C-bit is unchanged.

TYPE: Generic

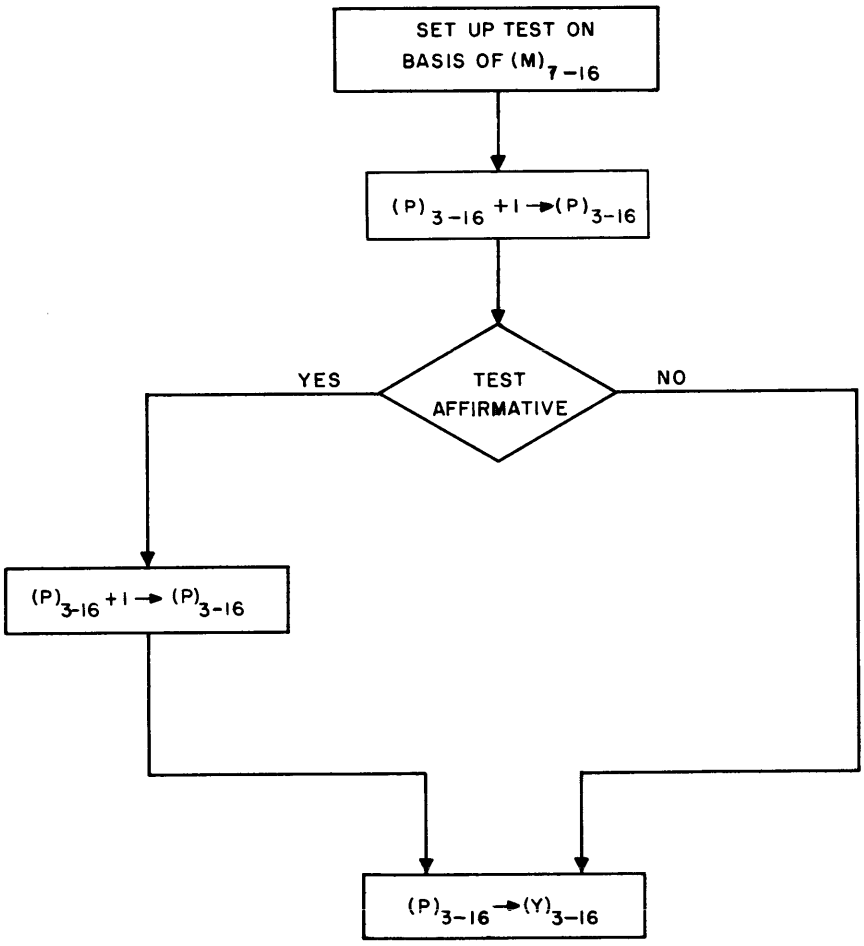
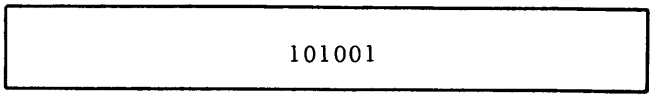


Figure 2-8. Skip Instructions, General Flow Diagram

Skip If C Set

SSC 1.7 μsec

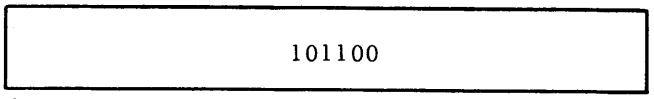


DESCRIPTION: Skip the next sequential instruction if the C-bit is set. Execute the next sequential instruction if the C-bit is reset. The C-bit is unchanged.

TYPE: Generic

Skip If A_{16} Non-Zero

SLN 1.7 μsec

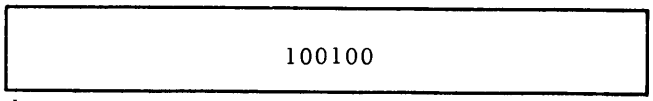


DESCRIPTION: If the least significant bit of the A-register is ONE (A_{16}=1), the computer skips the next sequential instruction. If the least significant bit of the A-register is ZERO (A_{16}=0), the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged.

TYPE: Generic

Skip If A_{16} Zero

SLZ 1.7 μsec

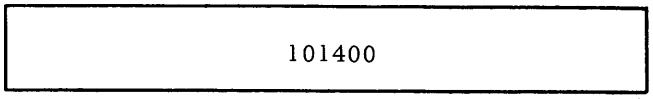


DESCRIPTION: If the least significant bit of the A-register is ZERO (A_{16}=0), the computer skips the next sequential instruction. If the least significant bit of the A-register is not ZERO (A_{16}=1), the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged.

TYPE: Generic

Skip If A Minus

SMI 1.7 μsec

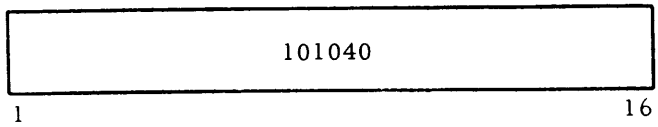


DESCRIPTION: If the sign of the A-register is negative (A_1=1), the computer skips the next sequential instruction. If the sign of the A-register is positive (A_1=0), the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged. The magnitude bits, bits 2 through 16, of the A-register are not examined.

TYPE: Generic

Skip If A Not Zero

SNZ 1.7 μsec

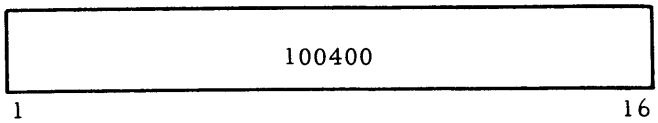


DESCRIPTION: If the contents of the A-register, bits 1 through 16, are not ZERO, the computer skips the next sequential instruction. If the contents of the A-register, bits 1 through 16 are ZERO, the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged.

TYPE: Generic

Skip If A Plus

SPL 1.7 μsec

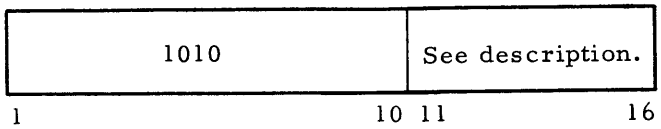


DESCRIPTION: If the sign of the A-register is positive (A_1=0), the computer skips the next sequential instruction. If the sign of the A-register is negative (A_1=1), the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged. The magnitude bits, bits 2 through 16, of the A-register are not examined.

TYPE: Generic

Skip If Sense Switches Set

SS1-2-3-4 1.7 μsec



DESCRIPTION: Skip the next sequential instruction if the sense switch specified by bits 11 through 16 of the instruction word is set. Execute the next sequential instruction if the sense switch specified by bits 11 through 16 of the instruction word is reset.

Sense Switch	Bits 11-16	
SS1	20	(bit 12 = 1)
SS2	10	(bit 13 = 1)
SS3	04	(bit 14 = 1)
SS4	02	(bit 15 = 1)

TYPE: Generic

Skip If Sense Switches Reset

SR1-2-3-4 1.7 μsec

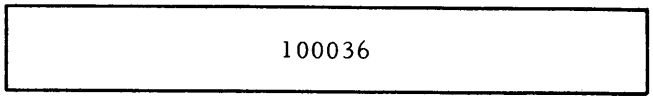
DESCRIPTION: Skip the next sequential instruction if the sense switch specified by bits 11 through 16 of the instruction word is reset. Execute the next sequential instruction if the sense switch specified by bits 11 through 16 of the instruction word is set.

Sense Switch	Bits 11-16	
SR1	20	(bit 12 = 1)
SR2	10	(bit 13 = 1)
SR3	04	(bit 14 = 1)
SR4	02	(bit 15 = 1)

TYPE: Generic

Skip If No Sense Switch Set

SSR 1.7 μsec

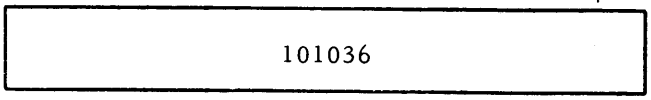


DESCRIPTION: Skip the next sequential instruction if no sense switch is set.

TYPE: Generic

Skip If Any Sense Switch Set

SSS 1.7 μsec

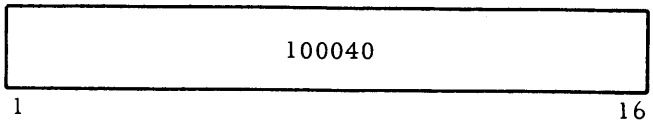


DESCRIPTION: Skip the next sequential instruction if any sense switch is set.

TYPE: Generic

Skip If A Zero

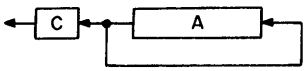
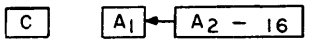
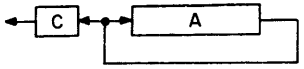
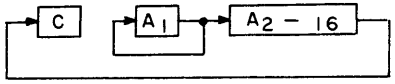
SZE 1.7 μsec



DESCRIPTION: If the contents of the A-register, bits 1 through 16, are ZERO, the computer skips the next sequential instruction. If the contents of the A-register, bits 1 through 16, are not ZERO, the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 16, are unchanged.

TYPE: Generic

Table 2-1.
Summary Description of DDP-116 Instructions

Mnemonic	Functional Description	Description	Time (μsec)
ACA	Add C to A	If C=1, (A)+1→(A) If C=0, (A) = (A) If Overflow 1 →(C) If no Overflow 0 →(C)	1.7
ADD	Add	(A)+ (EOA) →(A) If Overflow 1 →(C) If no Overflow 0 →(C)	3.4
ALR	Logical Left Rotate		1.7 + 0.34 N (N> 0) 2.04 (N=0)
ALS	Arithmetic Left Shift	 If A_1 changed 1 →(C) If A_1 not changed 0 →(C)	1.7 + 0.34 N (N> 0) 2.04 (N=0)
ANA	AND to A	(A)∧(EOA) →(A)	3.4
AOA	ADD 1 to A	(A) + 1 →(A) If Overflow 1 →(C) If no Overflow 0 →(C)	1.7
ARR	Logical Right Rotate		1.7 + 0.34 N (N> 0) 2.04 (N=0)
ARS	Arithmetic Right Shift		1.7 + 0.34 N (N> 0) 2.04 (N=0)

CAS	Compare Memory and A	If (A)> (EOA) Go to P
		If (A)= (EOA) Go to P+1
		If (A)< (EOA) Go to P+2
			5.1

Table 2-1. (Cont)
Summary Description of DDP-116 Instructions

Mnemonic	Functional Description	Description	Time (μsec)
CCA	Clear C and A	O → (C) O → (A)	1.7
CHS	Change Sign	$(\overline{A})_1 \rightarrow (A)_1$	1.7
CMA	Complement A	$(\overline{A}) \rightarrow (A)$	1.7
CRA	Clear A	$0 \rightarrow (A)$	1.7
CSA	Copy sign to C-bit and set A sign plus	$(A_1) \rightarrow (C)$, $0 \rightarrow (A_1)$	1.7
DIV*	Divide	$(A)(B) \div (EOA) \rightarrow (A)_{Quotient}\ (B)_{Remainder}$ A High order, B Low order } dividend	16.67 (maximum)
ENB	Turn Program Interrupt On	Enable Program Interrupt System	1.7
ERA	Exclusive OR to A	$(A) \veebar (EOA) \rightarrow (A)$ Logical Difference	3.4
HLT	Halt	Stop Program Operation	
IAB	Interchange A and B	$(A) \leftrightarrows (B)$	1.7
IMA	Interchange Memory and A	$(A) \leftrightarrows (EOA)$	5.1

INA	Input to A	If (Ready) = 0, Go to P, no input
		If (Ready) = 1, and $(IW)_7 = 1$, (INB) → (A) Go to P+1
		If (Ready) = 1, and $(IW)_7 = 0$, (INB) ∨ (A) → (A)- Go to P+1
			5.1

INH	Turn Program Interrupt Off	Inhibit Program Interrupt System	1.7

IRS	Increment, Replace and Skip	(EOA)+1→ (EOA)
		If (EOA)=0, Go to P+1
		If (EOA) ≠ 0, Go to P
			5.1

*Optional – Part of High-Speed Arithmetic Option

Table 2-1. (Cont)
Summary Description of DDP-116 Instructions

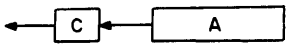
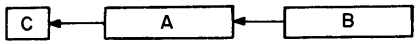
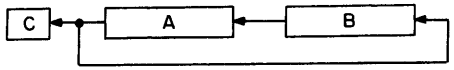
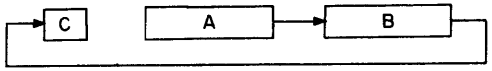
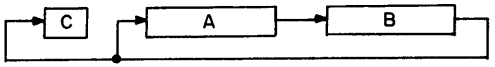
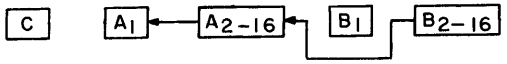
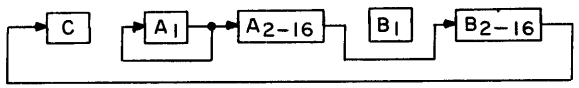
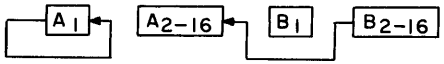
Mnemonic	Functional Description	Description	Time (μsec)
JMP	Unconditional Jump	EOA → (P)	1.7
JST	Jump and Store Location	$(P)_{3-16} \rightarrow (EOA)_{3-16}$ EOA + 1 → (P)	5.1
LDA	Load A	(EOA) → (A)	3.4
LGL	Logical Left Shift		1.7+0.34 N (N> 0) 2.04 (N=0)
LGR	Logical Right Shift		1.7+0.34 N (N> 0) 2.04 (N=0)
LLL	Long Left Logical Shift		1.7+0.34 N (N> 0) 2.04 (N=0)
LLR	Long Left Rotate		1.7+0.34 N (N> 0) 2.04 (N=0)
LRL	Long Right Logical Shift		1.7+0.34 N (N> 0) 2.04 (N=0)
LRR	Long Right Rotate		1.7+0.34 N (N>0) 2.04 (N=0)
LLS	Long Arithmetic Left Shift	 If A_1 changed 1 → (C) If A_1 not changed 0 → (C)	1.7+0.34 N (N>0) 2.04 (N=0)
LRS	Long Arithmetic Right Shift		1.7+0.34 N (N>0) 2.04 (N=0)

Table 2-1. (Cont)
Summary Description of DDP-116 Instructions

Mnemonic	Functional Description	Description	Time (μsec)
MPY*	Multiply	(A) x (EOA) → (A) (B) A - High order B - Low order	9.18
NOP	No Operation	Go to P+1	1.7
NRM*	Normalize	 If $(A)_1=1$, when $(A)_2=0$ terminate If $(A)_1=0$, when $(A)_2=1$ terminate N → (SCTR)	2.38+0.34 N
OCP	Output Command Pulse	$(IW)_{7-16} \rightarrow (ADB)_{7-16}$	3.4

OTA	Output from A	If (Ready)=0, Go to P, no output
		If (Ready)=1 (A)→ (OTB)
		Go to P+1
			5.1

RCB	Reset C-Bit	0 → (C)	1.7
SCA*	Shift Count to A	(SCTR) $\rightarrow (A)_{12-16}$ $0 \rightarrow A_{1-11}$	1.7
SCB	Set C-Bit	1 → (C)	1.7
SKS	Skip if Ready Line Set	If (Ready) = 0, Go to P If (Ready) = 1, Go to P+1	3.4
SLN	Skip if LSB of A Non Zero	If $(A)_{16} = 0$, Go to P If $(A)_{16} \neq 0$, Go to P+1	1.7

*Optional – Part of High-Speed Arithmetic Option

Table 2-1. (Cont)
Summary Description of DDP-116 Instructions

Mnemonic	Functional Description	Description	Time (μsec)
SLZ	Skip if LSB of A Zero	If $(A)_{16}=0$, Go to P+1 If $(A)_{16}\neq 0$, Go to P	1.7
SMI	Skip if A Sign Minus	If $(A)_{1}=1$, Go to P+1 If $(A)_{1}\neq 1$, Go to P	1.7
SNZ	Skip if A Non Zero	If $(A)_{1-16}\neq 0$, Go to P+1 If $(A)_{1-16}=0$, Go to P	1.7
SPL	Skip if A Sign Plus	If $(A)_{1}=0$, Go to P+1 If $(A)_{1}\neq 0$, Go to P	1.7 1.7
SR1	Skip if Sense Switch 1 Reset	If SS1=0, Go to P+1 If SS1=1, Go to P	1.7
SR2	Skip if Sense Switch 2 Reset	If SS2=0, Go to P+1 If SS2=1, Go to P	1.7
SR3	Skip if Sense Switch 3 Reset	If SS3=0, Go to P+1 If SS3=1, Go to P	1.7
SR4	Skip if Sense Switch 4 Reset	If SS4=0, Go to P+1 If SS4=1, Go to P	1.7
SRC	Skip if C-Bit Reset	If (C) =0, Go to P+1 If (C)=1, Go to P	1.7
SS1	Skip if Sense Switch 1 Set	If SS1=1, Go to P+1 If SS1=0, Go to P	1.7
SS2	Skip if Sense Switch 2 Set	If SS2=1, Go to P+1 If SS2=0, Go to P	1.7

Table 2-1. (Cont)
Summary Description of DDP-116 Instructions

Mnemonic	Functional Description	Description	Time (μsec)
SS3	Skip if Sense Switch 3 Set	If SS3=1, Go to P+1 If SS3=0, Go to P	1.7
SS4	Skip if Sense Switch 4 Set	If SS4=1, Go to P+1 If SS4=0, Go to P	1.7
SSC	Skip if C-bit Set	If (C)=1, Go to P+1 If (C)=0, Go to P	1.7
SSM	Set A Sign Minus	$1 \rightarrow (A)_1$	1.7
SSP	Set A Sign Plus	$0 \rightarrow (A)_1$	1.7
SSS	Skip if any Sense Switch Set	If $SS_{1-4} \neq 0$, Go to P+1 If $SS_{1-4} = 0$, Go to P	1.7
SSR	Skip if no Sense Switch Set	If $SS_{1-4} = 0$, Go to P+1 If $SS_{1-4} \neq 0$, Go to P	1.7
STA	Store A	$(A) \rightarrow (EOA)$	3.4
SUB	Subtract	$(A) - (EOA) \rightarrow (A)$ If Overflow $1 \rightarrow (C)$ If no Overflow $0 \rightarrow (C)$	3.4
SZE	Skip if A Zero	If $(A)_{1-16} = 0$, Go to P+1 If $(A)_{1-16} \neq 0$, Go to P	1.7
TCA	Two's complement A	$(\overline{A}) + 1 \rightarrow (A)$	1.7

Table 2-2.
Symbol Glossary

Symbol	Definition
EOA	Effective operand address is the address from which the operand will be obtained. This is only determined after all selection of sectors, indexing, and indirect addressing called for has been performed.
C	Carry flip-flop bit (C-bit)
IW	Instruction word
P	Address of next sequential instruction
N	Specifies length of multiply, divide, or shift operations; specifies block length of search or compare tables of operations; specifies number of words or characters to be transferred.
PIB	Program interrupt bit
INB	Input bus
OTB	Output bus
SCTR	Shift counter (used with multiply and divide instructions)
LSB	Least significant bit
( )	The contents of the register, counter, or memory location indicated within the brackets
$\longrightarrow$	Transfer to, becomes, or replaces
$\longleftrightarrow$	Exchanged
$\wedge$	Logical AND
$\vee$	Logical OR
$\veebar$	Exclusive OR

SECTION III
INPUT/OUTPUT

INPUT/OUTPUT CONTROL AND COMMUNICATION

The basic communication link between the DDP-116 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 or as few peripheral devices as are required may be attached to the bus. These devices then all communicate with the central processor by time sharing the bus. Since all DDP-116 I/O devices are individually buffered, and the bus is only used by a particular device while the computer is actually transferring information to or from the device, many devices can be operating simultaneously. The input/output bus lines are listed in Table 3-1.

