Honeywell

SERIES 16

HARDWARE

MODELS 316 AND 516
PROGRAMMERS' REFERENCE MANUAL

Honeywell

SERIES 16

MODELS 316 AND 516 PROGRAMMERS' REFERENCE MANUAL

SUBJECT:

Programming Information for Series 16 Models 316 and 516. Includes Hardware Description, Computer Organization, Standard Instructions, and Standard Mainframe Options.

DATE:

November, 1970

ORDER NUMBER: BX47 (Formerly M-490)

Rev. 0

DOCUMENT NUMBER: 70130072156

This manual describes the Models 316 and 516 and their more common options from the point of view of the programmer. Readers should have some familiarity with computers but need no assembly language experience.

Honeywell intends to make available separate programming manuals for all standard input/output devices, other language processors, and major program packages.

The 316 and 516 Operators Guide, Order No. M-491, is a companion volume describing computer operation. The DAP-16 and DAP-16 Mod 2 Assembly Language Manual, Order No. M-1756, is another companion volume describing the assembly language.

REVISION HISTORY

New Revision Level of Manual	Change No.	Effective Date	Pages Affected by Revision
B	7518	Oct. 1969	iv, 2-16, 2-33, 2-34
C	7871	Feb. 1970	iv, x, 1-12, 2-20, 2-21, 2-22, 2-24, 2-26 thru 2-41
D	8730	Sept. 1970	All
E	8998	Nov. 1970	Title page, copyright page, 2-1, 2-8, 2-9, 2-11, 2-13, 3-7, 3-9, 4-8, 4-12

CONTENTS

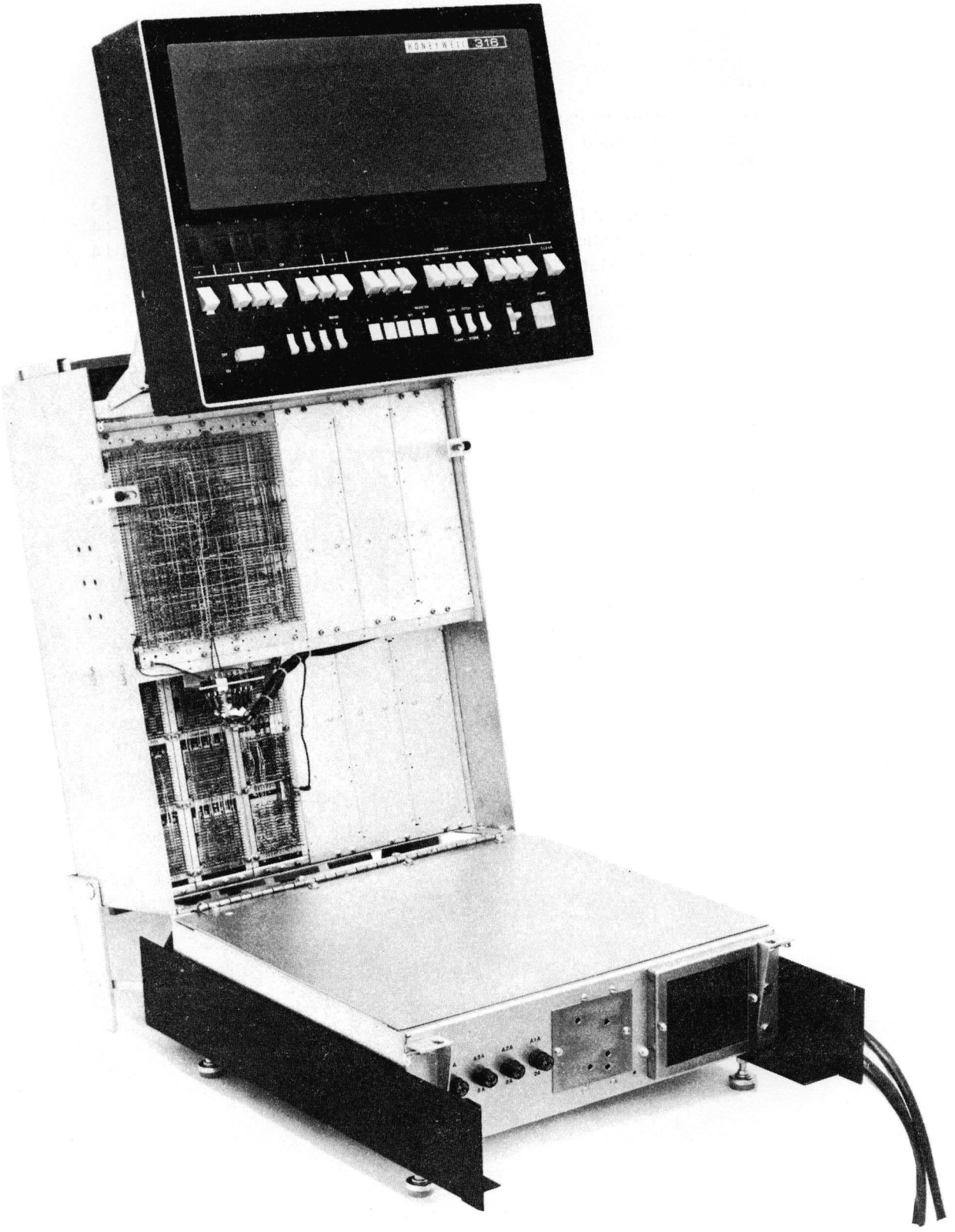
CONTENTS (Cont.)