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 computer to send out on the bus address 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 ready status, and the actual input or output can often be performed with a single instruction. As a rule, it is never necessary to separately select a device each time input or output is performed. This is especially important when many devices are being handled concurrently. 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 and OTA instructions.

Three basic modes of input/output are available with the DDP-116. The standard mode is single-word input/output transfer, with or without interrupt. The second mode is the optional DMC (direct multiplexed channel) which permits input/output to and from memory without program intervention. The third mode is the optional DDC (direct data channel), which can be used with special devices to achieve very high transfer rates.

Table 3-1.
Input/Output Bus Lines

Lines Available for Input/Output	Designation	Bit Capacity	Function
Output bus	OTB_{1-16}	16	Transmits data from the computer to an I/O device
Input bus	INB_{1-16}	16	Transmits data from an I/O device to the computer
Address bus	ADB_{7-10}	4	Bits 7 through 10 define the function to be performed by selected I/O command
	ADB_{11-16}	6	Bits 11 through 16 define the I/O device selected
Device ready line	DRL	1	Transmits a signal to the computer indicating that the device addressed by the I/O command is ready (device ready = gnd)
Output pulse line	OTP	1	Transmits an output pulse that defines the time at which the output bus is to be strobed on an OTA command
Output control pulse line	OCP	1	Transmits an output command pulse that defines the fact that an OCP command is being executed
Reset ready line	RRL	1	Transmits a pulse from the computer indicating that a data transfer has been made on INA or OTA
Master clear	MC	1	Transmits a master reset to devices when power is turned on to prevent spurious action by the device
Parity error	PE	1	Transmits a signal to the computer indicating that a parity error has been detected in an I/O device
Program-interrupt line	PIL	1	Transmits a signal to the computer indicating that an interrupt is requested
Program interrupt mask line	SMK	1	Transmits a predecoded output command from the computer indicating that the OTB contains a new setting for the interrupt mask flip-flop

Single-Word Transfer Mode

The single-word transfer under program control is the basic input/output mode of the standard computer. In this mode, full words or characters 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 (INA) instruction. If characters are being read into the computer, 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 DDP-116 extremely flexible in real-time applications. Thus, the computer 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 DDP-116, quite high data transfer rates can be accommodated in the single-word transfer mode. For example, input from the standard magnetic tape option can be performed at tape character rates in excess of 80 kc by utilizing 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 inserted 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 BCD).

The INA instruction sends out its device and function code on the I/O bus. It then looks for a ready signal on the DRL (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 RRL (reset ready line) to tell the device that the data has been accepted by the computer. 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 BCD).

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 DRL (device ready line). If a ready signal is received within a predetermined time interval, an output pulse is sent out on the OTP 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 OCP 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 OCP pulse is required to perform. The DRL 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 DRL. In the case of SKS, the test condition status specified by the function bits is placed on DRL. If an affirmative status signal is received within the prescribed time interval, the next instruction is skipped. If an affirmative status signal is not received, the next instruction is not skipped.

Standard Interrupt

The basic interrupt system consists of a single interrupt line. All standard I/O devices are connected to this line. Any number of additional 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 OTA '0020 instruction, which transfers the contents of the A-register via the OTB to the mask bits of standard devices as listed in Table 3-2. 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 used 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 an I/O bus, the logic associated with all the interrupt sources does not have to be centralized in a priority interrupt unit. The logic for each source 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 computer inhibits all further interrupts; it generates a jump and store location instruction (JST) indirectly through location $(30)_8$. 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 a test command (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. PIL. This ORs together interrupt request signals from all interrupt sources and sends them to the CPU.

b. DRL. This line is used by the SKS instruction to test each individual interrupt source 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 DRL.

c. SMK. 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 3-2.
Standard Interrupt Mask Assignments

OTB Bit No.	Device
1	TCU No. 1
2	TCU No. 2
3	TCU No. 3
4	TCU No. 4
5	I/O Channel No. 1
6	I/O Channel No. 2
7	I/O Channel No. 3
8	Unassigned
9	Paper Tape Reader
10	Paper Tape Punch
11	ASR-33
12	Card Reader
13	Card Punch
14	Line Printer
15	Memory Parity
16	Real Time Clock

ASR-33 TELETYPE UNIT (STANDARD)

The ASR-33 Teletype Unit (Figure 3-1) is the standard I/O device provided to communicate with the DDP-116. The ASR-33 is a versatile device providing a capability to read paper tape at 10 characters/second, and punch paper tape at 10 characters/second. The ASR-33 an also print out data from the DDP-116 at 10 characters/second and transfer data to the DDP-116 from the keyboard. In the local mode the unit can 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 73-character line. It also will automatically perform a carriage return and line feed when it reaches the end of a line. If the programmer wishes to print 71 or fewer characters, he must output the desired characters and then perform a carriage return and line feed in that order. If he wishes to print 72 characters, he must output the 72 characters and then a carriage return. The line feed will be performed automatically. However, the carriage return must be followed by a nonprinting character or a delay of one character time to avoid printing in the middle of the line while the carriage is returning. If 73 characters are to be printed, no carriage return or line feed is required. However, the next character must be nonprinting or a one character delay must be introduced to avoid printing during carriage return.

Keyboard Interlock

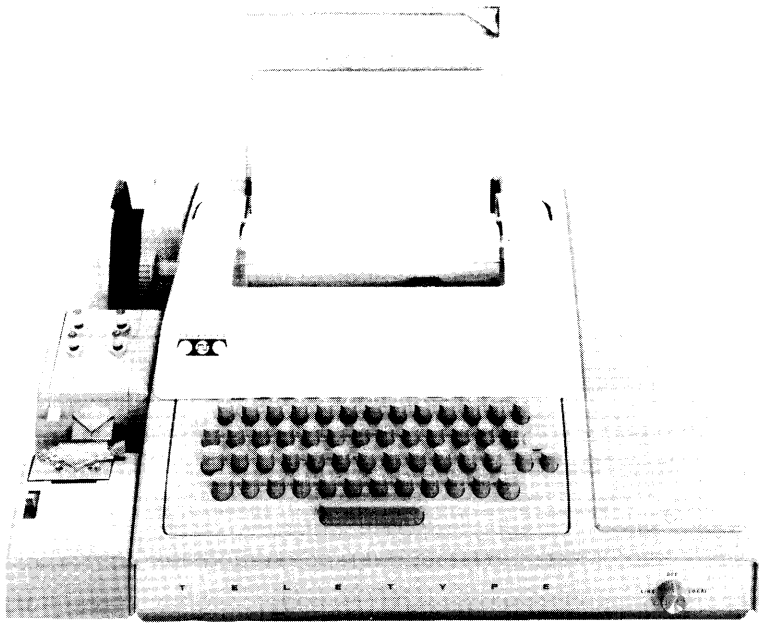
The ASR-33 keyboard is interlocked for all keys except the SHIFT, CTRL and REPT keys, preventing more than one key from being depressed at a time. The keyboard does not lock in the upper-case position. Therefore, the operator must hold the SHIFT key depressed to produce upper-case characters.

Tape Reader

Tape Format. -- See Figure 3-2.

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.



3083

Figure 3-1. ASR-33 Teletype Unit

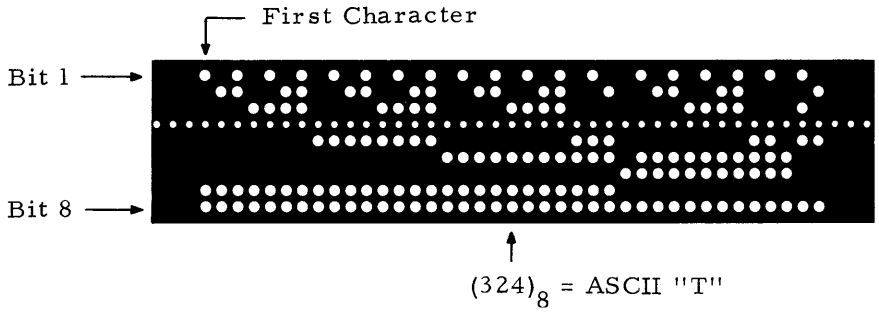


Figure 3-2. DDP-116 Tape Format

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 continute 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.

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

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 computer or from the reader to the computer. In either case, hard 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 computer. When an X-OFF, $(223)_8$, or $(023)_8$ is read, the reader will stop after reading the character following the X-OFF.

Output Mode. -- The output mode is used to transmit information from the computer to the ASR-33 printer or the printer and the punch. In either case, hard 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 computer will be punched whether it is printable or not. However, caution should be exercised since certain 8-bit codes -- $(221)_8$, $(021)_8$, $(005)_8$ and $(205)_8$ -- when transmitted from the computer will also cause a control action by the ASR-33 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 individual 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 computer or from the computer 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 and 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. The MASTER CLEAR pushbutton on the console will set up the ASR-33 input mode.

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.

OCP '0504 Set DMC ASCII Mode on ASR-33

This instruction is used when the DMC option (model 116-20) is connected to the ASR-33. It sets up the ASR-33 for a DMC transfer. As soon as the ASR-33 is ready, the first character of the transfer will be made. This instruction must be preceded by an OCP '0004 to set up the input mode. This instruction must precede an OCP '0104 to set up output mode. All transfers under DMC control with the ASR-33 are in ASCII mode.

OCP '0704 Reset DMC End of Transmission Interrupt on ASR-33

This instruction is used to reset the end-of-transmission interrupt signal associated with DMC stop interrupt option (model 116-20-2) when it is connected to the ASR-33.

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 computer in ASCII output mode or to present another character to the computer 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 computer 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 computer 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 computer. The stop code indication will remain present until the reader is restarted manually or (if the ASR-33 is switched to output mode) until the first OTA is given.

When a stop code is read by an ASR-33, the stop code and the following character will be transferred to the device buffer before the reader stops.

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

These instructions transmit 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 modifies channel 7 to form a valid ASCII alphanumeric character. To do this, bit 7 is made the inverse of bit 6. Thus, if the 8 least significant bits in the A-register were $(XX1XXXXX)_2$ they would be transmitted to the ASR-33 as $(X01XXXXX)_2$. If they were $(XX0XXXXX)_2$, they would be transmitted as $(X10XXXXX)_2$.

MEMORY PROTECT

A standard feature of the DDP-116 is memory protection against ac line power failure. If the ac power line fails or undergoes a transient which might cause computer failure while the computer is running, the computer will detect the power failure and halt without destroying any memory locations. This feature is also operative if the power ON/OFF button is depressed to the OFF position while a program is running. If the program does not resume when the START pushbutton is depressed, power must be turned off, then on.

SECTION IV
OPTIONS

PRIORITY INTERRUPT OPTION, MODEL 116-25/26

A multilevel priority interrupt system is available as an option. This option eliminates the need for an interrupt service routine to test for which one of the available interrupt lines caused an interrupt. A unique memory location is assigned to each interrupt line. These locations perform the same function for each line as the standard interrupt location 30_8 does for the basic interrupt system. When an interrupt occurs in this system, the computer generates a jump and store location instruction (JST) indirectly through the standard memory location associated with the source of the interrupt. The priority interrupt option also generates for each interrupt line an acknowledge signal, which indicates that the interrupt request has been acted upon. An acknowledge line can be used by an interrupting source to reset the interrupt request. Included in the system is a program-controlled mask register, which permits individual interrupt lines to be enabled and disabled under program control. This allows priority interrupts to be pyramided on top of other interrupts without an extensive executive routine.

The interrupt option is provided in groups of eight interrupt lines except for the first group, which handles seven interrupts plus the standard interrupt line. Up to eight groups or a total of 64 interrupt lines can be handled by the system. The interrupt lines are consecutively numbered, and have decreasing priority with increasing number. The standard interrupt line is designated line 1 and retains its standard location. The standard locations for the optional interrupt lines are shown in Table 4-1.

Table 4-1.
Standard Locations for First Eight Groups of Interrupt Lines

Priority Interrupt Group	Number of Lines	Standard Locations (Octal Codes)
1	7*	00031 to 00037
2	8	00040 to 00047
3	8	00050 to 00057
4	8	00060 to 00067
5	8	00070 to 00077
6	8	00100 to 00107
7	8	00110 to 00117
8	8	00120 to 00127

*8 including the standard interrupt line (standard location 30_8)

The mask bits associated with each group of interrupt lines are controlled by an OTA '0X20 instruction. This instruction sets the appropriate bit in the mask register if the corresponding bit in the A-register is a ONE and resets the mask register bit if the corresponding A-register bit is a ZERO. The OTA '0X20 instruction differs from other OTA instructions in that it is unconditional: it always executes its function and the following instruction is never skipped. Table 4-2 shows the mask assignments for the optional interrupt lines and the instructions that service them.

Table 4-2.
Priority Interrupt Mask Assignments

A-Register Bit Number	Interrupt Line Number			
1	1	17	33	49
2	2	18	34	50
3	3	19	35	51
4	4	20	36	52
5	5	21	37	53
6	6	22	38	54
7	7	23	39	55
8	8	24	40	56
9	9	25	41	57
10	10	26	42	58
11	11	27	43	59
12	12	28	44	60
13	13	29	45	61
14	14	30	46	62
15	15	31	47	63
16	16	32	48	64

Interrupt Line	Mask Instruction
1 - 16	OTA '0120
17 - 32	OTA '0220
33 - 48	OTA '0320
49 - 64	OTA '0420

It should be noted that when an interrupt option is added to a basic machine it is necessary to set up two mask bits in order to control any standard I/O devices utilizing the standard interrupt line. The first mask bit is that associated with the device, and the second that associated with line 1 of the interrupt option.

DIRECT MULTIPLEX CHANNEL OPTION

The direct multiplex channel (DMC) option permits data transfer between peripheral devices and computer memory concurrently with computation. Two modes of operation are available. One permits time sharing of input/output operations simultaneous with the computer program. The other, the hog mode, is available for servicing I/O devices which have such a high transfer rate that concurrent operation with the computer program is not feasible. When the hog mode is called for, the CPU is used exclusively to service the I/O device until the block transfer is completed.

The DMC operates by interrupting the computer between instructions in a manner similar to program interrupt. A single word is transferred between the computer memory and the I/O device on each interrupt. This process is repeated each time the I/O device indicates that it is ready until the required number of words has been transferred. When this occurs, the computer sends a stop signal to the device. The starting location -1 and final location for the block of data are stored in a pair of standard memory locations for each DMC subchannel. Up to eight devices can be connected to the DMC system simultaneously, independently transferring data between each device and a specified block of memory. A total of 6.8 μsec of computer time is required for each word transferred. This permits a maximum rate of over 145 kc in the hog mode. At slower transfer rates any time not used in data transfer is available for computation.

The standard locations assigned to different subchannels are listed in Table 4-3.

The format for setting up the DMC is shown in Figure 4-1. Even memory cells 10_8 through 26_8 contain the starting address minus one in bits 3 through 16. Bit two designates time-sharing mode when it is a zero and block-transfer or "hog" mode when it is a one. Bit one designates output mode when it is zero and input mode when it is one. Odd memory cells 11_8 through 27_8 contain the final addresses in the data block. Bits 3 through 16 designate the address. Bits one and two must agree with the mode bits of the even location associated with each channel. When a DMC channel is set up by properly loading its two address cells, it is necessary only to activate the device on the channel with an OCP in order to begin DMC operation. The additional signals in the I/O bus required for DMC operation are shown in Table 4-4. All other signals required to execute a DMC operation are identical to those used with the basic input/output commands.

If more than one device requests a DMC operation at the same time, the one with the numerically lower subchannel number is serviced first. When this operation is completed the lowest numbered unserviced request is processed. This sequence is repeated until no DMC request is present. The computer then proceeds to the next instruction. If priority interrupt requests and DMC requests are present at the same time, the DMC requests are serviced before the priority interrupt requests.

A DMC subchannel interrupt option (116-20-2) is also available with a basic DMC subchannel. This option produces an interrupt request on the standard interrupt line when the final location of the transfer block in memory associated with the particular subchannel is filled.

The specific instructions necessary to operate any standard DDP-116 I/O device in DMC mode are discussed in Section V.

Table 4-3.
DMC Standard Locations

DMC Channel	Octal Location
1	00010 00011
2	00012 00013
3	00014 00015
4	00016 00017
5	00020 00021
6	00022 00023
7	00024 00025
8	00026 00027

Table 4-4.
Additional I/O Bus Lines Required for DMC Option

Line	Designation	Bits	Description
Device Interrupt Lines	DIL01 through DIL08	1	A single DIL line is assigned to each DMC channel. It transmits a signal to the CPU indicating that a DMC data transfer is requested.
Device Address Lines	DAL01 through DAL08	1	A single DAL line is assigned to each DMC channel. It transmits a signal from the CPU to the device indicating that a DMC data transfer is to take place.
End of Range Line	ERL	1	Transmits a signal indicating that the end of the block of data assigned to the DMC channel has been reached and that no further interrupt requests can be issued by the device.
External Stop	STOP	1	Accepts an external stop signal from device which terminates hog mode when device data block is less than range setting of DMC channel.

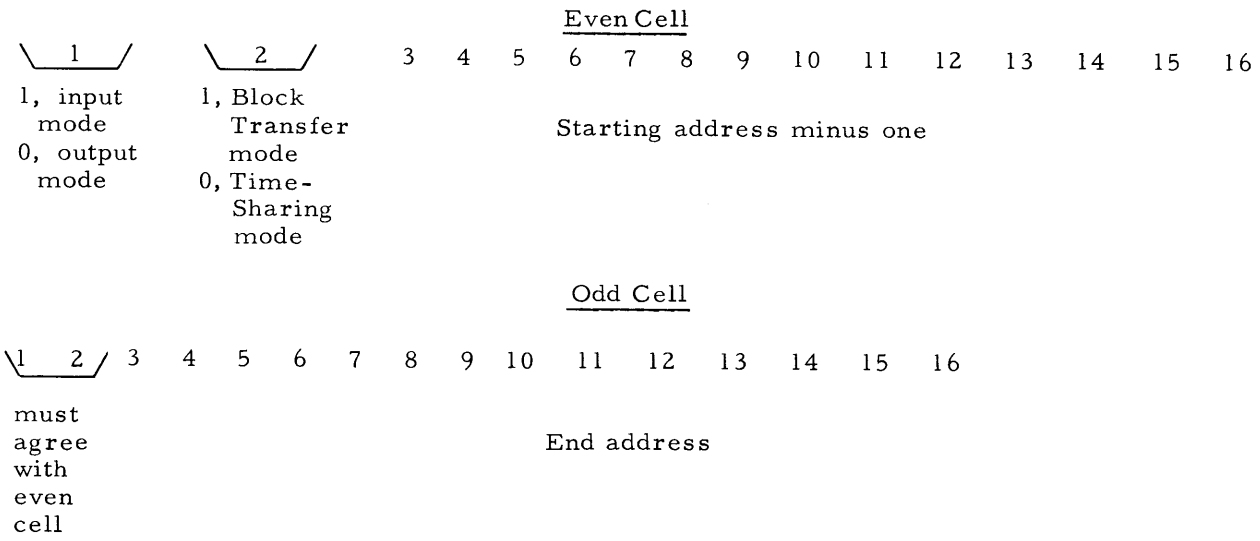


Figure 4-1. DMC Control Word Format

REAL TIME CLOCK, MODEL 116-12

The real time clock option permits the program to keep track of real time. This is done by automatically incrementing location 7 in the computer memory once every 16.67 ms. Whenever the memory location overflows from $(177777)_8$ to $(000000)_8$, an interrupt is produced, occurrence of this interrupt can be tested with an SKS instruction. The interrupt can be masked on or off by the use of an OTA instruction and bit 16 in the A-register.