CONTENTS (Cont.)

ILLUSTRATIONS

TABLES

TABLES (Cont.)



H316 General Purpose Computer

SECTION I
INTRODUCTION

The Honeywell 316/516 computers are designed for both open-shop scientific and commercial applications and real-time on-line data processing and control. Modular design, a flexible I/O structure, and a powerful instruction repertoire enable these machines to be tailored to a broad variety of applications both on-line and off-line. These applications include data reduction, process control, instrumentation, simulations, and computation.

General characteristics include fully parallel organization, indexing, multi-level indirect addressing, a powerful I/O system, a comprehensive 72-command instruction repertoire, and straightforward logic for easy system interface and field expansion. Selected optional capabilities are designed with plug-in modularity to permit custom tailoring at minimum expense.

The efficient DAP-16 Mod 2 Assembler* allows the programmer to specify either a one- or two-pass assembly for the same source program. One pass is used with a basic system. Two passes are used for systems with high-speed input devices or where more detailed listings are required. The programmer can directly address all of memory with his source program through the use of a desectorizing loader.

DAP-16 Mod 2 provides numerous pseudo-operations to supplement the standard instructions. These pseudo-operations allow the programmer to express concepts which do not have any counterparts in machine language. Among the important capabilities of these instructions are programmer-defined assembly and loader controls, conditional assembly, data definitions, and program linkages.

HARDWARE DESCRIPTION

Type

Parallel binary

Addressing

Single address with indexing and indirect addressing

Word Length

16 bits

Machine Code

TWOs complement

Memory Type

Magnetic core

---

*DAP-16 Mod 2 is an extension of DAP-16 as previously offered. No changes are required in existing source programs to make them compatible with this assembler. It is described in the DAP-16 and DAP-16 Mod 2 Assembler Manual.

Memory Size

4096, 8182, 12,288, 16,384, 24,576, or 32,768 words (DDP-516)
4096 to 32,768 words in 4096-word modules (H316)

Memory Cycle Time

0.96 μs (DDP-516)
1.60 μs (H316)

Speed

Add:	1.92 μs (DDP-516) 3.2 μs (H316)
Subtract:	1.92 μs (DDP-516) 3.2 μs (H316)
Multiply (hardware option):	5.28 μs (max) (DDP-516) 8.8 μs (max) (H316)
Divide (hardware option):	10.56 μs (max) (DDP-516) 17.6 μs (max) (H316)

Standard Peripheral Equipment

ASR-33 or -35 Teletype Unit provides the following capabilities:

a. Reads paper tape at 10 characters per second
b. Punches paper tape at 10 characters per second
c. Types at 10 characters per second
d. Keyboard input
e. Off-line paper-tape preparation, reproduction, and listing

Common Optional Peripheral Equipment

300 character-per-second photoelectric paper tape reader
110 character-per-second paper tape punch
300-950 line-per-minute high-speed printers
200 and 800 card-per-minute card readers
400/100 card-per-minute card reader/punch
2, 10, and 20-surface moving head disc files
Fixed head disc file
Fixed head drum
26, 36, and 80 ips magnetic tape units (7- and 9-track)
Alphanumeric keyboard/CRT display
Point plotting storage tube display