Applications

By use of a suitable interrupt subroutine, the program can keep track of seconds, minutes, hours and days by monitoring and counting the interrupts generated by the real time clock. A typical example would be the case where one wished to keep track of minutes after a particular action had been initiated by the computer. To do this, the program would load $(-3600)_{10}$ or $(170760)_8$ into location 7 and mask on the real time clock interrupt. Every time an interrupt occurs, a location in the appropriate subroutine would be incremented and $(170760)_8$ restored into location 7. The subroutine location would then at all times indicate the number of minutes since the event of interest had occurred.

The real time clock location is incremented as long as the computer is running. Every 16.67 milliseconds a forced IRS will be executed which interrupts processing for three cycles, or 5.1 μsec. This action is not affected by the INH or ENB instructions. It also continues after a real time clock interrupt. Incrementing will not occur while the computer is in the halt mode. The real time clock interrupt which occurs when location 7 overflows can be inhibited both by an INH instruction or by resetting the interrupt mask flip-flop of the real time clock. In the case where location 7 overflows while interrupt is inhibited for either of the two above-mentioned reasons, the interrupt request will remain until it is reset by the proper OCP. Location 7 will continue to be incremented

while the interrupt request is waiting. DMC requests and priority interrrupts have precedence over real time clock incrementing requests; however, the incrementing request will not be lost. An exception to this is operation of the DMC in the hog mode which will block incrementing of the real time clock. If the computer remains in the hog mode for more than 16 ms, one or more incrementing requests may be lost.

Sensing and Control

The following instructions are concerned with the real time clock option.

SKS '0020 Skip If Real Time Clock Not Interrupting

This instruction is used to determine whether the real time clock is the source of an interrupt occurring on the standard interrupt line.

OTA '0020 Set Mask

This instruction is used to set or reset the interrupt mask bits of all standard DDP-116 options. When this instruction is executed, bit 16 of the A-register controls the interrupt mask flip-flop of the real time clock. A ONE in bit 16 will set the mask; a ZERO in bit 16 will reset the mask; when the mask is set, real time clock interrupts are enabled.

OCP '0020 Reset Real Time Clock Interrupt

This instruction resets the real time clock interrupt.

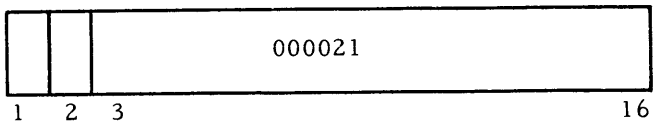
MEMORY PARITY, MODEL 116-07, 07-1

The memory parity option in the DDP-116 permits a parity check to be made on the operation of a DDP-116 memory module. A separate memory parity option (model 116-07 or model 116-07-1) must be used with each 4K or 8K memory module in a DDP-116 system. However, all memory modules make use of a common memory parity error flip-flop. The memory parity option provides facilities for checking parity on all memory read cycles and for generating parity on all memory write cycles. The computer's memory parity error flip-flop can be tested and reset under program control. The setting of the parity error flip-flop lights the parity check (PC) indicator on the DDP-116 console, and also generates an interrupt on the standard DDP-116 interrupt line. This interrupt can be tested under program control, and it can be masked on or off by a suitable OTA instruction.

The memory parity option adds the following instructions to the DDP-116 repertoire.

Reset Memory Parity Error

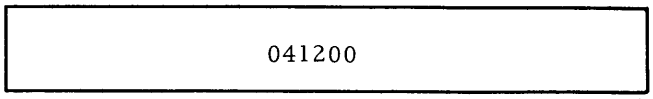
RMP 1.7 μsec



DESCRIPTION: This instruction resets the memory parity error flip-flop (and the PC indicator).
TYPE: Generic

Skip on Memory Parity Error

SPS 1.7 μsec

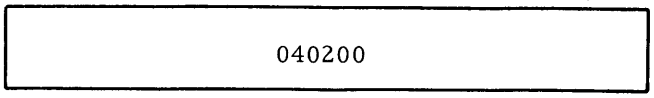


DESCRIPTION: This instruction skips the next instruction if the parity error flip-flop is set.

TYPE: Generic

Skip on No Memory Parity Error

SPN 1.7 μsec



DESCRIPTION: This instruction skips the next instruction if the parity error flip-flop is reset.

TYPE: Generic

OTA '0020 Set Mask

This masking instruction sets or resets the mask bits of priority interrupt sources which are tied to the standard DDP-116 priority interrupt line. When this instruction is executed, A-register bit 15 controls the setting or resetting of the memory parity interrupt mask flip-flop. When A_{15} is a ONE, the mask is set, enabling memory parity errors to interrupt the DDP-116; when A_{15} is a ZERO, the memory parity error interrupt mask will be reset, inhibiting memory parity errors from interrupting the computer. The master clear button on the DDP-116 console also resets the memory parity error flip-flop.

NOTE

The instruction following an OTA '0020 is never skipped.

OPTIONAL INPUT AND OUTPUT CHANNELS, MODELS 116-32/33/34

Three varieties of optional input and output channels are available for interfacing with nonstandard devices. All three channels operate from the standard DDP-116 I/O bus in the single word transfer mode described in Section III. Connection to the basic interrupt line is standard, and all can be adapted to operate with the DMC option by the addition of a standard subchannel.

Parallel Input Channel (Model 116-32)

The parallel input channel provides a basic control interface for transferring parallel data to the DDP-116 input bus by INA command. Since the data is unbuffered, the transfer must be made during the time the data is available from the device.

In addition to the input transfer capability, the parallel input channel also provides three OCP lines for user-assigned control functions and two extra SKS lines for testing of device status.

Parallel Output Channel (Model 116-33)

The parallel output channel provides a basic control interface for transferring parallel data from the DDP-116 output bus to a nonstandard device by OTA command. Since the data is unbuffered, the device must accept the data during the OTP pulse interval (approximately 3.0 μsec for standard I/O transfers or 1.0 μsec for DMC transfers).

In addition to the output transfer capability, the parallel output channel also provides three OCP lines for user-assigned control functions, and two extra SKS lines for testing of device status.

Parallel Buffered I/O Channel (Model 116-34)

The parallel buffered I/O channel combines the data transfer features of the two preceding channels and includes a 16-bit buffer register.

In the output mode, data is transferred from the DDP-116 output bus to the buffer register by OTA command. A ready flip-flop is reset at the time of loading. The interface will not accept another word until the device accepts the data and sets the ready flip-flop with the trailing edge of its load strobe. The condition of the ready flip-flop can be tested with an OTA or SKS command. (If the standard interrupt is masked on, the ready condition will cause an interrupt request.)

In the input mode, data is loaded from the device to the buffer register by a drop-in pulse from the device. The trailing edge of the drop-in pulse sets the ready flip-flop, which can be tested by an INA or SKS instruction. (If the standard interrupt is masked on, the ready condition will cause an interrupt request.)

When the channel is used for both input and output transfers, the direction of transfer must be set up by an input or output device select OCP.

Three OCP lines are supplied with the channel for assignment by the user according to the control requirements of the device or system, and two extra SKS lines for testing of device status.

Sensing and Control

The transfer, control, and sense instructions defined below apply to all three types of channel, except where otherwise specified.

NOTE

The last octal digit of the channel address can be varied to provide unique addresses for up to three channels per system. Code assignments are:

Channel	Last Octal Digit (X)
First in a system	4
Second in a system	5
Third in a system	6

INA '103X Clear A-Register and Input Data If Ready

This instruction applies to either a parallel input channel or the buffered I/O channel in input mode. Ready is interrogated; if ready, the A-register is cleared, the contents of the input bus is ORed to the A-register, and the next sequential instruction is skipped, if not ready, the A-register is unchanged and the next sequential instruction is executed.

Before this instruction is addressed to a buffered I/O channel, the channel must be set up for input mode by an input device select OCP (OCP '013X, '023X, '043X).

INA '003X Input Data If Ready

This instruction is identical to INA '103X except that the A-register is not cleared.

OTA '003X Output to Parallel Channel If Ready

This instruction applies either to the parallel output channel or to the buffered I/O channel in output mode. Ready is interrogated if ready, data from the A-register is applied to the output bus and strobed to the interface by an OTP pulse, and the next sequential instruction is skipped; if not ready, the next sequential instruction is executed.

Before this instruction is addressed to a buffered I/O channel, the channel must be set up in the output mode by an output select OCP command (OCP '013X, '023X, '043X).

OCP '013X Optional OCP Line No. 1

OCP '023X Optional OCP Line No. 2

OCP '043X Optional OCP Line No. 3

These lines are made available for assignment by the user.

OCP '003X Enable DMC Mode

This instruction sets up the DMC mode for a channel equipped with the DMC subchannel option (model 116-20-1). Once enabled by this command, the channel demands a DMC data transfer every time the device is ready. When the DMC reaches the end of range, the DMC mode is disabled. This instruction must precede an OCP output select instruction and must follow an OCP input select instruction.

OCP '073X Reset DMC Stop Interrupt

In a channel equipped with the DMC stop interrupt option (model 116-20-02), this instruction resets the interrupt that occurs at the end of range.

SKS '003X Skip If Channel Is Ready

This instruction tests the condition of the channel ready line; if ready, the next instruction is skipped.

SKS '013X Skip If Optional Sense Line No. 2 Is Set

SKS '023X Skip If Optional Sense Line No. 1 Is Set

These instructions test the condition of two user-assigned sense lines. If the addressed line is set, the next instruction is skipped.

SKS '043X Skip If Channel Not Interrupting

This instruction tests whether the addressed channel is generating an interrupt. A standard interrupt occurs when the channel is ready and the mask flip-flop is set by an OTA '0020 instruction. If equipped with the DMC stop interrupt option (model 116-20-2), the channel will interrupt when the mask is set and the end of range is reached.

OTA '0020 Set Interrupt Masks

All standard I/O channels respond to the standard mask-setting instruction. Mask bit assignments for multichannel systems are as follows:

Channel No.	A-Register Bit
1	5
2	6
3	7

DIRECT DATA CHANNEL, MODEL 116-21

The direct data channel (DDC) is a high-speed data channel capable of achieving input/output data transfer rates in excess of 500 kc. It is capable of operation in a time-sharing mode with the basic computer and the DMC option, or in a block transfer mode during which all other system operations are suspended. When the DDC requests access, the computer is interrupted at the completion of the current instruction. When data transfer has been completed, the computer resumes operation at the point where it was interrupted.

Program control of the DDC is accomplished entirely through the device with which it is interfaced. Such a device must have address and range registers plus input/output mode and block transfer mode control lines, all of which may be conditioned by OCP instructions.

The DDC requires 3.4 μsec for the first word transferred after interrupting the computer and 1.7 μsec for each additional word transferred before control is returned to the program.

POWER FAILURE INTERRUPT, MODEL 116-09

The power failure interrupt (PFI) option is designed to detect the onset of an ac line power failure and, during the 5 ms available before dc power fails, to cause a program interrupt. The interrupt initiates a subroutine that stores the contents of all volatile registers in the computer and initiates an orderly shutdown of any equipment connected to the computer in a real-time system. (Equipment is not shut down automatically by

the PFI option alone.) If the interrupt subroutine terminates with a halt instruction within 5 ms of the failure, no data in the magnetic core memory is destroyed. Both the program that was present before power failure and any information stored by the interrupt subroutine are available to the program when power is restored. The program can continue where it left off, roll back, or be re-initialized in a manner suitable to the particular application.

The interrupt generated by the power failure causes a forced indirect JST to location $(000006)_8$. The interrupt overrides the status of the permit-interrupt control flip-flop (which is normally controlled by the interrupt enable (ENB) and interrupt inhibit (INH) instructions.) Once power failure interrupt occurs, the permit-interrupt control flip-flop should not be enabled since this would cause another forced JST to location $(000006)_8$.

The option utilizes an S-374 Power Failure Sense PAC which senses for ac line failure on the ac feed to the computer main frame. The PACs sense threshold can be adjusted from 95 to 120 vac; the interrupt is generated when line voltage falls within one percent of the preset voltage.

Power failure interrupts are generated when power is turned off at the front panel POWER switch, as well as from power failures.

To start the DDP-116 after a PFI has caused shutdown, the console POWER switch must be turned OFF, then ON.

WATCHDOG TIMER, MODEL 116-13

The watchdog timer option is an error-detecting device that sounds an audible alarm and provides relay contact closures when the time lapse between two programmed events exceeds a predetermined limit.

The device contains a 12-stage down-counter which can be preset to a specified value (up to octal 7777) by an OTA instruction. The counter is decremented by a signal developed from the 60-cps line and can be set up in increments of 16.67 ms up to a maximum range of 68.3 seconds. Worst-case accuracy is ±1 count pulse. If the counter is not loaded with a new range before it counts down to $(0000)_8$, the alarm circuit is energized.

Applications

The watchdog timer is typically used in the initialization loop of a system program to ensure valid operation. Since the total time required to complete the program can be determined, the watchdog timer is loaded with one count more than the approximate time to complete the loop. If the computer hangs up in a loop or comes to an erroneous halt, the alarm sounds, noting the failure. Unlike parity or other common forms of fault detection, this type of validity check relates directly to the total timing of an instruction loop.

Alarm Circuits

The audible alarm is generated by two 3C Sonalerts® each capable of emitting an 80-db signal at 2 kc. Both sides of a normally open alarm relay contact are brought out to

an external terminal block. When the alarm sounds, this contact is closed. There is an interlock with the computer power circuits, so that the alarm will sound if there is a power failure in the area where the watchdog timer is wired.

An override switch located on the alarm box will inhibit the alarm. An override relay coil is brought out to an external terminal block. If this relay is pulled in (by -6 vdc at 40 ma), the alarm system will be inhibited.

Sensing and Control

The following instruction sets up the time limit of the alarm.

OTA '0021 Output to Watchdog Timer

This instruction transfers A-register bits 5-16 to the down-counter. Bit 16, the least significant bit, corresponds to a 16.67 ms time increment. This instruction always skips.

MEMORY EXPANSION, MODEL 116-15

Memory expansion above 16K in the DDP-116 is handled by increasing the size of the program counter to 15 bits and the introduction of the extend mode. The added program counter bit (P02) provides the fifteenth bit of the 32K address field and conditions Y02 when the sector being addressed is other than 0. The extend mode changes the interpretation of the index bit of the indirect address word, which becomes part of a 15-bit indirect address. In the extend mode just one level of indexing is possible. It is specified by bit 2 of the instruction word and is always the final operation in generating the effective operand address. Figure 4-2 illustrates the operation of a 32K system.

The extend mode is set or reset by generic instructions. It is also set by the occurrence of a program interrupt. An additional control flip-flop called the previous mode indicator (PMI) is also added to the mainframe. This is used to store information as to which mode the program was operating in when a program interrupt occurred. The PMI bit is set if the CPU is in the extend mode when a priority interrupt occurs.

The PMI is reset if:

a. the computer is not in the extend mode when a priority interrupt occurs, and
b. a master clear is executed.

JST Instruction Modification

The extend mode alters JST to allow it to store a 15-bit program counter. Bit 1 of the memory location specified by the effective operand address is left unchanged.

Additional Instructions

The following additional instructions are utilized with memory expansion.

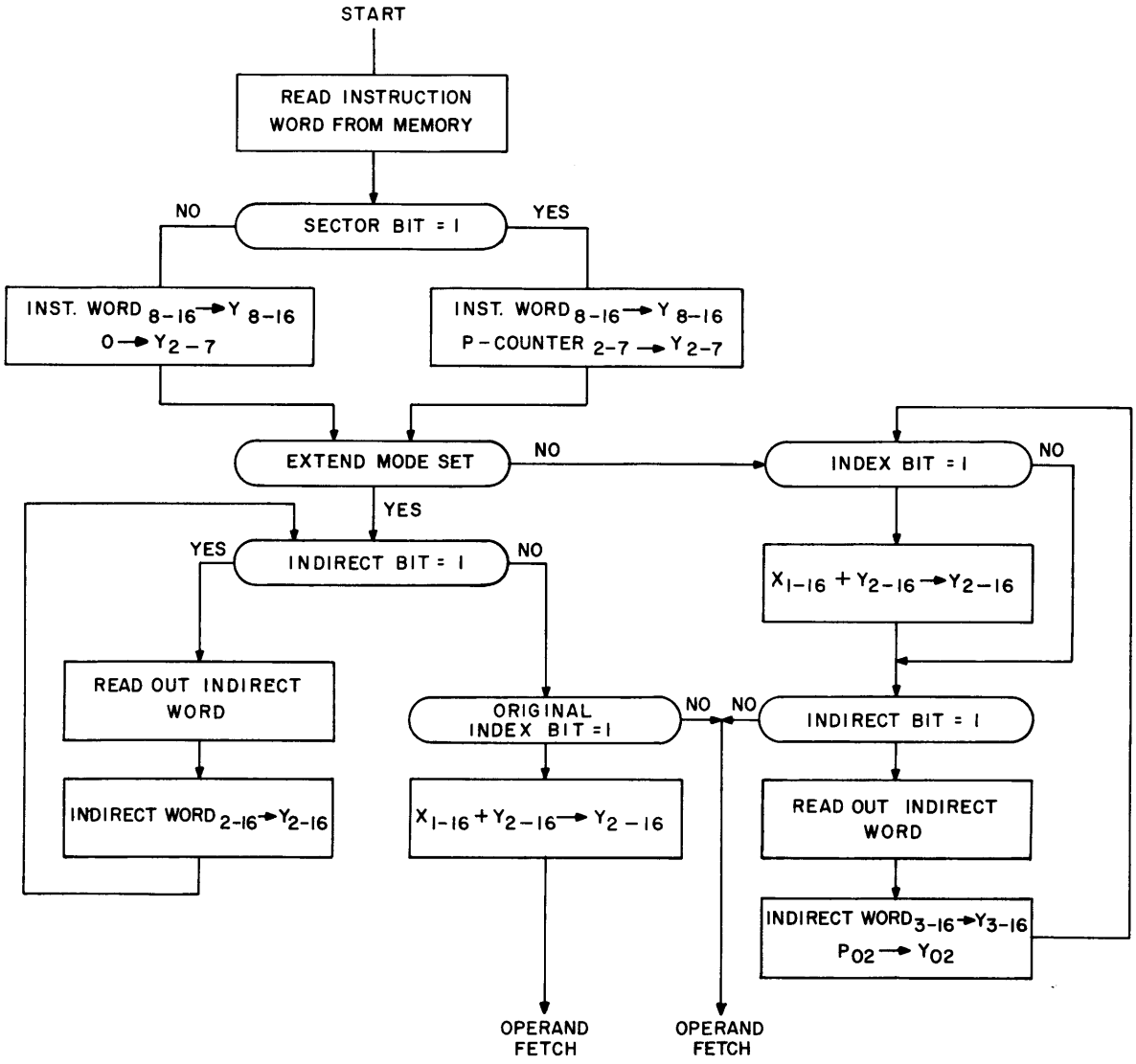
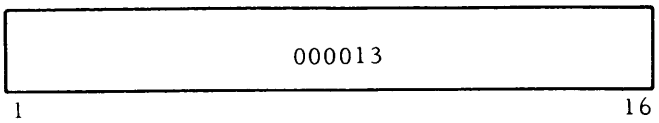


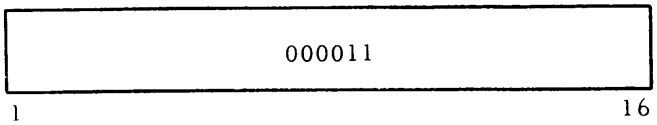
Figure 4-2. Alteration in Phase Sequence for 32K Memory Expansion

EXA Enable Extended Addressing



This instruction plaes the computer in the extend mode. Execution time is 1.7 μsec.

DXA Disable Extended Addressing



This instruction restores the normal mode. When the instruction is executed, a JMP instruction must be performed in the existing mode before the normal mode takes effect. Any number of non-JMP instructions may be included between the DXA and the first JMP instruction. The purpose of this feature is to allow the computer to exist from an interrupt subroutine.

SKS '0120

This instruction skips the immediately following instruction if the PMI is set. If the PMI is reset, it executes the next instruction in sequence. Execution time is 3.4 μsec.