General Input/Output Options

Input, Output, and I/O channels, buffered and unbuffered
Sense lines and control pulses
Priority interrupt and memory increment lines
Data line controllers, both synchronous and asynchronous

Central Processor Options

Real-Time Clock
Memory Lockout
Extended Arithmetic
Priority Interrupt
Extended Addressing

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

Three modes are available for data transfer between peripheral devices or I/O channels and the Honeywell 316/516:

a. Single-word transfer with or without interrupt
b. Direct multiplex control (DMC) (optional)
c. Direct memory access (DMA) optional (DDP-516 only)

Interrupt

Single interrupt line standard; up to 48 optional priority interrupts are available.

Power Failure Protection

Power failure interrupt standard. Core memory is protected against loss of information on ac power failure. One millisecond of operation after power failure.

SOFTWARE DESCRIPTION

Executives and Operating Systems

BOS Batch Operating System (Core and Disc)
MOS Magnetic Tape Operating System
DOP Drum Operating System
OLERT On-Line Real-Time Operating System
RTX-16 Real-Time Executive

Language Processors

DAP-16 DDP Assembly Program
DAP-16M2 DDP Assembly Program Mod 2
FORTRAN IV ASA Standard FORTRAN
MAC Macro Preprocessor Program
SSUP Source Update Program
XREF Concordance Generator

Libraries

Statistical
Mathematical
Input/Output
FORTRAN run-time

SERIES 16 PROGRAMMERS REFERENCE MANUALS

This manual is one of a series of Programmers Reference Manuals. Some of these manuals are listed in the Appendix. The first manual in the Appendix, the Programmers Reference Guide, is a frequently-revised listing of all the available manuals in the series.

This 316/516 Programmers Reference Manual describes the computer mainframe and the more common options from the point of view of the programmer. This document supersedes the December 1966 edition of the DAP-16 Manual and the DDP-516 Programmers Reference Manual.

This revision supersedes all previous revisions. Some material which was formally in the 316/516 Programmers Reference Manual is now in the DAP-16 Assembler Manual and the 316/516 ASR Teletypewriter, High-Speed Paper Tape Reader, and High-Speed Paper Tape Punch Programmers Manual.

CONVENTION

Octal numbers are preceded by an apostrophe or followed by the word "octal," except where there can be no ambiguity.

SECTION II
COMPUTER ORGANIZATION

This section contains the basic description of the Honeywell 316/516 computers.

SYSTEM DESCRIPTION

Figure 2-1 is a simplified block diagram of the Honeywell 316/516. The details of the control logic are not shown. The components shown are described below.

Memory

A random access magnetic-core memory expandable from 4096 to 32,767 sixteen-bit words.

M Register.—A 16-bit register which temporarily stores information transferred to and from memory.

Y Register.—A 16-bit register which holds the address of the memory cell currently being accessed. (Bit 1 is unused. Bit 2 is used only when Extended Addressing is installed.)

Adder

A central adder used for all arithmetic tasks including incrementing the contents of the program counter.

Programmable Registers

A Register.—A 16-bit register used as the primary arithmetic, logic, and input/output register of the computer.

B Register.—A 16-bit register used as a secondary working register. It is also used for the low-order part of doubleword (32-bit) operands.

C Bit.—The carry/borrow bit for arithmetic operations. It is also used in shift operations to store the last bit shifted. It may be directly modified and tested.

X Register.—A 16-bit index register which may be used in forming the effective address of any memory-reference instruction except those which access it directly.

P Register.—A 15-bit program counter containing the address of the next instruction to be executed.

I/O Bus

The following three busses plus control lines described in Section III comprise the Input/ Output Bus.