SECTION V
PERIPHERAL DEVICES

HIGH-SPEED PAPER-TAPE READER, MODEL 116-50

A unidirectional perforated-tape reader (Figure 5-1) is available as an option. The unit reads eight data channels plus a sprocket hole channel, at the rate of 30 in./sec (10 characters per inch, or 300 characters per second).

The unit uses standard paper or mylar tapes (black paper recommended) 0.004 to 0.005 in. thick. The tape can be loaded without removing power by rotating a front-mounted READY-LOAD switch clockwise to the LOAD position. After the tape has been loaded, the READY-LOAD switch must be rotated counter-clockwise to the READY position.

Reader Modes

The high-speed paper-tape reader operates in the continuous mode at a rate of 300 characters per second. Reader start is initiated by an OCP command, after which the reader will continuously transfer characters to its buffer until the complete tape has been read or an OCP stop-reader command is executed.

Codes

Eight-level code will be read into the device buffer. The reader is passive and will transfer all eight-level bit configurations into the computer (including tape leader). The reader hardware will not interpret any code as a stop or delete code.

Specifications

The paper-tape reader has the following characteristics.

a. Reads at speed up to 300 cps

b. Start time 5 ms

c. Reads paper or mylar tape 0.004 to 0.005 in. thick. Can be adjusted to read tapes from 0.0025 to 0.008 in.

d. Density of 10 characters per inch.

Sensing and Control

The following instructions are used for communication between the reader and the computer.

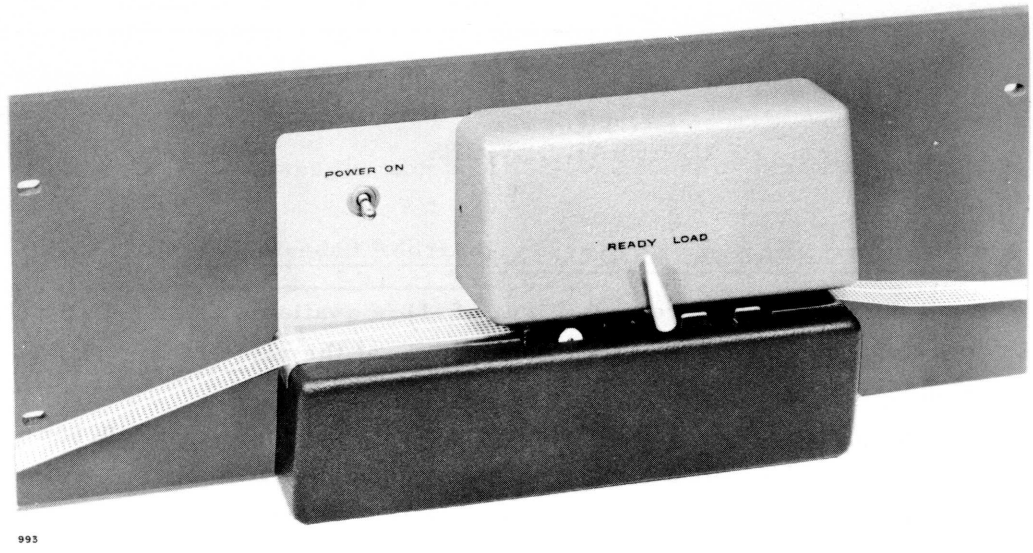


Figure 5-1. High-Speed Paper-Tape Reader, Model 116-50

OCP '0001 Start Reader

This instruction starts tape motion. The first character to pass the read station is transferred to the interface buffer for transmission to the DDP-116.

OCP '0101 Stop Reader

This instruction stops tape motion. In order to avoid losing a character after a restart, this OCP must be executed within 1.0 ms of a character-ready signal.

OCP '0301 Set DMC Mode

For an interface equipped with the DMC subchannel option (model 116-20-1), this instruction enables DMC data transfers. An OCP '0001 must precede this OCP.

OCP '0701 Reset Stop Interrupt

For an interface equipped with the DMC stop interrupt option (model 116-20-2), this instruction resets the interrupt generated when end of range is reached.

SKS '0401 Skip If Tape Reader Not Interrupting

The tape reader is interrupting when a character is ready and the interrupt mask flip-flop is set.

SKS '0001 Skip If Tape Reader Ready

The tape reader is ready when a character is available in the interface buffer.

INA '0001 Input from Paper Tape Reader If Ready

Execution of this instruction causes a character to be ORed into the eight least significant bits of the A-register and causes the next program instruction to be skipped.

INA '1001 Clear A-Register and Input from Paper Tape Reader If Ready

This instruction is identical to INA '0001, but the A-register is cleared before the character is transferred in.

OTA '0020 Set Interrupt Masks

This instruction sets the mask bit if the corresponding bit of the A-register is a ONE and resets the mask bit if the corresponding bit is a ZERO. The instruction is unconditional and never skips. A-register bit 09 controls the paper-tape reader interrupt mask flip-flop.

HIGH-SPEED PAPER-TAPE PUNCH, MODEL 116-52

The paper tape punch (Figure 5-2) is capable of punching 1-inch, 8-level tape at a rate of up to 110 characters per second. Power to the punch is controlled by the DDP-116 program. After a power-on command is initiated, the interface prevents data transfers for a 3-second delay period to allow the punch mechanism to come up to full operating speed.

The punch is a synchronous unit; pulses generated by a magnetic pickup coil synchronize the interface control circuits.

Eight levels of data are transferred to the punch from the DDP-116 output bus, bits 9 through 16. Tape tracks 1 through 8 represents output bus bits 16 through 9, respectively.

Specifications

The Model 116-52 Paper-Tape Punch has the following characteristics:

a. An 8-inch, NAB hub take-up reel is provided with the unit.

b. The perforator drive is enclosed in a buffered oil case which results in quiet operation and little maintenance.

c. Punched oiled paper, foil or mylar tape.

d. Paper supply of 1000 feet.

e. Tape width of 1.000 in.

f. Density of 10 characters per in.

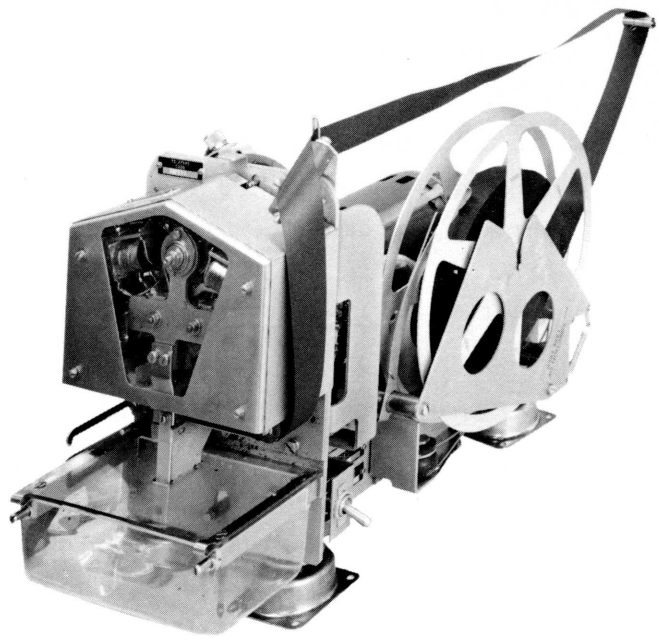


Figure 5-2. High-Speed Paper-Tape Punch, Model 116-52

Sensing and Control

The following instructions are used for communication between the punch and the computer.

OCP '0002 Enable Paper Tape Punch

This instruction turns on power to the punch. A 3-second delay elapses before the punch is ready to accept a character. If DMC operation is desired, an OCP '0202 must precede this OCP.

OCP '0102 Turn Punch Power Off

This instruction removes punch power. Before executing this instruction, however, the punch must be sensed for a ready condition to avoid turning the punch off while a character is being punched.

OCP '0202 Set DMC Mode

For an interface equipped with the DMC subchannel option (model 116-20-1), this instruction enables DMC data transfers. This instruction must precede an OCP '0002.

OCP '0402 Reset Stop Interrupt

For an interface equipped with the DMC stop interrupt option (model 116-20-2), this instruction resets the interrupt that occurs at end of range.

SKS '0002 Skip If Paper Tape Punch Is Ready

The punch is ready when its buffer is empty and ready to accept new data from the computer.

SKS '0402 Skip If Paper Tape Punch Is Not Interrupting

The tape punch is interrupting when its interrupt mask flip-flop is set and its buffer is ready to receive a character.

SKS '0102 Skip If Punch Power Is On

This instruction must precede an OCP '0002. If the latter is executed when power is already enabled, the loss of a character might occur.

OTA '0002 Output to Tape Punch If Ready

This instruction performs the output transfer; if ready, the 8 least significant bits of the A-register are transferred to the punch interface, the next program instruction is skipped, and a punch cycle is started. (During a punch cycle, the interface responds to this instruction or the SKS '0002 as not ready.)

OTA '0020 Set Interrupt Masks

This instruction sets a mask bit if the corresponding bit of the A-register is a ONE and resets the mask bit if the corresponding bit is a ZERO. The instruction is unconditional and never skips. The paper tape punch interrupt mask flip-flop is controlled by A-register bit 10.

CARD READER, MODEL 116-60

The standard punched card reader (Figure 5-3) available as an optional I/O device for the DDP-116 computer is capable of reading 100 cards per minute (nominal); its hopper has a capacity of 430 cards.

Modes of Operation

Two modes of operation accommodate cards punched either in Hollerith or binary format.

The Hollerith code (see Table 5-1) divides the card into 80 columns of 12 bits, each column containing the code for one character. When reading Hollerith-coded cards, the unit reads the 12-bit characters one by one and converts them to a 6-bit Hollerith code. These 6-bit

characters are transferred onto the input bus such that the low order bit goes into bit 16 and the high order bit goes into bit 11.

In the binary mode each column sends 12 bits of data into the computer as one column. Rows 12, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 are transferred respectively into A-register, bits 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 with an INA '0005.

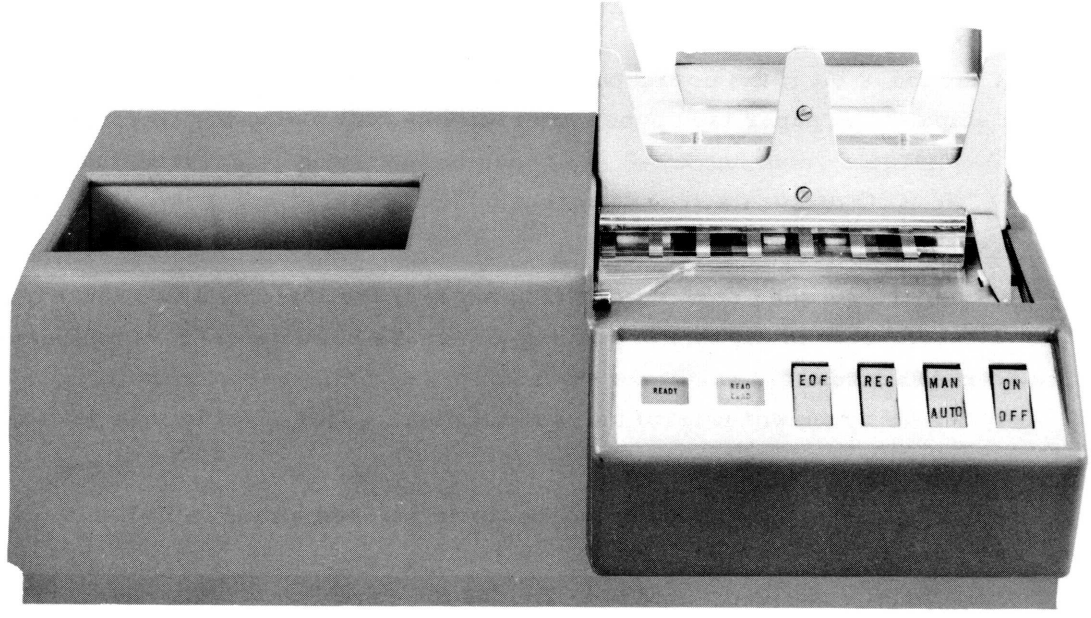
Specifications

Serial Feed	Reads cards column by column
Speed	88-90 cards per minute nominal
Hopper Capacity	430 cards
Card Type	Square cornered cards are recommended for use with this unit

Timing

The ready flip-flop is set approximately 17.5 ms after initially feeding a card and every 7 ms thereafter.

The card reader remains busy for approximately 600 ms after receiving an OCP to feed a card. When clutching the last card in the hopper, the card reader goes to a non-operational state approximately 17.5 ms after reading the character contained in column 80. SKS '0305 will not skip at this time.



A3078

Figure 5-3. Card Reader, Model 116-60

Table 5-1.
Card Code

Hollerith*	ASCII	Card	Character	Hollerith	ASCII	Card	Character
00	260	0	0	40	255	11	-
01	261	1	1	41	312	11-1	J
02	262	2	2	42	313	11-2	K
03	263	3	3	43	314	11-3	L
04	264	4	4	44	315	11-4	M
05	265	5	5	45	316	11-5	N
06	266	6	6	46	317	11-6	O
07	267	7	7	47	320	11-7	P
10	270	8	8	50	321	11-8	Q
11	271	9	9	51	322	11-9	R
12		8-2		52	273	11-8-2	;
13	275	8-3	=	53	244	11-8-3	$
14	247	8-4	,	54	252	11-8-4	*
15	272	8-5	:	55	333	11-8-5	[
16	241	8-6	!	56		11-8-6	End-of-File
17	276	8-7	>	57	274	11-8-7	<
20	240		Space	60	253	12	+
21	257	0-1	/	61	301	12-1	A
22	323	0-2	S	62	302	12-2	B
23	324	0-3	T	63	303	12-3	C
24	325	0-4	U	64	304	12-4	D
25	326	0-5	V	65	305	12-5	E
26	327	0-6	W	66	306	12-6	F
27	330	0-7	X	67	307	12-7	G
30	331	0-8	Y	70	310	12-8	H
31	332	0-9	Z	71	311	12-9	I
32		0-8-2		72	336	12-8-2	↑
33	254	0-8-3	,	73	256	12-8-3	.
34	250	0-8-4	(	74	251	12-8-4	)
35		0-8-5		75	245	12-8-5	%
36	335	0-8-6	]	76	334	12-8-6	\
37	242	0-8-7	"	77	337	12-8-7	←

*Hollerith column is 6-bit code generated by card reader.

Sensing and Control

Control and sense lines provided for operation of the card reader are as follows. A read-one-card OCP must be given for each card read. The card reader must be operational and not busy before a read-one-card OCP is given.

OCP '0005 Read One Hollerith Card

This OCP will cause the card reader to feed one card and will allow BCD characters to be read into the A-register with an INA 'X005 instruction.

OCP '0105 Read One Binary Card

This OCP will cause the card reader to feed one card and will allow binary characters to be read into the A-register with an INA 'X005.

OCP '0305 Set DMC Mode

Provided when DMC Subchannel Model 116-20-1 is used with the card reader, this OCP will enable the card reader to operate with the DMC. This OCP has to be given each time a card is read since the DMC mode flip-flop will be reset when the card reader has completed reading a card or the DMC range limit is exceeded. Do not attempt to set this flip-flop while the card reader is busy.

OCP '0705 Reset DMC Stop Interrupt

This OCP resets the interrupt which occurs in the DMC mode when the DMC has reached the end of range for the card reader. This OCP is used when the card reader interface includes a DMC Subchannel Model 116-20-2.

SKS '0005 Skip If Card Reader Ready

This instruction will skip if the card reader is ready to send a character to the input bus.

SKS '0105 Skip If Card Reader Not Busy

This instruction will skip if the card reader is not busy. The card reader is busy from the time OCP '0005 or OCP '0105 is received until 84 ms after the 80th column of the card has been read.

SKS '0205 Skip If Not End of File

This instruction will skip if the end-of-file flip-flop is not set. The end-of-file flip-flop is set when an 11-8-6 punch is read or by pushing the EOF button. It is reset every time a card is read.

SKS '0305 Skip If Card Reader Operational

This instruction will skip if the card reader is in an operational state (that is, power on, feed hopper not empty, no card jam, automatic mode, etc.).

SKS '0405 Skip If Card Reader Not Interrupting

This instruction will skip if the card reader has not caused an interrupt. This SKS is used when operating with standard interrupt to determine which device is ready to send or receive new data. This SKS is also used when operating with the DMC option to determine which device has completed a block transfer.

INA '0005 Input from Card Reader If Ready

If the card reader is ready to transfer a character, the next instruction will be skipped and a character (binary or BCD) will be ORed into the A-register. If the card reader is not ready, no transfer will occur, and the program will continue (the next instruction will not be skipped).

INA '1005 Clear A-Register and Input from Card Reader If Ready

This instruction is identical to INA '0005 but the A-register is cleared before data is transferred.

OTA '0020 Set Interrupt Masks

This instruction sets the interrupt mask bit of the card reader if the corresponding bit of the A-register is a ONE and resets the mask bit if the corresponding bit is a ZERO. The instruction is unconditional and never skips. A-register bit 12 controls the card reader interrupt mask.

Operation

Cards are placed face down in the feed hopper with row 9 toward the rear of the reader. With the reader on, the register (REG) pushbutton is pressed and held until card motion stops. The reader is then ready for operation. Control and indicator functions are as follows.

a. READY-indicates that the reader is on and cards are present in the card hopper.

b. READ CARD-indicates a card is being read.

c. ON/OFF -two-position switch for switching a-c power on or off.

d. MAN/AUTO-two-position switch used to manually control the operational status of the card reader. When the switch is in the MAN (manual) position, the card reader appears not operational when status is interrogated with SKS '305. This switch must be in the AUTO (automatic) position when reading cards.

The MAN/AUTO switch may be used anytime that it is desired to stop the card reader while it is reading cards or when restarting after the feed hopper has become empty (providing the card read program properly interrogates the operational status of the card reader before giving a read-one-card OCP). Typically, when the card feed hopper has become empty, the restart procedure is as follows:

(1) Place card reader in manual mode.

(2) Place cards in feed hopper.

(3) Register a card.

(4) Return card reader to automatic mode.

e. REG-positions the first card for reading in automatic or manual mode.

f. EOF-sets the end-of-file flip-flop, which may be queried by SKS '0205.

CARD PUNCH, MODEL 116-64

The standard card punch used with the DDP-116 (Figure 5-4) processes up to 100 cards per minute (nominal), or one card every 600 ms. The card-punching operation is divided into four cycles: feeding, punching, checking and stacking.

a. Cycle I - Feeding. After the POWER ON and START switches have been pressed, a card is fed into the ready station and held there by the die card lever.

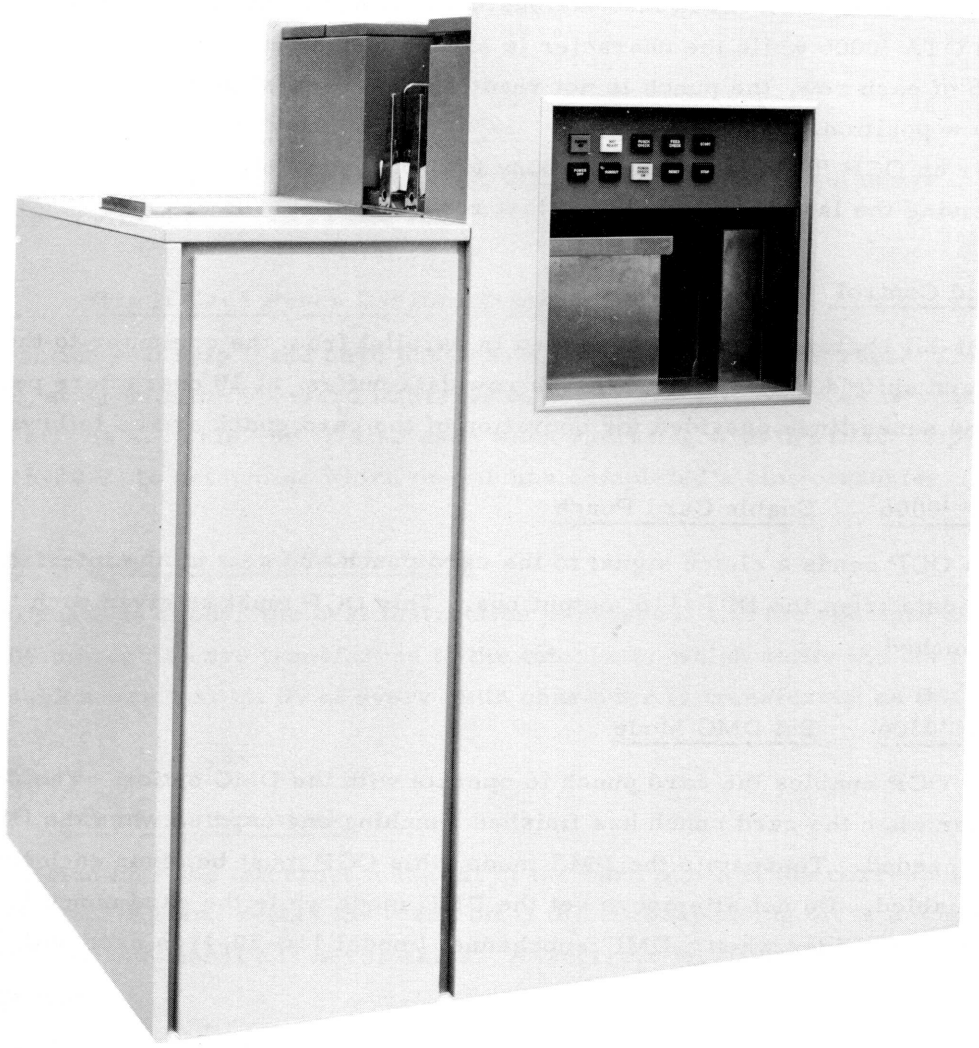
b. Cycle II - Punching. Upon receipt of a feed-and-punch command (OTA '0006) a card movement mechanism passes the card (12th row first, face down) under a row of 80 punch dies, stopping momentarily at each of the 12 card row positions. Holes are punched at column positions determined by the punch code received from the computer. When a card in the ready position starts its movement for punching, a second card is fed into the ready station.

c. Cycle III - Checking. After the card is punched, it is fed through punch check brushes and checked for errors (if the punch check mode is selected). The punch check mode is selected by the PUNCH CHECK ON button, which lights when selected. During the punching cycle, all ONE bit punches cause a 4-bit counter to be stepped. While the card is being read, all detected punches step another 4-bit counter. After row 12 has been read, the two counters are compared; if they disagree, the punch check condition occurs. The punch check condition causes the program to skip on an SKS '0306; no more punch-enable commands (OCP '0006) should be generated until the error is cleared by a runout followed by pressing of the RESET and START pushbuttons.

d. Cycle IV - Stacking. During this operation, cards are removed from the feeding mechanism and stacked into the receiving hopper.

Specifications

Speed	100 cards per minute
Feed	Row by row punching, 12th row first. Corner turning from column by column to row by row done by program.



988

Figure 5-4. Card Punch, Model 116-64

Timing

Row 12 is in punching position and the interface is ready for a character by 90 ms after the punch-enabling OCP '0006. To complete each row, ten 8-bit characters must be provided by as many OTA '0006 instructions. The punch is not ready for 85 to 90 μsec after each OTA '0006 while the character is shifted to the punch buffer. After the tenth OTA '0006 of each row, the punch is not ready for 41.5 ms while the card is advanced to the next row position.

After an OCP '0006, the punch is busy for approximately 600 ms or until 28.9 ms after receiving the last character of the last row (row 9).

Sensing and Control

Eight-bit characters are transferred in parallel from the computer to the punch interface and shifted serially to the punch row data buffer, at 10 characters per row. Control and sense lines provided for operation of the card punch are as follows.

OCP '0006 Enable Card Punch

This OCP sends a clutch signal to the card punch and sets up the interface controls to receive data from the DDP-116 output bus. This OCP must be given each time a card is to be punched.

OCP '0306 Set DMC Mode

This OCP enables the card punch to operate with the DMC option. The DMC mode is reset either when the card punch has finished punching one card or when the DMC range limit is exceeded. To sustain the DMC mode, this OCP must be given each time the card punch is enabled. Do not attempt to set the DMC mode while the card punch is busy. This instruction is provided when a DMC subchannel (model 116-20-1) is used with the card punch.

OCP '0706 Reset DMC Stop Interrupt

This OCP must be given to reset an interrupt which occurs when a DMC stop interrupt option (model 116-20-2) is used with the card punch and the DMC has reached the end of range.

SKS '0006 Skip If Card Punch Ready

This instruction will skip if the card punch is ready to receive an 8-bit character from the output bus.

SKS '0106 Skip If Card Punch Not Busy

The card punch is busy from the receipt of the OCP '0006 until 3.78 ms after receiving the tenth 8-bit character of the last row (row 9).

SKS '0206 Skip If Card Row Not Positioned

This instruction will skip if the card punch is not ready to punch a new row. The row ready condition is true from the time the card punch has positioned the card to punch a row until the 80th information bit of that row has been received. There are 12 row ready signals per card cycle.

SKS '0306 Skip If Card Punch Operational

This instruction will skip if none of the NOT READY conditions listed under card punch operating notes is in effect.

SKS '0406 Skip If Card Punch Not Interrupting

This instruction will skip if the card punch is not requesting an interrupt. This SKS is used when operating with the standard interrupt to determine which device is ready to send or receive new data. This SKS is also used when operating with the DMC stop-interrupt option (model 116-20-2) to determine which device has completed a block transfer.

OTA '0006 Output to Card Punch Ready

If the card punch is ready, the next instruction is skipped, and the contents of DDP-116 output bus bits 09 through 16 are transferred to the interface, which shifts the character to the punch in such a way that bit 09 of every tenth character is transferred as the first bit of a row.

OTA '0020 Set Interrupt Mask

This instruction sets the interrupt mask bit of the card punch if the corresponding bit of the A-register is a ONE and resets the mask bit if the corresponding bit is a ZERO. This instruction is unconditional and never skips. A-register bit 13 controls the card reader interrupt mask.

Operating Notes

The functions of the card punch controls and indicators are summarized below.

a. POWER ON - applies primary power to punch
b. NOT READY (not operational) - indicates that one of the following conditions is present:
    (1) Hopper empty
    (2) Stacker full
    (3) Chad box not in place
    (4) A cover not in place
    (5) Punch die improperly installed
    (6) Start switch not pressed
    (7) Stop switch pressed
    (8) Carry-punch switch on

c. PUNCH CHECK - indicates that the code sent was not punched when in the punch check on mode, or that less than 80 information clocks were received.

d. FEED CHECK - indicates a card jam or improper feed.

e. START - places punch in operating mode, ready for computer punch instructions.

f. POWER OFF - removes primary power.

g. RUN OUT - causes punch to clear any cards in any of the four operating cycles when in not ready condition.

h. PUNCH CHECK ON - causes punch to check that codes received are codes punched.

i. RESET - resets error circuits.

j. STOP - puts punch in not operational mode.

MAGNETIC TAPE SYSTEMS, MODELS 116-40,-41 and -42

Three types of magnetic tape transports are available with the DDP-116. The low-speed unit, model 116-40, shown in Figure 5-5, operates at a tape speed of 45 ips and at densities of 200 or 556 bits per inch. The high-speed units, models 116-41 and 116-42, operate at a tape speed of 75 ips. (See Figure 5-6. Only one transport is shown because both models are identical in appearance.) Model 116-41 is a dual-density unit (200 or 556 bits per inch) and model 116-42 is a triple-density unit (200, 556 or 800 bits per inch). Control of the magnetic tape transports (MTT) and reading from or writing on tape are operations performed by the tape control unit (TCU).

Modes of Operation

There are two binary modes of operation: 3 characters/word and 2 characters/word. In the 3 characters/word binary mode, 16 bits are transmitted to the buffer from the computer and are written on magnetic tape as three 6-bit characters. The third character will contain only 4 information bits and 2 zero bits. This is represented by bits 13-16 of the word. During input, three characters are read into the buffer (two 6-bit characters and one 4-bit character), forming 16 information bits which are transmitted to the computer via the input bus. The 4-bit character is made up of tracks 1 through 4.

There is one BCD mode of operation: 2 characters/word. This mode is identical to the 2 character/word binary mode except recording is in even parity and the zero character is converted to (or from) 12_8 to avoid writing a false gap on tape.

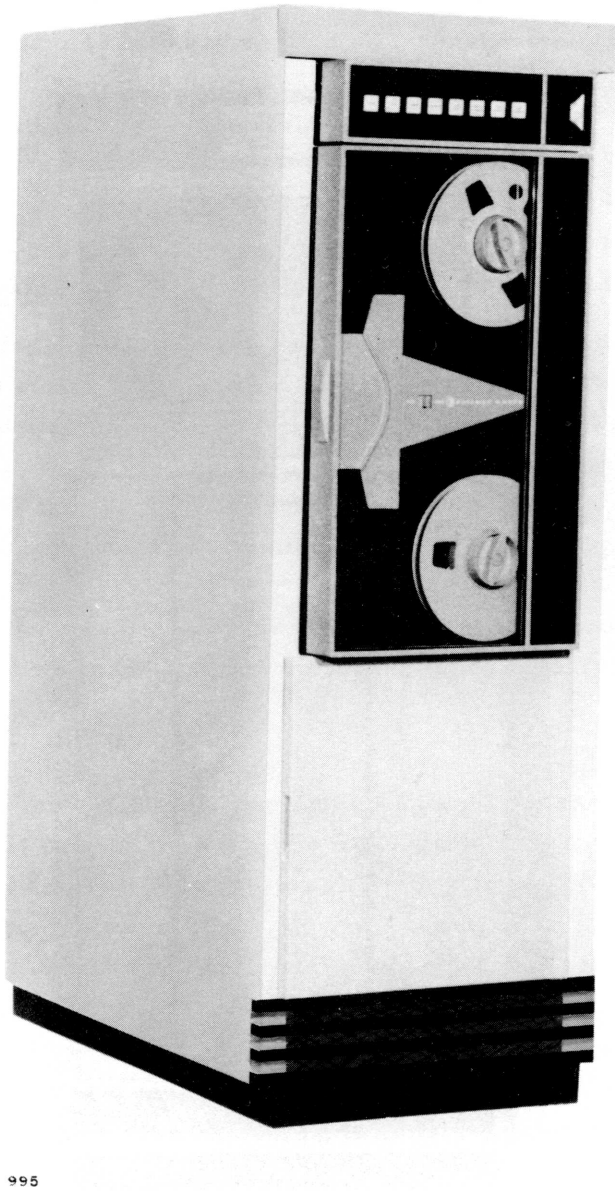


Figure 5-5. Magnetic Tape Transport, Model 116-40

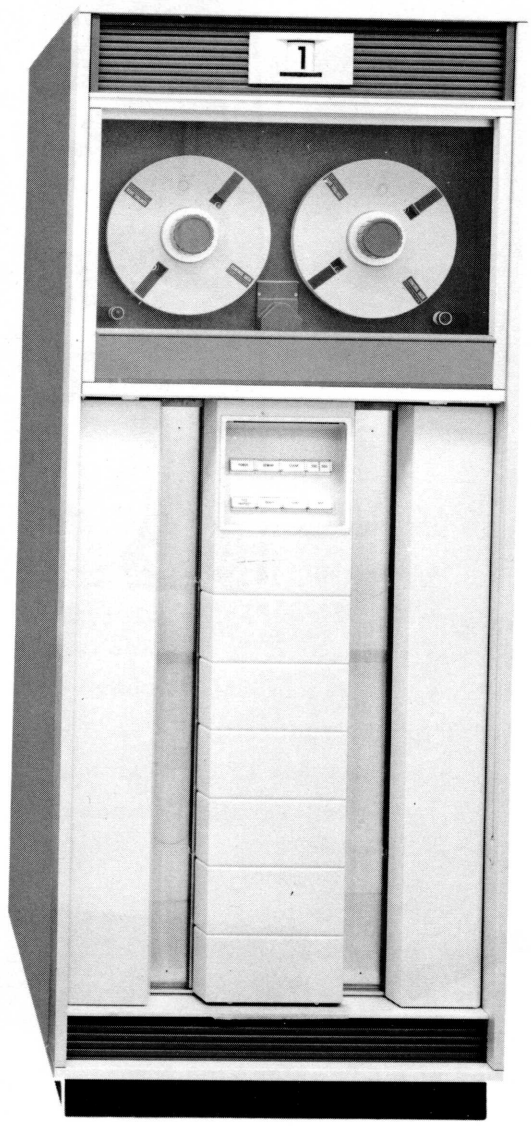


Figure 5-6. Magnetic Tape Transport, Models 116-41 and 116-42

Utilizing the 2 characters/word mode on output, a 16-bit word is transmitted to the buffer, but only the leftmost 12 bits are written on magnetic tape (as two characters). Similarly, when inputting from magnetic tape, two characters of information are read into the buffer and transmitted to the computer via the input bus.

Special Character Conversion

The tape control logic performs the 0 to 12 and 12 to 0 conversion when transferring in BCD mode. (This is consistent with IBM BCD code format.) When a BCD $(00)_8$ is loaded into the word-forming buffer from the output bus, the tape control logic converts it, so as to write a $(12)_8$ on tape. When reading a BCD $(12)_8$ from tape, the tape control logic re-converts it and transfers the character to the input bus as a $(00)_8$.

It must be pointed out that this invalidates the use of $(12)_8$ being used by the computer in BCD mode.

Since the internal working code of the DDP-116 is ASCII, programmed code conversion is necessary to read and write IBM code compatible tapes (see Table 5-2).

Timing

In the low-density mode (200 bits per inch) it takes 111 μsec for a Model 116-40 Magnetic Tape Transport to read or write one 6-bit character on tape. Since the computer can operate at transfer speeds well above this level, transfer operations of the computer can be calculated from this rate. For instance, if an MTT is in the 2 characters/word mode, the average time between input/output data transfer instructions would be 2 x 111 μsec, or 222 μsec. Under the same conditions in the 3 characters/word mode, the average time is 3 x 111 μsec, or 333 μsec. These speeds would also be true when operating in the optional DMC mode. When operating in the DMC mode, 6.8 μsec are required to service the buffer. The remaining time is available for concurrent computation and additional input/output. In the high-density mode of operation (556 bits per inch), the average time between I/O data transfers is 80 μsec for 2-character words and 120 μsec for 3-character words.

When using a 75 ips MTT(models 116-41/42), the character rates are one character every 66.6 μsec, 24 μsec and 16.7 μsec for 200, 556 and 800 bpi, respectively. Average times between computer operations can be determined as described in the preceding paragraph.

A temporary storage buffer is part of the TCU to prevent loss of a data character when long instructions prevent transfer of high-speed tape data words.

Table 5-2.
ASCII and Magnetic Tape Codes

Character	ASCII Code	Magnetic Tape Code	Character	ASCII Code	Magnetic Tape Code
0	260	12*	V	326	25
1	261	01	W	327	26
2	262	02	X	330	27
3	263	03	Y	331	30
4	264	04	Z	332	31
5	265	05	Space	240	20
6	266	06	!	241	16
7	267	07	"	242	37
8	270	10	$	244	53
9	271	11	%	245	75
A	301	61	'	247	14
B	302	62	(	250	34
C	303	63	)	251	74
D	304	64	*	252	54
E	305	65	+	253	60
F	306	66	,	254	33
G	307	67	-	255	40
H	310	70	.	256	73
I	311	71	/	257	21
J	312	41	:	272	15
K	313	42	;	273	52
L	314	43	<	274	57
M	315	44	=	275	13
N	316	45	>	276	17
O	317	46	?	277	
P	320	47	[	333	55
Q	321	50	\	334	76
R	322	51	]	335	36
S	323	22	↑	336	72
T	324	23	←	337	77
U	325	24			

* When writing magnetic tapes in even parity (BCD) mode, 00_8 is written as 12_8; conversely, when reading in even parity, 12_8 is read as 00_8.

Specifications of Model 116-40 MTT

Tape Speed

Forward and reverse - 45 ips
Start time - 5 ms bi-directional
Stop time - 1.5 ms bi-directional
Rewind - less than 3 minutes for a full reel of tape

Modes

Remote: Control by TCU
Manual: Under the following pushbutton control
- Rewind
- Reverse
- Load Point - tape moves forward until reflective photo-sensor on tape passes by the read and write heads
- Hi Density
- Lo Density
- Unit Number Select

Expansion

Up to 4 transports on a single TCU

Specifications of Models 116-41/42 MTT

Tape Speed

Forward and reverse - 75 ips
Start time - 2.75 ± 0.5 ms
Stop time - 2.25 ± 0.5 ms
Rewind - over 350 in./sec

Modes

Remote: Controlled by computer signals as with 116-40 MTT
Manual: Under following pushbutton control

Power	200 bpi
Ready	556 bpi
Load	800 bpi (on 116-42 only)
Rewind	Clear Unit Number Selector

Expansion

Up to 4 transports on a single TCU

Tape Format

Figure 5-7 shows how data is organized into records and files on the tape. Six data tracks and one parity track are used with the spacing as shown.

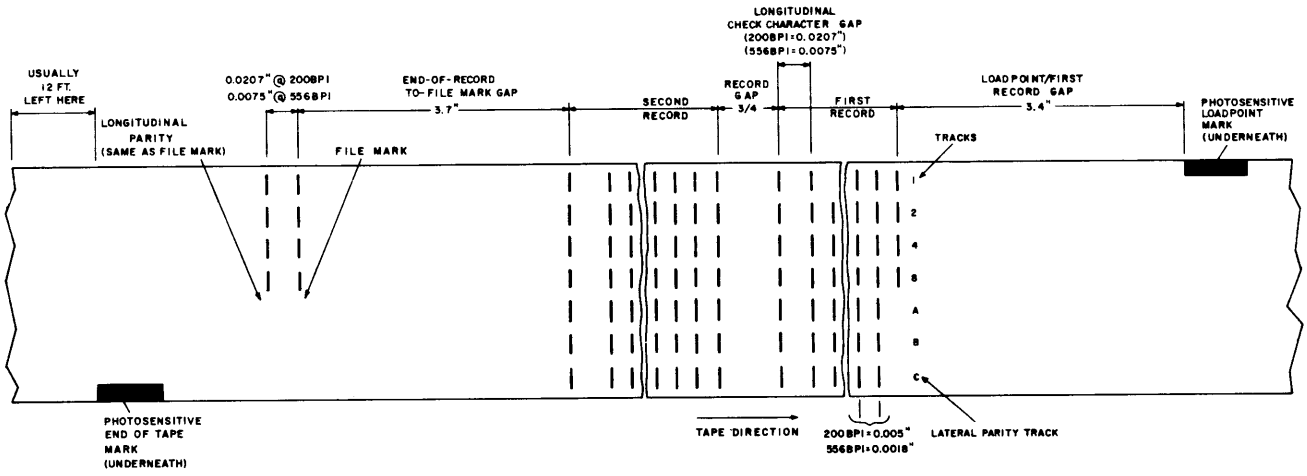


Figure 5-7. Magnetic Tape Format

MTT Parity Checking

Two types of parity checking are used: longitudinal and lateral. Parity bit assignments are as follows.

a. Longitudinal Parity (Binary or BCD)

ONE - Odd number of bits in that channel of a record
ZERO- Even number of bits in that channel of a record

b. Lateral Parity (Binary)

ONE - Even number of ONEs in a character
ZERO-Odd number of ONEs in a character

c. Lateral Parity (BCD)

ONE - Odd number of ONEs in a character
ZERO-Even number of ONEs in a character

Sensing and Control

The following instructions are used to control the TCU and to transfer data to and from the magnetic tape transports. Note that the last octal digit (X) of all MTT control instructions is assigned according to the TCU configuration (see the following table).