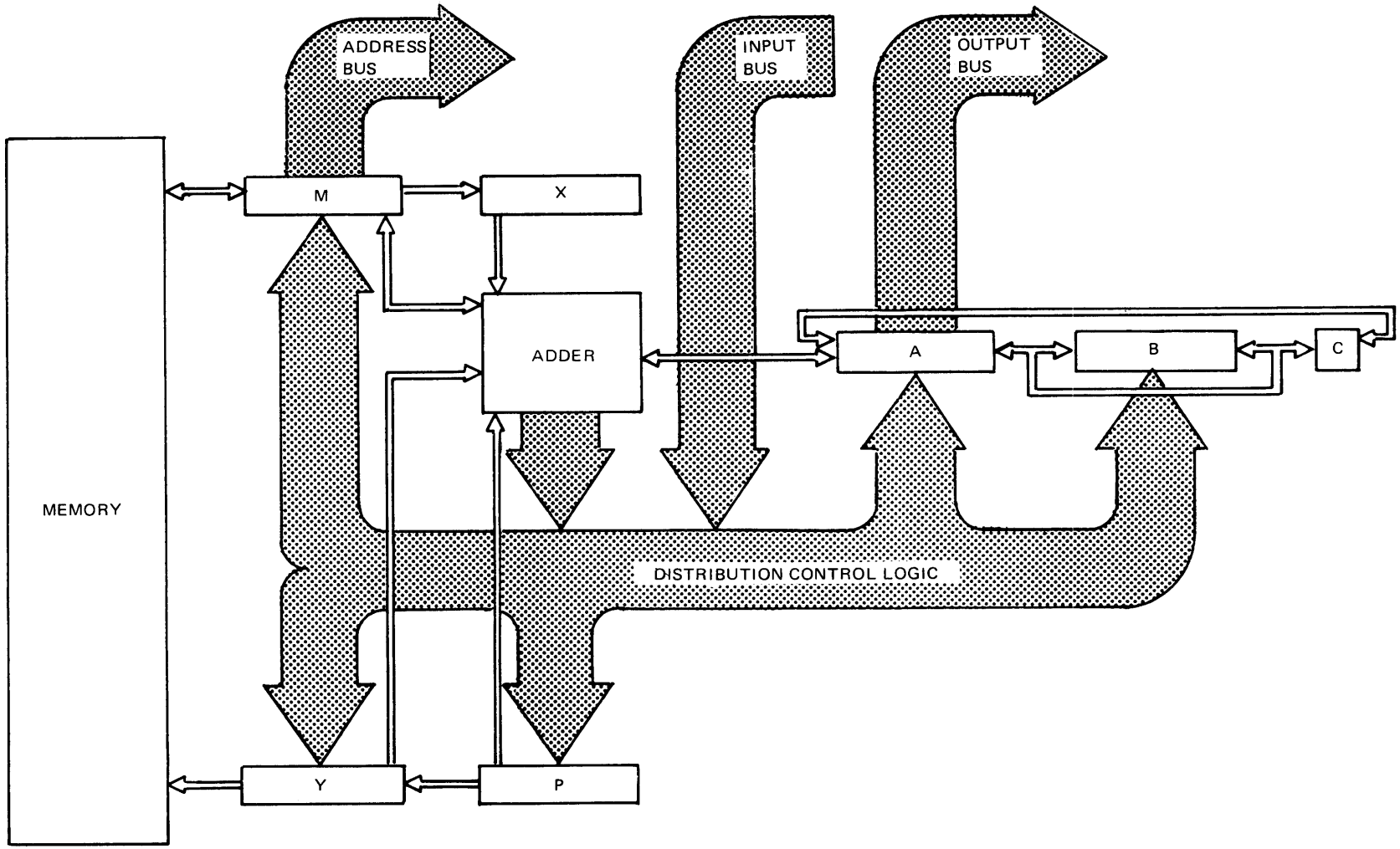


Figure 2-1. Honeywell 316/516 Simplified Block Diagram

Address Bus.—Ten lines which always reflect the 10 least-significant bits of the M-Register. By convention, bits 7 through 10 are used in input/output to define the function to be performed by the input/output device and bits 11 through 16 designate the controller address.

Input Bus.—Sixteen lines used to transmit data from an input/output device to the computer. The data are gated to the A-Register during input.

Output Bus.—Sixteen lines that always reflect the contents of the A-Register. During output operations they are used to transmit data. During DMC cycles they reflect the contents of the M-Register.

WORD FORMATS

The Honeywell 316/516 computers operate upon 16-bit words with the bits numbered 1-16 from left (most significant) to right (least significant). (See Figure 2-2.)

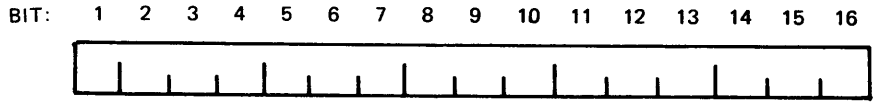


Figure 2-2. General Word Format

Fixed Point, Single Precision

Single precision numbers employ one 16-bit word. They use TWOs complement representation. (See Figure 2-3.)