TCU	MTT			
	1	2	3	4
Single No. 1	0			
Single No. 2	1			
Single No. 3	2			
Single No. 4	3			
Dual No. 1	4	5		
Dual No. 2	6	7		
Triple	4	5	6	
Quad	4	5	6	7

Standard TCU configurations are as follows:

a. Up to four singles
b. Four single and one or two duals
c. Four single and one triple or one quad

OCP '001X Read BCD

This instruction causes the selected tape drive to move forward for one record during which time data is transferred to the word-forming buffer of the TCU, with every two 6-bit characters forming a word. Longitudinal parity is checked and lateral parity checked in a BCD mode (even). The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

NOTE

When the end-of-tape marker is reached it will in no way affect the operation of the instruction being performed. A sense line will be set which can be tested by an SKS '051X.

OCP '011X Read Binary

This instruction is identical to the read BCD instruction except that lateral parity checking is done in the binary mode (odd).

OCP '021X Read Binary 3 Char./Word

This instruction causes the selected tape drive to move forward for one record length. Data is read from tape and formed into 3-character words. Each word is comprised of two 6-bit characters and one 4-bit character (every third character off the tape is the 4-bit character). Longitudinal parity and lateral parity are checked in a binary mode (odd). The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '031X Set DMC Mode

This instruction is used when the DMC option (model 116-20) is connected to the TCU. It sets up the TCU for a DMC transfer. As soon as the TCU is ready (a full word has been formed), the first word will be transferred. This instruction must be followed by a write OCP instruction. When doing an input transfer, an OCP read instruction must precede the DMC enable OCP. The TCU will remain in this mode until either the end of range or an end of record is reached.

OCP '041X Write BCD

This instruction causes the selected tape drive to move forward and, after the proper inter-record gap, to begin writing data on tape. Data words transferred to the TCU are written on tape as two 6-bit characters. Even lateral parity is generated; read-after-write lateral parity and longitudinal parity are checked. The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '051X Write Binary

This instruction is identical to write BCD except that lateral parity generation and read-after-write parity checking are done in the binary mode (odd).

OCP '061X Write End of File

This instruction causes the selected tape drive to write a file mark (octal 17) with the required gap distance in front of it. Not output command is necessary. The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '071X Reset DMC End-of-Transmission Interrupt on TCU

This instruction is used to reset the end-of-transmission interrupt signal associated with the DMC interrupt option (model 116-20-2) when it is connected to the TCU.

OCP '101X Write Binary 3 Char./Word

This instruction causes the selected tape drive to move forward and, after the proper inter-record gap, to begin writing data on tape. Data words transferred to the TCU are written on tape as three 6-bit characters with every third character having only 4 bits of meaningful information. Odd lateral parity is generated and read-after-write lateral and longitudinal parity are checked. The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '111X Space Forward One Record

This instruction causes the selected tape drive to move forward for one record length. No data transfer takes place. The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '121X Space Forward One File

This instruction causes the selected tape drive to move forward to the next file mark. When the file mark is detected, the drive stops, positioning the tape between the file mark and the next record. No data transfer takes place. The instruction must not be given while the TCU is busy. Therefore, an SKS "not busy" test should precede the instruction.

OCP '141X Rewind

This instruction causes the selected tape drive to rewind and continue rewinding until the load point is reached. The programmer should SKS the selected tape's rewind busy status to determine when the tape has concluded rewinding. Simultaneous rewinding can be accomplished on multiple unit systems by interleaving SKS "not busy" tests with the OCP rewind instructions. For the model 116-40 single TCU, the rewind instruction causes the TCU to remain busy for 50 ms. Model 116-40 multiple unit TCUs are busy for 20 μsec and model 116-41 TCUs remain busy for 10 μsec.

OCP '151X Backspace One Record

This instruction is functionally identical to space forward one record except that the selected tape drive moves in a reverse direction. If the load point is encountered before a record, tape motion stops. If the instruction is given when the head lies between the first and second record, the instruction will cause the tape to stop between load point and the first record.

OCP '161X Backspace One File

This instruction is functionally identical to space forward one file except that the selected tape drive moves in a reverse direction. If a load point is encountered before a file mark, tape motion stops. An SKS load point test at this time will indicate the tape is positioned at the load point.

SKS '001X Skip If the TCU Is Ready

This instruction tests whether the TCU is ready

(a) to accept another word from the computer in the output mode or

(b) to present another word to the computer in the input mode.

SKS '011X Skip If the TCU Is Not Busy

The TCU normally remains busy as long as the selected tape drive is in motion (except for a rewind instruction). When a stop signal is initiated, a 5-ms delay is set, after which the TCU busy is reset. This holds for all model 116-40 systems. However, the model 116-41 is equipped with "through-put" logic which allows the selected transport to execute the next OCP command without stopping if the direction of motion is not changed by the OCP instructions. The stop signal then resets TCU busy eliminating the 5-ms delay.

SKS '021X Skip If the TCU Has Not Detected a Parity Error

Lateral and longitudinal parity are both checked while writing or reading. This instruction senses the flip-flop that is set whenever an error does occur. The error is stored until the beginning of the next OCP instruction. While writing, this instruction should only test for error after the TCU has become not busy. While reading, the test can be made after each transfer. Following a parity error, the parity indicator will remain illuminated until the next movement of the tape transport.

SKS '031X Skip If the TCU Has Not Detected Beginning of Tape

This instruction tests the load point status of the selected tape drive. The load point status will be reset upon execution of the next TCU OCP that involves forward motion.

SKS '041X Skip If the TCU Is Not Interrupting

This instruction tests whether the TCU is enabling a TCU interrupt on the standard interrupt line. (The TCU must be ready and its interrupt mask must be set.)

SKS '051X Skip If the TCU Has Not Detected End of Tape

This instruction tests the end-of-tape status of the selected tape drive. When set, the end-of-tape status can only be reset by executing a rewind command (on model 116-40 systems) or by executing either a rewind or a reverse motion command (on model 116-41 systems.)

SKS '061X Skip If the TCU Has Not Detected End of File

This instruction tests for an end of file on the selected tape drive. The file mark detection is stored until the next motion command.

SKS '141X Skip If the MTT Is Not Rewinding

This instruction tests whether the selected transport is still rewinding. Each transport on a multiple tape system can be individually sensed for its rewind status.

INA '001X Input from the TCU*

This instruction transmits the 16 bits from the WFB to the A-register if the TCU is ready.

INA '101X Clear Register A and Input from the TCU*

Identical to INA '001X except that the A-register is cleared before transmission from the WFB.

OTA '001X Output to the TCU*

This instruction transmits a 16-bit word from the A-register to the WFB if the TCU is ready.

Program Restriction

Condition. -- Operating the TCU at its maximum rate of transfer (800 bpi at 75 ips) through a DMC subchannel on a system equipped with a multiply/divide option.

Problem. -- When the TCU causes a DMC interrupt request (DIL), the demand must be accepted within a finite period of time or a data character will be lost. It is possible for a DIL to be initiated 1.34 μsec prior to the initiation of an instruction and not be accepted until the conclusion of the instruction. The longest instruction is divide, thus bringing the total elapsed time between initiation and acceptance of a DIL to 1.34 +16.7 (divide) = 18.04 μsec. If the instruction prior to the divide is a shift instruction, an additional 0.34 μsec delay can occur for each bit shifted or a total of 10.9 for a 32-bit shift. Thus, for this sequence of instructions, the total elapsed time could be a maximum of 12.24 + 16.7 = 28.94 μsec.

When the computer releases control to the DMC, an additional 3.85 μsec is required to process the data transfer to or from the TCU. This brings the total elapsed time between DIL request and acceptance to 18.04 + 3.85 = 21.89 μsec for divides without shifts immediately preceding them and 28.94 + 3.85 = 32.79 μsec for divides with shifts immediately preceding them.

The maximum allowable time between TCU transfers before a data character is lost is as follows:

	800 bpi	556 bpi
Writing 2 char./word mode	25.4 μsec	40.0 μsec
Writing 3 char./word mode	28.4 μsec	43.0 μsec
Reading 2 char./word mode	24.0 μsec	38.6 μsec
Reading 3 char./word mode	27.0 μsec	41.6 μsec

*When reading in the 2 characters/word mode, an INA will transfer bits 13/16 as ZEROs to the A-register. When writing in the 2 characters/word mode, bits 13/16 of the A-register will be ignored by an OTA.

Conclusion. --

a. Since the total time for a shift with the maximum number of shifts immediately preceding a divide exceeds all of the maximum TCU transfer rate delays at 800 bpi, it is clear that this sequence would result in the loss of a data character. It would be possible to reliably program around this by adding a dummy instruction between the shift and divide (such as an NOP).

b. It is recommended that the maximum allowable TCU transfer times be taken into consideration for other possible restrictive combinations.

The preceding examples of shift and divide instructions did not include any indirect addressing or indexing. Indirect addressing can continue indefinitely if called for with each indirect address cycle requiring an additional 1.7 μsec. When indexing is used, an additional 2.04 μsec are required for instruction execution.

It is therefore necessary that the table of maximum allowable delay times before transfer be carefully analyzed for both 800 bpi and 556 bpi when generating programs using the TCU in all modes of operation.

Transport Operational Precautions

a. Do not rewind an alignment tape or stop an alignment tape in middle of reel.

b. When loading tape do not apply excessive tension on tape around rollers.

c. Use Freon TF for cleaning tape path.

d. Tape should not be allowed to remain in the columns for an extended period of time while the transport is powered but in an unloaded condition.

LINE PRINTER, MODEL 116-67

The standard DDP-116 line printer (Figure 5-8) accepts coded signals from the computer and produces printed copy at a rate of 300 lines per minute. The printer can approach a speed of 360 lines per minute printing numeric characters only. The line printer is normally connected to the computer through the I/O bus and includes its own 16-bit buffer with an integral 120-character delay line storage. The line printer can print 62 out of a total of 64 characters. The two extra characters are space or blank codes (octal 00 and octal 40). This character code set is ASCII compatible (see Table 5-1). The printer accepts standard forms up to 19 inches wide. Conventional 1-inch tabulating ribbon is used.

Line Printer Operation

The line printer operation is comprised of paper advance, loading the serial memory buffer, and printing.

Figure 5-8. Line Printer, Model 116-67

Paper Advance. -- The paper advance logic in the line printer receives one of eight types of paper advance OCPs from the computer for each movement of paper. Paper advance in the line printer is controlled by a preprogrammed paper-tape loop on the paper tape reader in the line printer. Track No. 1 of the form control paper tape loop is the top of the form track, and Track No. 8 is the single space track. The other tracks may be punched as desired by the programmer to control the paper advance throughout the program. If no tape is placed in the paper tape reader any of the eight OCPs will activate a single line advance.

Regardless of whether the line printer program is using the standard I/O, DMC or standard interrupt, all loading may be done during a paper advance. The buffer must be loaded during this time to maintain the maximum speed of the line printer (31 ms are required to complete an advance of a single line). If the programmer wishes to stack paper advances (that is, an advance to one track followed by an advance to another track with no print cycles between), there must be a paper-advancing SKS (SKS '0300) before another advance OCP is given. This is the only time that this SKS is required.

Loading. -- There are two modes of buffer loading: word mode (two 6-bit characters per OTA '0000) and character mode (one 6-bit character per OTA '0000). In the word mode, A-register bits 3-8 form the first character to be transferred, and A-register bits 11 through 16 form the second character. In the character mode, A-register bits 11 through 16 form the character to be transferred. The first character sent to the line printer buffer is the character to be printed in the first print hammer position (that is, the first print column on the left-hand side of the page). All successive characters will print in sequential locations. If less than 120 characters are sent, all print columns to the right of the last characters sent will be blank.

The mode is selected by OCP '0000 (character mode) or OCP '0100 (word mode), and characters are loaded with an OTA '0000. Loading of the buffer is accomplished during the paper advance time. After the last character to be printed is loaded, the program must give a print OCP (OCP '0200) within 27 μsec. Otherwise, the character ready flip-flop will be set, causing an interrupt if that mode is selected. If this ready flip-flop setting is ignored by the program, the print OCP (when it occurs) will clear the ready flip-flop before entering the print mode. The line printer logic will wait until paper advance is completed before entering the print mode. Characters are loaded into a serial delay line memory at a rate of 28 μsec per character. If the program does not load a character within 27 μsec after a demand (the ready flip-flop being set), the loading cannot be accomplished until the delay line makes a complete rotation, of about 896 μsec when that character position is reached in the line again.

Printing. -- Printing is done by means of a rotating character drum and a bank of linear motion print hammers. The character drum contains 64 characters, including the capital letters of the alphabet, number 0 through 9, and all of the standard symbols commonly used on business and mathematical forms. Each symbol is repeated 120 times lengthwise

across the cylindrical character drum. After the printer buffer is loaded and either a print command (OCP '0200) or an automatic print command is received, the line printer logic will enter the print mode. One full revolution of the character drum is required to print a full line. As each line of characters passes the print hammers, a comparison is made between the code of the character presently in front of the hammers and the characters contained in the delay line storage. Each comparison generates a pulse which sets one of the 120 flip-flops controlling the print hammers. The hammers corresponding to the data in the delay line impact the paper against a printing ribbon and onto the appropriate character on the drum thus printing all like characters at one time. When the print cycle is completed, a flip-flop will be set if the DMC stop interrupt option (model 116-20-2) is provided which will cause an interrupt to the computer. During the time that the printing mode is enabled, the line printer will be busy (SKS '0100 will not skip), and this busy flip-flop will be reset upon exiting from the print mode, signaling the program that another paper advance, load buffer subroutine can be executed.

The print mode can be entered in three different ways:

a. With the standard model 116-67, an OCP '0200 enables the logic to enter the print mode.

b. When 120 characters have been loaded into the buffer, the print mode is entered automatically.

c. If a DMC subchannel option (model 116-20-1) is provided, the print command is automatic when the DMC channel reaches the end of range.

An OCP print does not conflict with an automatic print command. The actual printing will not occur until the present paper advance is completed. (This is taken care of by internal logic and an SKS on paper advancing before a print command is unnecessary and time consuming).

Specifications

Speed	300 lines per minute
Print Positions	120 print positions per line
Ribbon	Conventional 1-inch horizontal tabulating ribbon
Forms	Accepts standard pin-feed forms up to 19 inches wide with 1/2 inch hole centers
Print Area	12 in. wide
Characters per inch	10
Paper Line Advance Time	30 ms
Paper Shift Feed	20 in. per second
Line-to-Line Spacing	0.167 (±0.015) in., 6 lines per in.
Paper Capability	Up to 6 parts, 12-lb bond with single-shot carbon, or on a tabulating card (0.007 in.) plus a second record sheet
Character Type	Open Gothic
Horizontal Character Spacing	0.100 (±0.010) in. between centers

Sensing and Control

Control and sense lines provided for operation of the line printer are as follows.

OCP '0000 Enable Character Mode

This OCP allows characters to be loaded into the buffer in the character mode. A-register bits 11-16 are loaded into the next sequential location of the delay line.

OCP '0100 Enable Word Mode

This OCP allows characters to be loaded into the buffer in the word mode. A-register bits 3-8 (first character) and A-register bits 11-16 (second character) are loaded into the next two sequential locations of the delay line.

OCP '0200 Print One Line

This OCP sends the printer logic into the print mode and must be given after the last character in the buffer is loaded into the delay line. Paper advance is finished before printing takes place.

OCP '0300 Set DMC Mode

This OCP sets the DMC mode flip-flop, enabling DMC operation with the standard DMC subchannel (model 116-20-1). This flip-flop is reset at the end of every print cycle; therefore it must be set each time DMC operation is desired. This OCP should be given before either word or character mode is selected.

OCP '0700 Reset Stop Interrupt

This OCP, used only with the DMC stop interrupt option (model 116-20-2), resets the DMC stop interrupt flip-flop, terminating the interrupt which occurs after the print cycle is completed.

OCP '1X00 Advance Paper

This OCP will cause paper to be advanced until a hole is detected in channel X+1 in the form control tape. Channel 1 (X = 0) is reserved for the top of the form, and channel 8 (X = 7) is usually reserved for single space. The remaining channels may be used as desired. The printer must not be busy and paper must not be advancing when a paper advance OCP is given.

SKS '0000 Skip If Ready to Receive a Character

This instruction will skip if the line printer is ready to receive a 6-bit character in character mode or two 6-bit characters in word mode.

SKS '0100 Skip If Line Printer Not Busy

This instruction will skip if the line printer is not busy. The line printer is busy from the time the last character has been loaded into storage and a print command is received until the contents of the serial memory is printed.

SKS '0200 Skip If No Parity Error

This instruction will skip if the parity error flip-flop is not set. Parity is checked during the print mode (while the printer is busy). While loading, even parity is generated. Therefore, this is a parity check on the data loaded into the delay line. This parity flip-flop is reset upon starting a new loading cycle (OCP '0000 or OCP '0100).

SKS '0300 Skip If Paper Not Advancing

This instruction, which will skip if the line printer is not advancing paper, will not skip from the time an OCP '1X00 is received until the paper advance is completed (about 31 ms).

SKS '0400 Skip If Line Printer Not Interrupting

This instruction will skip if the line printer is not requesting an interrupt. When operating with standard interrupt, this SKS determines which device is ready to send or receive new data. This SKS is also used when operating with the DMC stop interrupt option (model 116-20-2) to determine which device has completed a block transfer.

OTA '0000 Output to Line Printer

If the ready-to-transfer-a-character flip-flop is set, the next instruction will be skipped, and the contents of the output bus will be sent to the line printer buffer. If the character mode is selected, bits 11 through 16 will form the character to be transferred. If the word mode is selected, bits 03 through 08 will be the first character transferred, and bits 11 through 16 will be the second character transferred.

OTA '0020 Set Mask

If bit 14 of the A-register is set, the line printer interrupt mode will be masked on, allowing an interrupt to occur if the ready flip-flop is set (when operating in the normal mode) or when the DMC end-of-block-transfer flip-flop is set (when operating with a DMC stop interrupt option)(model 116-20-2).

Timing

The first character ready indication occurs about 3 μsec after the OCP '0000 or OCP '0100. The next character ready occurs from 28 to 896 μsec after the first OTA '0000 is executed. All other ready's will occur at 28-μsec intervals (in the character mode) or 56-μsec intervals (in the word mode). If a character transfer does not occur within 26 μsec after the ready indication, the next ready will occur 896-μsec after the OTA '0000. The line-printer-busy flip-flop will be set from the time the 120th character has been loaded into the delay line and/or a print command is executed until all characters in the delay line are printed. This operation takes a minimum of 3 ms and a mximum of 167 ms. Paper advance time is 31 ms, and all loading of the buffer should be accomplished during this time.

Operating Notes

Line printer control and indicator functions are as follows.

a. POWER ON - pressing this pushbutton alternately applies and removes power to the printer. When power is on, the pushbutton is illuminated.

b. STANDBY - when on, indicates that line printer is not ready to print. This light comes on when the line printer is out of paper, the hammer inhibit switch is on, or the standby condition is selected by depressing the button.

c. SKIP FEED - pressing this pushbutton will slew the paper; after releasing, the paper will stop at the top of the form.

d. TOP OF FORM - pressing this pushbutton will allow the paper to advance the punch in Track No. 1.