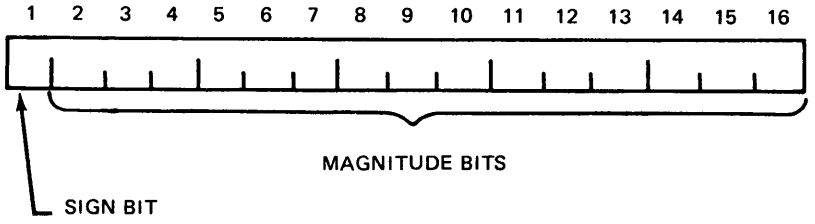


Figure 2-3. Fixed Point, Single Precision Data Word

When performing arithmetic operations upon fixed-point words the computer considers the binary point to be between bits 1 and 2. The unique value '100000 is assigned to -1.0. Therefore the range of fixed-point numbers is from -1 to $+1-2^{-15}$ in steps of 2^{-15}. When the binary point is considered to lie following bit 16 this range is -32,768 to +32,767.

Fixed Point, Double Precision

Double precision numbers, contained in two 16-bit words, also use TWOs complement representation. Bit 1 of the second (least-significant) word must be 0. This requirement is automatically handled by the optional High-Speed Arithmetic Unit. (See Figure 2-4.)

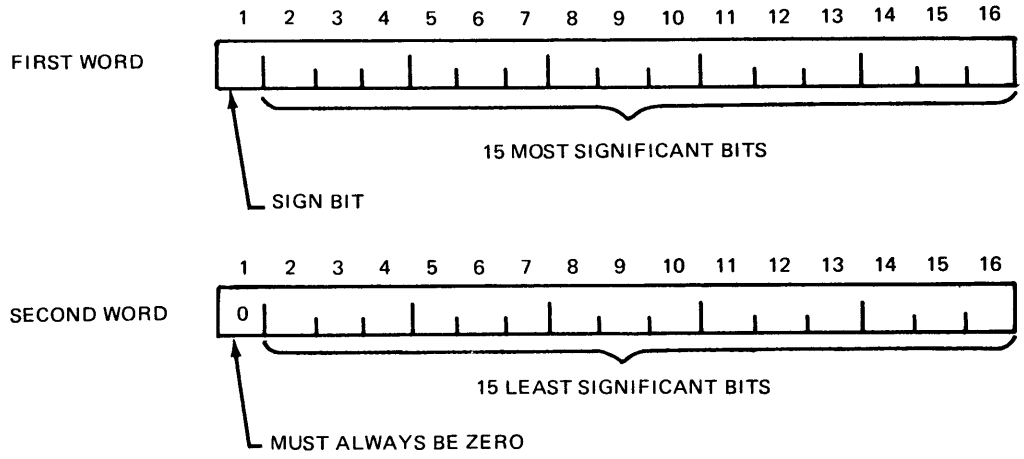


Figure 2-4. Fixed Point, Double Precision Data Word

When performing arithmetic operations upon fixed-point double precision words the computer considers the binary point to be between bits 1 and 2 of the first word. Since bit 1 of the second word carries no information, there are 30 bits of significance. The unique value '100000 000000 is assigned the value -1.0. Therefore the range of fixed-point double precision numbers is from -1 to $+1-2^{-30}$ in steps of 2^{-30}. When the binary point is considered to lie following bit 16 of the second word this range is -1,073,741,824 to +1,073,741,823.

Logical Data

Up to 16 bits of logical data (such as the condition of 16 binary indicators) can be stored in a single data word. In this case, bit 1 of the word does not represent the sign. This type of data is generally not treated arithmetically by the program; but logically, by means of the Boolean operators AND and EXCLUSIVE OR.

Floating Point and Complex

The format illustrated in Figure 2-5 has been defined for floating point numbers. These are used by the DAP-16 Mod 2 Assembler and FORTRAN IV. When complex numbers are required, two single precision floating point numbers are used, with the first being the real part and the second the imaginary part.