SECTION VI
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  PZE   **                RETURN ADDRESS
      STA   COMM              SAVE HIGH ORDER A
      LDA*  DADD              ENTER ADDRESSES OF YH AND YL
      STA   COMM+1            ADDRESS OF YH
      ADD   =1                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  PZE   **
      STA   COMM                 SAVE A
      LDA*  DSUB                 ENTER ADDRESS OF YH AND YL
      STA   COMM+1
      ADD   =1
      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

*     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
```

Fixed Point, Single Precision Multiply Subroutine (Cont)

```
      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 DIV 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
```

Fixed Point, Single Precision Divide Subroutine (Cont)

```
*     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
```

Paper Tape Read Subroutine

```
* THIS SUBROUTINE READS DATA FROM THE HIGH SPEED PAPER TAPE READER.
* DATA ARE READ TWO CHARACTERS PER ENTRY.  THE TWO CHARACTERS ARE PACKED
* INTO A WORD WITH THE FIRST CHARACTER READ IN THE LEFT HALF AND THE
* SECOND CHARACTER READ IN THE RIGHT HALF.  THE PACKED WORD IS LEFT IN
* THE A REGISTER ON RETURN TO THE CALLING PROGRAM.
*
      REL
      SUBR  PTR                ESTABLISH SUBROUTINE NAME.
*
PTR   DAC   **                 SUBROUTINE ENTRY POINT.
      OCP   1                  START TAPE READER.
      INA   '1001              CLEAR A AND INPUT FIRST CHARACTER
      JMP   *-1                (DELAY IF DATA NOT READY).
      LGL   8                  POSITION FIRST CHARACTER IN LEFT
*                              HALF OF A.
      INA   1                  INPUT SECOND CHARACTER.  NOTE THAT
*                              A IS NOT CLEARED AND THAT THE
*                              SECOND CHARACTER IS "ORED" INTO A
      JMP   *-1                (DELAY IF DATA NOT READY).
      OCP   '101               STOP TAPE READER (NOTE THAT TAPE
*                              READER STOPS IN TIME TO PREVENT
*                              LOSING THE FOLLOWING CHARACTER).
      JMP*  PTR                RETURN TO CALLING PROGRAM.
      END
```

Output on High-Speed Paper Tape Punch

```
* THIS SUBROUTINE PERFORMS THREE DIFFERENT FUNCTIONS DEPENDING ON WHICH
* ENTRY POINT IS USED.  THE "POWER ON" ENTRY IS USED TO TURN ON PUNCH
* POWER (THIS IS MADE A SEPARATE FUNCTION BECAUSE THE PUNCH WILL NOT BE
* READY TO RECEIVE DATA UNTIL ABOUT THREE SECONDS AFTER POWER IS TURNED
* ON AND AS A SEPARATE FUNCTION, POWER MAY BE TURNED ON PRIOR TO THE
* PROGRAM BEING READY TO PUNCH DATA).  THE "POWER OFF" ENTRY IS USED TO
* TURN PUNCH POWER OFF AFTER THE LAST DATA BLOCK HAS BEEN PUNCHED.  THE
* "PUNCH DATA" ENTRY IS USED TO PUNCH A "CARD IMAGE" (80 CHARACTERS)
* FROM A PACKED DATA FIELD BEGINNING AT LOCATION '100 AND EXTENDING TO
* LOCATION '147 (40 WORDS).
*
      REL
      SUBR  PON              ESTABLISH SUBROUTINE NAME FOR "POWER ON"
      SUBR  POFF             ESTABLISH SUBROUTINE NAME FOR "POWER OFF"
      SUBR  PNCH             ESTABLISH SUBROUTINE NAME FOR "PUNCH DATA"
*
PON   DAC   **               "PUNCH POWER ON" ENTRY.
      SKS   '102             TEST FOR POWER ALREADY ON.
      OCP   2                IF NOT ON, TURN ON.
      JMP*  PON              RETURN TO CALLING PROGRAM.
*
POFF  DAC   **               "PUNCH POWER OFF" ENTRY.
      SKS   2                WAIT FOR PUNCH READY TO ACCEPT
      JMP   *-1              DATA TO INSURE AGAINST TURNING
*                            PUNCH OFF WHILE IN PUNCH CYCLE.
      OCP   '102             TURN PUNCH POWER OFF.
      JMP*  POFF             RETURN TO CALLING PROGRAM.
*
PNCH  DAC   **               "PUNCH DATA ENTRY.
      LDA   0                SAVE CURRENT INDEX VALUE.
      STA   X
      LDA   =-40             SET INDEX TO -40 FOR 40 WORD
      STA   0                BLOCK (80 CHARACTERS).
      SKS   '102             "FAIL-SAFE" POWER ON TEST -
      OCT   2                IF POWER OFF, TURN ON.
LOOP  LDA   '100+40,1        PICK UP PACKED DATA WORD.
      ARR   8                POSITION LEFT CHARACTER FOR PUNCHING.
      OTA   2                OUTPUT CHARACTER TO PUNCH
      JMP   *-1              (DELAY IF PUNCH NOT READY).
      ARR   8                REPOSITION RIGHT CHARACTER FOR PUNCHING.
      OTA   2                OUTPUT CHARACTER TO PUNCH
      JMP   *-1              (DELAY IF PUNCH NOT READY).
      IRS   0                STEP INDEX.
      JMP   LOOP             IF INDEX NOT ZERO, CONTINUE PUNCHING.
      LDA   X                OTHERWISE, RESTORE ORIGINAL
      STA   0                INDEX VALUE AND
      JMP*  PNCH             RETURN TO CALLING PROGRAM
*                            (NOTE THAT PUNCH POWER IS NOT TURNED
*                            OFF AFTER DATA BLOCK HAS BEEN PUNCHED).
X     BSS   1                STORAGE FOR INDEX VALUE.
      END
```

Line Printer Output Subroutine

```
* THIS SUBROUTINE ADVANCES PAPER ONE LINE (SINGLE SPACE) AND
* PRINTS ONE LINE OF DATA.  DATA TRANSFER IS IN WORD MODE.
* DATA ARE ASSUMED TO BE IN MEMORY BEGINNING AT LOCATION '100 AND
* EXTENDING TO LOCATION '173 (60 WORDS).
*
      REL
      SUBR  PRINT             ESTABLISH SUBROUTINE NAME.
*
PRIN  DAC   **                SUBROUTINE ENTRY POINT.
      LDA   0                 SAVE CURRENT INDEX VALUE.
      STA   X
      LDA   =-60              SET INDEX TO -60 FOR 60 WORD
      STA   0                 DATA TRANSFER TO PRINTER.
      SKS   '100              TEST FOR PRINTER BUSY (STILL IN
*                             THE PRINT CYCLE FROM PREVIOUS LINE).
      JMP   *-1               DELAY IF PRINTER BUSY.
      OCP   '1700             ADVANCE PAPER TO PUNCH IN CHANNEL 8
*                             OF FORM CONTROL TAPE (SINGLE SPACE).
      OCP   '100              ENABLE PRINTER TO LOAD IN WORD MODE.
      LDA   '100+60,1         TRANSFER 60 WORDS FROM MEMORY
      OTA   0                 TO PRINTER BUFFER
      JMP   *-1               (DELAY IF PRINTER NOT READY).
      IRS   0                 NOTE THAT DATA TRANSFER TAKES PLACE
      JMP   *-4               WHILE PAPER IS ADVANCING.
      OCP   '200              PRINT LINE.  NOTE THAT IT IS NOT NECESSARY
*                             TO TEST FOR PAPER ADVANCING PRIOR TO THE
*                             PRINT OCP.  (IN THIS CASE, THE PRINT OCP IS
*                             REDUNDANT BECAUSE 120 CHARACTERS WERE
*                             SENT TO THE PRINTER)
      LDA   X                 RESTORE ORIGINAL INDEX VALUE.
      STA   0
      JMP*  PRIN              RETURN TO CALLING PROGRAM.
X     BSS   1                 STORAGE FOR INDEX VALUE.
      END
```

Card Read (ASCII) Subroutine

```
* THIS SUBROUTINE READS ONE CARD IN HOLLERITH MODE AND THEN
* CONVERTS THE DATA FROM THE SIX BIT HOLLERITH CODE TO ASCII.
* DATA ARE PACKED TWO CHARACTERS PER WORD AND STORED IN MEMORY
* IN LOCATIONS '100 THRU '147 (40 WORDS).
*
      REL
      SUBR  CARDIN            ESTABLISH SUBROUTINE NAME.
*
CARD  DAC   **                SUBROUTINE ENTRY POINT (NOTE THAT
*                             NAME OF ENTRY POINT MATCHES FIRST
*                             FOUR LETTERS OF SUBROUTINE NAME).
      LDA   0                 SAVE CURRENT INDEX VALUE.
      STA   X
      LDA   =-40              SET INDEX TO -40 FOR 40 WORD
      STA   0                 DATA TRANSFER FROM CARD READER.
      SKS   '305              TEST FOR CARD READER NOT OPERATIONAL
*                             (HOPPER EMPTY, CARD JAM, ETC).
      JMP   *-1               DELAY UNTIL READER IS OPERATIONAL.
      SKS   '105              TEST FOR CARD READER BUSY
*                             (PREVIOUS READ CYCLE NOT COMPLETE).
      JMP   *-1               DELAY IF READER BUSY.
      OCP   5                 READ ONE CARD, HOLLERITH.
```

Card Read (ASCII) Subroutine (Cont)