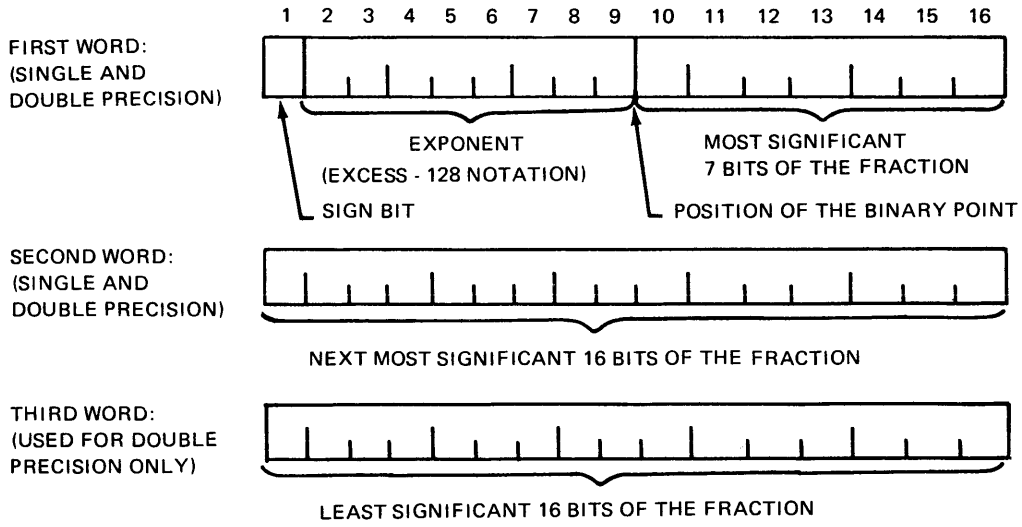


Figure 2-5. Floating Point Word Formats

Instruction Words

There are four classes of instructions; Memory Reference Instructions (MR), Shift Instructions (SH), Input/Output Instructions (IO), and Generic Instructions (G). Figures 2-6 through 2-9 show the bit format of each class. Bits 3 through 6 of the instruction word identify the type of instruction as shown in Table 2-1.

Memory Reference.—Figure 2-6 shows the format of memory reference instructions. Bit 1, the indirect bit, = 1 whenever an indirect memory reference is specified. Bit 2, the index bit, = 1 whenever indexing is specified as part of address formation. The operation code bits allow 14 different memory reference instructions to be specified. The other two configurations of these four bits escape to other format interpretations. (See Table 2-1.)

The sector bit = 0 if the reference is to the base sector and = 1 if the reference is to the sector in which the instruction is located. The 9-bit address field contains the displacement from zero in the sector specified.

Input/Output.—Figure 2-7 shows the word format for input/output instructions. The operation code specifies the desired function from the point of view of the computer. The function code further defines the desired function from the point of view of the device. The controller address specifies the I/O option which is to perform the function.

Shift.—The shift instruction word format is shown in Figure 2-8. The operation code specifies the type of shift and the direction. The shift count field is interpreted as a modulo 64 negative number in TWOs complement notation. Up to 64 shifts may be specified, although 33 is the maximum normally considered usable.

Generic.—Generic instructions have a 16-bit operation code without any modifying bits as shown in Figure 2-9.

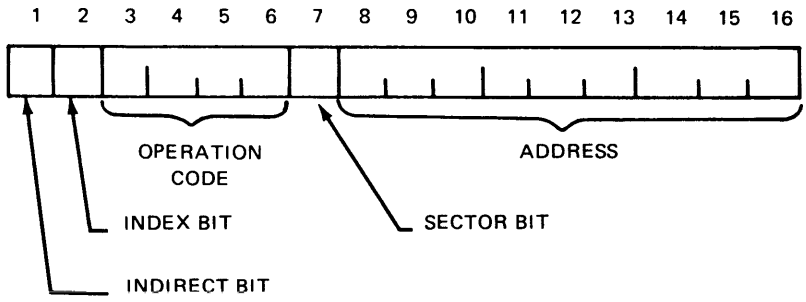


Figure 2-6. Memory Reference Instructions

TABLE 2-1. IDENTIFICATION OF INSTRUCTION WORDS

Bits 1-2	Bits 3-6	Type	Comments	Bits 1-2	Bits 3-6	Type	Comments
00	0000	G	Control Group	10	0000	G	Skip Group
	1100	IO	OCP		1100	IO	INA
	other	MR	Unindexed, Direct		other	MR	Unindexed, Indirect
01	0000	SH	Shift Group	11	0000	G	Arithmetic and Halfword Group
	1100	IO	SKS		1100	IO	OTA
	other	MR	Indexed, Direct		other	MR	Indexed, Indirect