```
LOOP  JST    READ              INPUT CHARACTER AND CONVERT TO ASCII
      LLR    8                 SAVE CHARACTER IN B.
      JST    READ              INPUT CHARACTER AND CONVERT TO ASCII.
      LRR    8                 MOVE FIRST CHAR FROM B TO LEFT HALF
*                              OF A, SECOND CHAR TO RIGHT HALF OF A.
      STA    *100+40,1         STORE WORD IN DATA BLOCK.
      IRS    0                 INCREMENT INDEX.
      JMP    LOOP              IF NOT ZERO, CONTINUE READING,
      LDA    X                 OTHERWISE, RESTORE
      STA    0                 ORIGINAL INDEX VALUE AND
      JMP*   CARD              RETURN TO CALLING PROGRAM.
*
READ  DAC    **                INPUT CHARACTER AND CONVERT TO ASCII.
      INA    *1005             CLEAR A AND INPUT CHARACTER
      JMP    *-1               (DELAY IF READER NOT READY).
      ARR    1                 DIVIDE CHARACTER VALUE BY TWO FOR TABLE
*                              REFERENCE, UNIT BIT TO A (1) TO INDICATE
*                              CHARACTER POSITION IN TABLE WORD.
      ADD    BASE              ADD TABLE BASE TO MODIFIED CHARACTER
*                              VALUE. BIT ONE OF BASE WORD CAUSES
*                              INDICATOR BIT TO CARRY TO C BIT IF SET.
      SSP                      FORCE A (1) TO ZERO.
      STA    ADDR              SET TABLE REFERENCE ADDRESS.
      LDA*   ADDR              GET TABLE WORD CONTAINING CONVERTED CHAR.
      SRC                      TEST C BIT FOR CHARACTER POSITION.
      LGL    8                 IF IN RIGHT HALF, LEFT-JUSTIFY,
      JMP*   READ              OTHERWISE, RETURN.
*
* HOLLERITH (SIX BIT) TO ASCII (EIGHT BIT) CONVERSION TABLE.
*
BASE  DAC*   *+1               TABLE ADDRESS, BIT ONE SET.
      BCI    5,012345678⅋
      OCT    137675            ILLEGAL, =
      BCI    7,*:1> /STUVWXYZ
      OCT    137654,124277     ILLEGAL, , ( ILLEGAL
      BCI    6,1"-JKLMNOPQR
      OCT    175644            SEMI-COLON, $
      BCI    1,*1
      OCT    137674            ILLEGAL, LESS THAN
      BCI    5,+ABCDEFGHI
      OCT    157256            UP ARROW, PERIOD
      BCI    1,)%
      OCT    156377            BACKSLASH, LEFT ARROW
X     BSS    1                 STORAGE FOR INDEX VALUE.
ADDR  BSS    1                 TABLE ADDRESS LINK.
      END
```

SECTION VII
OPERATION

The DDP-116 is designed for efficient programming operation and monitoring. The control panel (Figure 7-1), mounted on the front of the console, includes the controls and indicators required for normal system programming and operation. The maintenance panel includes the controls and indicators required for computer testing and trouble isolation. This panel is mounted behind a hinged panel on the front of the console, directly below the control panel. Controls and indicators on the maintenance panel are not used during normal system operation.

CONTROL PANEL

The control panel includes the operational controls, indicators, sense switches, etc., required for normal system programming and operation. These are listed and described in Table 7-1.

Data within the computer registers can be monitored on the control panel indicators. Data can also be manually entered into the respective registers from the control panel. Procedures for reading out and entering data into the respective registers are provided in the following paragraphs.

HARDWARE REGISTERS

Data Display

Depress the appropriate REGISTER SELECT A-, B-, P-, Y-pushbutton. The contents of the selected register is displayed on the indicators of the data display register.

Data Insertion

Enter data into the desired register as follows.

a. Set MODE switch at SINGLE INST.

b. Press appropriate DATA REGISTER pushbuttons to select desired bit pattern to be entered.

c. Press desired REGISTER SELECT pushbutton to select register into which data will be entered.

d. Press ENTER pushbutton. Data selected in step b is now entered into register selected in step c.

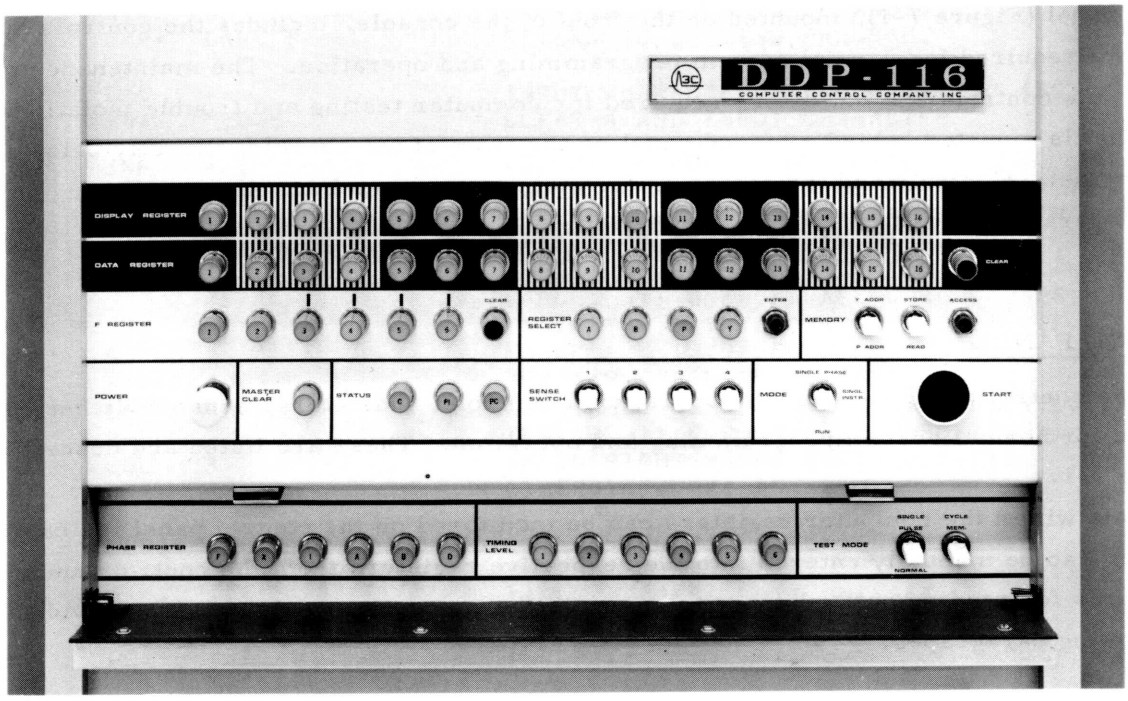


Figure 7-1. DDP-116 Control Panel, Front View

MEMORY LOCATION DATA

Single Memory Location Data Display

Display the contents of a single memory location as follows.

a. Set the MODE switch at SINGLE INST.

b. Set MEMORY Y ADDR -- P ADDR switch at its Y ADDR position.

c. Press appropriate DATA REGISTER pushbuttons to select desired memory location.

d. Press REGISTER SELECT Y-pushbutton

e. Press ENTER pushbutton. The address of the desired memory location is now in the Y-register.

f. Set MEMORY STORE-READ switch at READ.

g. Press MEMORY ACCESS pushbutton. The contents of the selected memory location are now displayed on the DATA REGISTER indicators. This action does not destroy the contents of the P-register. If a program was halted to display data, the program may be restarted by putting the machine in the RUN mode and pressing START.

Successive Memory Location Data Display

Display the contents of successive memory locations as follows.

a. Set the MODE switch to SINGLE INST.

b. Set Y ADDR -- P ADDR switch at P ADDR.

c. Select the first memory location to be displayed by depressing appropriate DATA REGISTER pushbuttons.

d. Press REGISTER SELECT P pushbutton.

e. Press ENTER pushbutton.

f. Set MEMORY STORE-READ switch at READ.

g. Press MEMORY ACCESS pushbutton. The contents of the selected memory locations are displayed on the DATA REGISTER indicators.

h. Repeat step g to display the contents of successive memory locations on the DATA REGISTER indicators. This action destroys the contents of the P-register; therefore, in order to restart a program that was halted to display data, the P-register must be reloaded as explained under hardware registers.

Single Memory Location Data Insertion/Change

Enter data into a single memory location as follows.

a. Set MODE switch at SINGLE INST.

b. Set MEMORY Y ADDR -- P ADDR switch at its Y ADDR position.

c. Press appropriate DATA REGISTER pushbuttons to select desired memory location.

d. Press REGISTER SELECT Y-pushbutton.

e. Press ENTER pushbutton.

f. Set MEMORY STORE-READ switch at STORE.

g. Press DATA REGISTER pushbuttons corresponding to the data to be entered into memory.

h. Press MEMORY ACCESS pushbutton. This action does not destroy the contents of the P-register. If a program was halted to enter data, the program may be restarted by putting the machine in the RUN mode and pressing START.

Successive Memory Location Data Insertion/Change

Enter data into or change data within successive memory locations as follows.

a. Set MODE switch at SINGLE INST.

b. Set MEMORY Y ADDR -- P ADDR switch at P ADDR.

c. Press appropriate DATA REGISTER pushbuttons to select the first memory location to be changed.

d. Press REGISTER SELECT P pushbutton.

e. Press ENTER pushbutton.

f. Set MEMORY STORE-READ switch at STORE.

g. Press DATA REGISTER pushbuttons corresponding to data to be entered into memory.

h. Press MEMORY ACCESS pushbutton.

i. Repeat steps g and h to write into successive memory locations. This action destroys the contents of the P-register; therefore, in order to restart a program that was halted to enter data, the P-register must be reloaded as explained in the hardware register section.

START-UP PROCEDURE

a. Press the POWER pushbutton on the DDP-116 control panel.

b. Press MASTER CLEAR pushbutton.

c. Set registers to desired value as described above.

d. Set MODE switch at RUN.

e. Press START pushbutton.

Table 7-1.
Control Panel Controls and Indicators and Their Functions

Control or Indicator	Function
DATA REGISTER pushbuttons/lamps	Pushbuttons (16) are used to enter data into any register or core memory location when the machine is halted. Each pushbutton corresponds to 1 bit of the 16-bit memory information register (M-register). Depressing a pushbutton causes its lamp to light and enter a binary 1 in the corresponding position of the M-register.
CLEAR pushbutton	Clears M-register when pressed
REGISTER SELECT P-A-B-Y pushbuttons/lamps	Select register contents to be displayed on DATA DISPLAY REGISTER indicators. A = A-register B = B-register P = P-register (program register) Y = Y-register (memory address)
MEMORY Y ADDR -- P ADDR switch	Two-position toggle switch used with the MEMORY ACCESS pushbutton. When in the P ADDR position, it causes the contents of the P-counter to be used as the memory address in accessing memory. After each memory access the contents of the P-register are incremented by 1. In the Y ADDR position, it causes the contents of the Y-register to be used as the memory address.
MEMORY STORE-READ switch	Two-position toggle switch used with the MEMORY ACCESS pushbutton. Informs the memory of the type of the ensuing memory cycle. Switch in the READ position causes a memory read cycle. Switch in the STORE position causes a memory write cycle.
MEMORY ACCESS pushbutton	In conjunction with the STORE-READ toggle switch, reads out the contents of a core location or transfers information between the M-register and a core location. This function can be performed only in the single instruction mode.
POWER pushbutton	Controls main frame ac power
MASTER CLEAR pushbutton	Clears all registers. Resets the timing level generator and phase register to F cycle and resets certain control flip-flops.
SENSE SWITCH 1-2-3-4 switches	Four two-position toggles can be set by the operator to a set or reset condition. Switch positions are sensed under program control.
SINGLE PHASE-RUN SINGLE INST switch	Selects mode of operation a. Run mode is the normal operating mode when executing instructions in the automatic mode of operation. b. Single instruction mode, if selected while machine is running, causes the computer to halt upon completion of the current instruction. If the machine is halted in the single instruction mode, each depression of the START pushbutton causes the machine to execute the next instruction and halt.

Table 7-1. (Cont)
Control Panel Controls and Indicators and Their Functions

Control or Indicator	Function
SINGLE PHASE-RUN SINGLE INST switch (cont)	c. Single phase mode causes the machine to halt upon completion of the current phase. When the machine is halted and the single-phase mode is selected, each depression of the START pushbutton causes the computer to execute one minor cycle. This mode enables single-stepping through each status level of the program since all normal functions, including memory cycles, are executed.
START pushbutton/indicator	Pushbutton starts machine in all modes. Indicator lights when machine is halted and is extinguished when started.
ENTER pushbutton	Transfers data, from the DATA REGISTER pushbutton indicators, to the hardware registers.
DATA DISPLAY REGISTER indicators	Display contents of any hardware register or core location selected by a manual display operation
F-REGISTER pushbuttons/lamps	Pushbuttons used to enter data into the F-register. The lamps continuously display the contents of the F-register (generally used for maintenance work)
F CLEAR pushbutton	Clears F-register when pressed
C STATUS indicator	Indicates state of C-bit. Indicator is lighted when C-bit is set (ONE), extinguished when C-bit is reset (ZERO).
PI STATUS indicator	Indicates state of priority interrupt. Indicator is lighted when priority interrupt system is enabled, extinguished when priority interrupt system is disabled.
PC indicator	Indicates state of I/O parity. When lighted, indicates I/O parity error; when extinguished, no I/O parity error exists. Also used with the memory parity option.
Maintenance Panel	
TIMING LEVEL indicators	Indicates the state of the timing level generator
PHASE REGISTER indicators	Indicates the state of the phase register, i.e., which cycle of the instruction the machine is executing
CYCLE MEM. switch	Performs a simulated high-speed memory access identical to that of pushing the memory access button at a 1 mc rate.
SINGLE PULSE-NORMAL switch	Permits stepping one timing level at a time (through the timing level) generator (with the RUN SINGLE INSTRUCTION-SINGLE PHASE switch in the SINGLE PHASE position) by pushing the START button.

APPENDIX A
I/O CODE ASSIGNMENTS

OCP CODES

OCP	0000	Enable Line Printer Buffer for Character Mode
OCP	0100	Enable Line Printer Buffer for Word Mode
OCP	0200	Print One Line on Line Printer
OCP	0300	Set DMC Mode on Line Printer
OCP	0700	Reset DMC End-of-Transmission Interrupt on Line Printer
OCP	1000	Advance Paper to Hole in Channel 1 on Line Printer (Top of Form)
OCP	1100	Advance Paper to Hole in Channel 2 on Line Printer
OCP	1200	Advance Paper to Hole in Channel 3 on Line Printer
OCP	1300	Advance Paper to Hole in Channel 4 on Line Printer
OCP	1400	Advance Paper to Hole in Channel 5 on Line Printer
OCP	1500	Advance Paper to Hole in Channel 6 on Line Printer
OCP	1600	Advance Paper to Hole in Channel 7 on Line Printer
OCP	1700	Advance Paper to Hole in Channel 8 on Line Printer
OCP	0001	Start Paper Tape Reader in Forward Direction
OCP	0101	Stop Paper Tape Reader
OCP	0201	Start Paper Tape Reader in Reverse Direction
OCP	0301	Set DMC Mode on Paper Tape Reader
OCP	0701	Reset DMC End-of-Transmission Interrupt on Paper Tape Reader
OCP	0002	Enable Paper Tape Punch
OCP	0102	Turn Paper Tape Punch Power Off
OCP	0202	Set DMC Mode on Paper Tape Punch
OCP	0402	Reset DMC End-of-Transmission Interrupt on Paper Tape Punch
OCP	0003	Enable Memory Lockout
OCP	0004	Enable ASR-33 in Input Mode
OCP	0104	Enable ASR-33 in Output Mode
OCP	0504	Set DMC (ASCII) Mode on ASR-33
OCP	0704	Reset DMC End-of-Transmission Interrupt on ASR-33

OCP CODES (Cont)

OCP	0005	Read One Hollerith Card on Card Reader
OCP	0105	Read One Binary Card on Card Reader
OCP	0305	Set DMC Mode on Card Reader
OCP	0705	Reset DMC End-of-Transmission Interrupt on Card Reader
OCP	0006	Enable Card Punch
OCP	0306	Set DMC Mode on Card Punch
OCP	0706	Reset DMC End-of-Transmission Interrupt on Card Punch
OCP	0010	Read BCD on MTT 1
OCP	0011	Read BCD on MTT 2
OCP	0012	Read BCD on MTT 3
OCP	0013	Read BCD on MTT 4
OCP	0014	Read BCD on MTT 5
OCP	0015	Read BCD on MTT 6
OCP	0016	Read BCD on MTT 7
OCP	0017	Read BCD on MTT 8
OCP	011X	Read Binary on MTT X+1
OCP	021X	Read Binary 3 Char./Word on MTT X+1
OCP	031X	Set DMC Mode on MTT X+1
OCP	041X	Write BCD on MTT X+1
OCP	051X	Write Binary on MTT X+1
OCP	061X	Write End-of-File on MTT X+1
OCP	071X	Reset DMC End-of-Transmission Interrupt on MTT X+1
OCP	101X	Write Binary 3 Char./Word on MTT X+1
OCP	111X	Space Forward One Record on MTT X+1
OCP	121X	Space Forward One File on MTT X+1
OCP	141X	Rewind on MTT X+1
OCP	151X	Backspace One Record on MTT X+1
OCP	161X	Backspace One File on MTT X+1
OCP	0020	Reset Real Time Clock Interrupt Line
OCP	013X	Optional OCP Line No. 1
OCP	023X	Optional OCP Line No. 2
OCP	043X	Optional OCP Line No. 3

The least significant octal digit identifies which magnetic tape transport is to be selected.

SKS CODES

SKS	0000	Skip If Line Printer Is Ready
SKS	0100	Skip If Line Printer Is Not Busy
SKS	0200	Skip If Line Printer Has Not Detected a Parity Error
SKS	0300	Skip If Line Printer Paper Is Not Advancing
SKS	0400	Skip If Line Printer Is Not Interrupting
SKS	0001	Skip If Paper Tape Reader Is Ready
SKS	0401	Skip If Paper Tape Reader Is Not Interrupting
SKS	0002	Skip If Paper Tape Punch Is Ready
SKS	0102	Skip If Paper Tape Punch Is Enabled
SKS	0402	Skip If Paper Tape Punch Is Not Interrupting
SKS	0003	Skip If Memory Lockout Is Enabled
SKS	0004	Skip If ASR-33 Is Ready in ASCII Mode
SKS	0104	Skip If ASR-33 Is Not Busy
SKS	0204	Skip If ASR-33 Is Ready in Binary Mode
SKS	0404	Skip If ASR-33 Is Not Interrupting
SKS	0504	Skip If Stop Code Was Not Read on ASR-33
SKS	0005	Skip If Card Reader Is Ready
SKS	0105	Skip If Card Reader Is Not Busy
SKS	0205	Skip If Card Reader Stop Flip-Flop Is Not Set
SKS	0305	Skip If Card Reader Is Operational
SKS	0405	Skip If Card Reader Is Not Interrupting
SKS	0006	Skip If Card Punch Is Ready
SKS	0106	Skip If Card Punch Is Not Busy
SKS	0206	Skip If Card Punch Is Not Ready to Punch New Row
SKS	0306	Skip If Card Punch Is Operational
SKS	0406	Skip If Card Punch Is Not Interrupting
SKS	0007	Skip If Power Failure Is Not Interrupting

SKS CODES (Cont)

SKS	0010	Skip If MTT 1 Is Ready
SKS	0110	Skip If MTT 1 Is Not Busy
SKS	0210	Skip If MTT 1 Has Not Detected a Parity Error
SKS	0310	Skip If MTT 1 Has Not Detected Beginning of Tape
SKS	0410	Skip If MTT 1 Is Not Interrupting
SKS	0510	Skip If MTT 1 Has Not Detected End of Tape
SKS	0610	Skip If MTT 1 Has Not Detected End of File
SKS	1410	Skip If MTT 1 Is Not Rewinding
SKS	0020	Skip If Real Time Clock Is Not Interrupting

INA CODES

INA	0001	Input from Paper Tape Reader
INA	1001	Clear Register A and Input from Paper Tape Reader
INA	0004	Input in ASCII from ASR-33
INA	0204	Input in Binary from ASR-33
INA	1004	Clear Register A and Input in ASCII from ASR-33
INA	1204	Clear Register A and Input in Binary from ASR-33
INA	0005	Input from Card Reader
INA	1005	Clear Register A and Input from Card Reader
INA	0010	Input from MTT 1
INA	1010	Clear Register A and Input from MTT 1

OTA CODES

OTA	0000	Output to Line Printer
OTA	0002	Output to Paper Tape Punch
OTA	0003	Output to Memory Lockout Mask 1
OTA	0103	Output to Memory Lockout Mask 2
OTA	0203	Output to Memory Lockout Mask 3
OTA	0303	Output to Memory Lockout Mask 4
OTA	0004	Output in ASCII to ASR-33
OTA	0204	Output in Binary to ASR-33
OTA	0006	Output to Card Punch
OTA	0010	Output to MTT 1

OTA CODES (Cont)

OTA 0020 (SMK1) Set First Group of Mask Flip-Flops for Program Interrupt)

OTB Bit Assignments

Bit	Interrupt Line
1	TCU 1
2	TCU 2
3	TCU 3
4	TCU 4
5	Parallel Channel 1
6	Parallel Channel 2
7	Parallel Channel 3
8	Unassigned
9	Paper Tape Reader
10	Paper Tape Punch
11	ASR-33
12	Card Reader
13	Card Punch
14	Line Printer
15	Memory Parity
16	Real Time Clock

OTA 0021 Output to Watchdog Timer

OTA 0120 (SMK2) Controls Interrupt Lines 1-16

OTB Bit Assignments

Bit	Interrupt Line
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

OTA CODES (Cont)

OTA 0220 (SMK3) Controls Interrupt Lines 17-32

OTB Bit Assignments

Bit	Interrupt Line
1	17
2	18
3	19
4	20
5	21
6	22
7	23
8	24
9	25
10	26
11	27
12	28
13	29
14	30
15	31
16	32

OTA 0320 (SMK4) Controls Interrupt Lines 33-38 (As Above)

OTA 0420 (SMK5) Controls Interrupt Lines 49-64 (As Above)

APPENDIX B
DDP-116 CHARACTER CODES

Character	Card Code (Hollerith)	ASCII Code	Mag. Tape Code
0	0	260	12*
1	1	261	01
2	2	262	02
3	3	263	03
4	4	264	04
5	5	265	05
6	6	266	06
7	7	267	07
8	8	270	10
9	9	271	11
A	12-1	301	61
B	12-2	302	62
C	12-3	303	63
D	12-4	304	64
E	12-5	305	65
F	12-6	306	66
G	12-7	307	67
H	12-8	310	70
I	12-9	311	71
J	11-1	312	41
K	11-2	313	42
L	11-3	314	43
M	11-4	315	44
N	11-5	316	45
O	11-6	317	46
P	11-7	320	47
Q	11-8	321	50
R	11-9	322	51

*When writing magnetic tapes in even parity (BCD) mode, 00_8 is written as 12_8; conversely, when reading in even parity, 12_8 is written as 00_8.

APPENDIX B (Continued)
DDP-116 CHARACTER CODES

Character	Card Code (Hollerith)	ASCII Code	Mag. Tape Code
S	0-2	323	22
T	0-3	324	23
U	0-4	325	24
V	0-5	326	25
W	0-6	327	26
X	0-7	330	27
Y	0-8	331	30
Z	0-9	332	31
Space	Blank	240	20
!	8-6	241*	16
"	0-8-7	242*	37
$	11-8-3	244*	53
%	12-8-5	245*	75
'	8-4	247*	14
(	0-8-4	250*	34
)	12-8-4	251*	74
*	11-8-4	252*	54
+	12	253*	60
,	0-8-3	254	33
-	11	255	40
.	12-8-3	256	73
/	0-1	257	21
:	8-5	272	15
;	11-8-2	273	52
<	11-8-7	274*	57
=	8-3	275*	13
>	8-7	276*	17
?		277*	

*Upper case characters on ASR-33

Character	Card Code (Hollerith)	ASCII Code	Mag. Tape Code
[	11-8-5	333 (Note 1)	55
\	12-8-6	334 (Note 2)	76
]	0-8-6	335 (Note 3)	36
↑	12-8-2	336 (Note 4)	72
←	12-8-7	337	77

End of Msg	203 (Note 5)
Bell	207 (Note 6)
Line feed	212 (Note 6)
Return	215 (Note 6)
X on	221 (Note 6)
X off	223 (Note 6)
Rub out	377 (Note 6)

Notes:
1. Upper case VT
2. Upper case "FORM"
3. Upper case M on ASR-33
4. Not on ASR-33
5. Control-C on ASR-33
6. ASR-33 function controls

APPENDIX C
ASCII CODE*

Standard Code

Bits b_4 b_3 b_2 b_1	Row	Column 0 (000)	Column 1 (001)	Column 2 (010)	Column 3 (011)	Column 4 (100)	Column 5 (101)	Column 6 (110)	Column 7 (111)
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	DC3	Device Control 3
SOH	Start of Heading (CC)	DC4	Device Control (stop)
STX	Start of Text (CC)	NAK	Negative Acknowledge (CC)
ETX	End of Text (CC)	SYN	Synchronous Idle (CC)
EOT	End of Transmission (CC)	ETB	End of Transmission Block (CC)
ENQ	Enquiry (CC)	CAN	Cancel
ACK	Acknowledge (CC)	EM	End of Medium
BEL	Bell (audible or attention signal)	SS	Start of Special Sequence
BS	Backspace (FE)	ESC	Escape
HT	Horizontal Tabulation (punched card skip) (FE)	FS	File Separator (IS)
LF	Line Feed (FE)	GS	Group Separator (IS)
VT	Vertical Tabulation (FE)	RS	Record Separator (IS)
FF	Form Feed (FE)	US	Unit Separator (IS)
CR	Carriage Return (FE)	DEL	Delete
SO	Shift Out		
SI	Shift In		
DLE	Data Link Escape (CC)		
DC1	Device Control 1		
DC2	Device Control 2		

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

Graphic Characters

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

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

APPENDIX D
INSTRUCTIONS

Octal Code	Mnemonic	Instruction	Type	Execution Time (μsec)	Page
01	JMP	Unconditional Jump	MR	1.7	2-24
02	LDA	Load A	MR	3.4	2-3
03	ANA	AND to A	MR	3.4	2-8
04	STA	Store A	MR	3.4	2-4
05	ERA	Exclusive OR to A	MR	3.4	2-8
06	ADD	Add	MR	3.4	2-4
07	SUB	Subtract	MR	3.4	2-6
10	JST	Jump and Store Location	MR	5.1	2-25
11	CAS	Compare Memory and A	MR	5.1	2-17
12	IRS	Increment, Replace Memory and Skip	MR	5.1	2-6
13	IMA	Interchange Memory and A	MR	5.1	2-3
14	OCP	Output Command Pulse	IOT	3.4	2-22
16	MPY*	Multiply*	MR	9.5	2-6
17	DIV*	Divide*	MR	17.9(max)	2-4
34	SKS	Skip If Ready Line Set	IOT	3.4	2-24
54	INA	Input to A	IOT	5.1	2-22
74	OTA	Output from A	IOT	5.1	2-24
0400	LRL	Long Right Logical Shift	S	1.7+0.34n	2-13
0401	LRS	Long Arithmetic Right Shift	S	1.7+0.34n	2-15
0402	LRR	Long Right Rotate	S	1.7+0.34n	2-14
0404	LGR	Logical Right Shift	S	1.7+0.34n	2-9
0405	ARS	Arithmetic Right Shift	S	1.7+0.34n	2-12
0406	ARR	Logical Right Rotate	S	1.7+0.34n	2-11
0410	LLL	Long Left Logical Shift	S	1.7+0.34n	2-12
0411	LLS	Long Arithmetic Left Shift	S	1.7+0.34n	2-15
0412	LLR	Long Left Rotate	S	1.7+0.34n	2-13
0414	LGL	Logical Left Shift	S	1.7+0.34n	2-8
0415	ALS	Arithmetic Left Shift	S	1.7+0.34n	2-11
0416	ALR	Logical Left Rotate	S	1.7+0.34n	2-9
000000	HLT	Halt	G		2-20
000041	SCA*	Shift Counter to A*	G	1.7	2-3
000101	NRM*	Normalize*	G	2.38+0.34n	2-16
000201	IAB	Interchange A and B	G	1.7	2-3
000401	ENB	Enable Interrupt	G	1.7	2-19

*Optional (part of high-speed arithmetic option)

n = Number of shifts

Octal Code	Mnemonic	Instruction	Type	Execution Time (μsec)	Page
001001	INH	Inhibit Interrupt	G	1.7	2-20
100001	SRC	Skip If C Reset	G	1.7	2-26
100002	SR4	Skip If Sense Switch No. 4 Reset	G	1.7	2-29
100004	SR3	Skip If Sense Switch No. 3 Reset	G	1.7	2-29
100010	SR2	Skip If Sense Switch No. 2 Reset	G	1.7	2-29
100020	SR1	Skip If Sense Switch No. 1 Reset	G	1.7	2-29
100036	SSR	Skip If All Sense Switches Reset	G	1.7	2-29
100040	SZE	Skip If A Zero	G	1.7	2-29
100100	SLZ	Skip If LSB A Zero	G	1.7	2-27
100400	SPL	Skip If A Sign Plus	G	1.7	2-28
101000	NOP	No Operation	G	1.7	2-20
101001	SSC	Skip If C-Bit Set	G	1.7	2-27
101002	SS4	Skip If Sense Switch No. 4 Set	G	1.7	2-28
101004	SS3	Skip If Sense Switch No. 3 Set	G	1.7	2-28
101010	SS2	Skip If Sense Switch No. 2 Set	G	1.7	2-28
101020	SS1	Skip If Sense Switch No. 1 Set	G	1.7	2-28
101036	SSS	Skip If Any Sense Switch Set	G	1.7	2-29
101040	SNZ	Skip If a Non-Zero	G	1.7	2-28
101100	SLN	Skip If LSB of A Non-Zero	G	1.7	2-27
101400	SMI	Skip If A Sign Minus	G	1.7	2-27
140024	CHS	Change Sign of A	G	1.7	2-19
140040	CRA	Clear A	G	1.7	2-1
140100	SSP	Set A Sign Plus	G	1.7	2-21
140200	RCB	Reset C-Bit	G	1.7	2-21
140240	CCA	Clear C and A	G	1.7	2-1
140320	CSA	Copy Sign to C-Bit and Set A Sign Plus	G	1.7	2-19
140401	CMA	Complement A	G	1.7	2-19
140407	TCA	2's Complement A	G	1.7	2-22
140500	SSM	Set A Sign Minus	G	1.7	2-21
140600	SCB	Set C-Bit	G	1.7	2-21
141206	AOA	Add One to A	G	1.7	2-17
141216	ACA	Add C to A	G	1.7	2-17