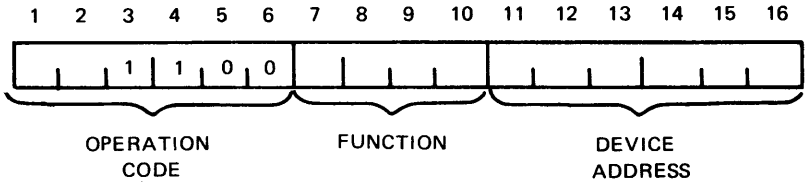


Figure 2-7. Input/Output Instruction

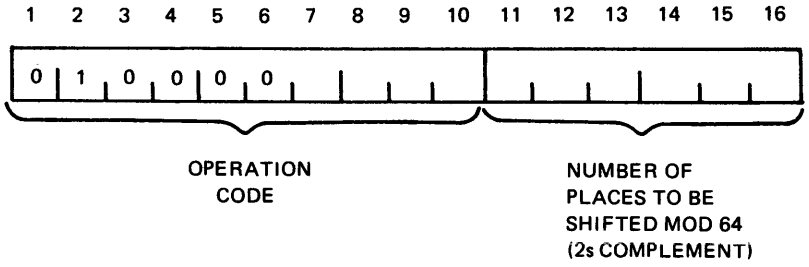


Figure 2-8. Shift Instruction

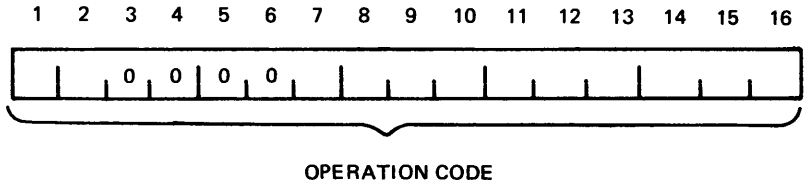


Figure 2-9. Generic Instruction

Indirect Address Words.—Indirect address words are pointers to a core location. They are used in conjunction with memory reference instructions to access locations outside of the two sectors that a memory reference instruction may access directly. They are also used whenever a pointer is convenient. Because of the two addressing modes of Honeywell Series-16 computers, the format of indirect address words will be discussed later.

SECTOR ORGANIZATION

The memory of the Honeywell 316/516 is divided into sectors of 512 words each (i.e., a 4096-word computer has eight sectors). Since 512 is 2^9, a 9-bit address can access any word in a sector ('000-'777). The address field of a memory reference instruction (bits 8 through 16) can thus directly address all locations in any one sector. The sector bit (bit 7) selects that sector from either a base sector or the sector in which the instruction is located (which could be the base sector). The base sector is sector 0 unless the Memory Lockout Option has been installed. In this case any sector may be specified as the base sector.

For example, an instruction in location '2100 (sector 2, word '100) with an address portion of '444 references '00444 if the sector bit = 0 (and sector 0 is the base sector) and '02444 if the sector bit = 1. Figure 2-10 shows the sector relationship in direct addressing.

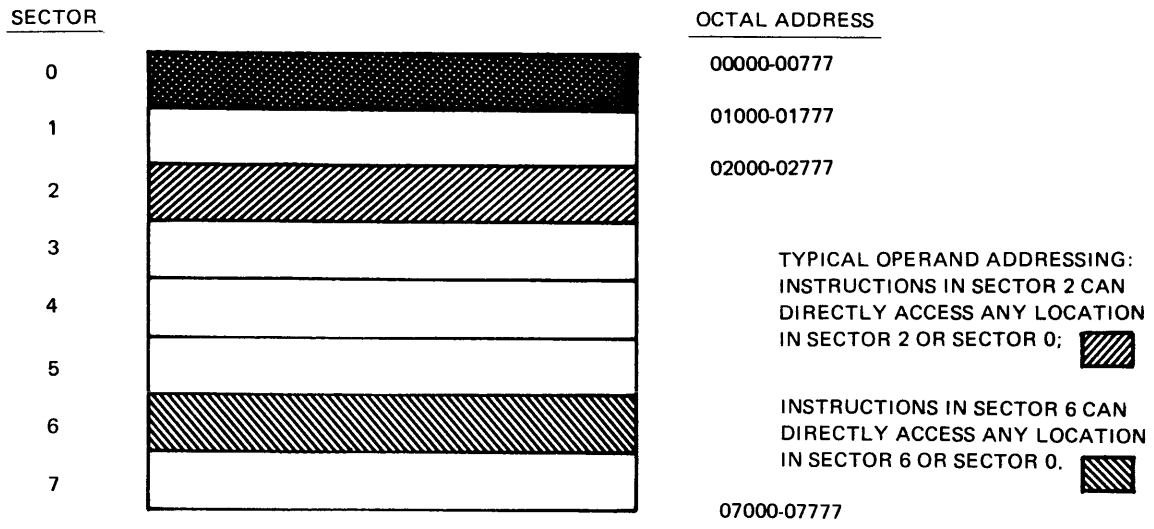


Figure 2-10. Memory Sectors in 4096-Word Honeywell 316/516

MEMORY ADDRESSING

Honeywell 316/516 computers have two distinct address formation modes: Normal Addressing and Extended Addressing. Extended Addressing is mandatory on computers with more than 16K of core, but is often installed on smaller computers. The Extended Addressing Option is discussed in detail in Section IV. For machines with Extended Addressing it is often desirable to write programs that operate properly in either mode of addressing.

Series 16 address constants contain 16 bits. In the normal address formation mode they are interpreted as 14 bits of address, sufficient to address 16K, an index (tag) bit, and an indirect (flag) bit. In the extended address formation mode they are interpreted as a 15-bit address, sufficient to address 32K, and an indirect bit.

The flow charts in Figures 2-11 and 2-13 depict the programmer-visible information about memory addressing presented in this section.

Normal Addressing

This section describes the techniques employed by the computer in forming an effective address for transmittal to the memory in Normal Addressing. Additional information about the impact of Base Sector Relocation on address is given in Section IV for systems with the Memory Lockout Option.

Addressing Summary.—Figure 2-11 shows the method of forming effective addresses in Honeywell 316/516 computers operating in the normal addressing mode. The changes in this diagram caused by the Memory Lockout option are described in Section IV.

Indirect Addressing.—When indirect addressing is specified by bit 1 of a memory reference instruction, the address specifies the location of an indirect address word rather than the location of the operand. Figure 2-12 shows the indirect address word format for normal addressing. Bit 1, the indirect bit, = 0 whenever this word points to the desired address. The indirect bit = 1 whenever this work points to another indirect address word. Continued indirect addressing is possible; each indirect reference adds a cycle to the execution time. When the Memory Lockout Option is installed, and the computer is operating in the RESTRICT mode, only eight levels of indirect addressing are allowed. (See Section IV.) Bit 2 is used as the index bit. It specifies that the contents of the index register are to be used at the present time in forming the effective address.

For example, a program operating in sector 2 which must load the contents of a location in sector 6 into the A-Register, would specify an Indirect Load A-Register instruction (LDA* in DAP-16), and place the sector 6 address in the indirect address word referenced by the address portion of the instruction (location '0444 or '2444 would be acceptable, to continue the previous example).

Indexing.—The index register is a 16-bit hardware register whose contents can be added to the direct address of an instruction as part of effective address formation. This action causes no increase in instruction execution time. Indexing is specified by a 1 in bit 2 of a memory reference instruction or an indirect address word.

If indexing is specified, the value in the index register is added algebraically to the direct address. The index register can contain either a positive or negative value, although negative values are used more commonly.

For example, if the index register contained -2 ('177776) and the previously mentioned memory reference instruction at '2100 (referencing '2444) were executed, the effective address formed would be '2444 +'177776 = '2442. Sector boundaries can be crossed with no increase in instruction execution time.

The index register can be loaded or stored directly by means of the load and store index register instructions LDX and STX. In addition, any instruction that addresses location 0 of the base sector addresses the index register. The usual method of incrementing the index register is the increment, replace, and skip instruction IRS 0.

Post-Indexing.—Indexing following indirect addressing may be specified by use of the index bit in an indirect address word.

Extended Addressing

This section describes the technique employed by the computer in forming an effective address for transmittal to the memory when Extended Addressing has been installed. Additional information about the impact of Base Sector Relocation on addressing is given in Section IV for systems with the Memory Lockout Option.

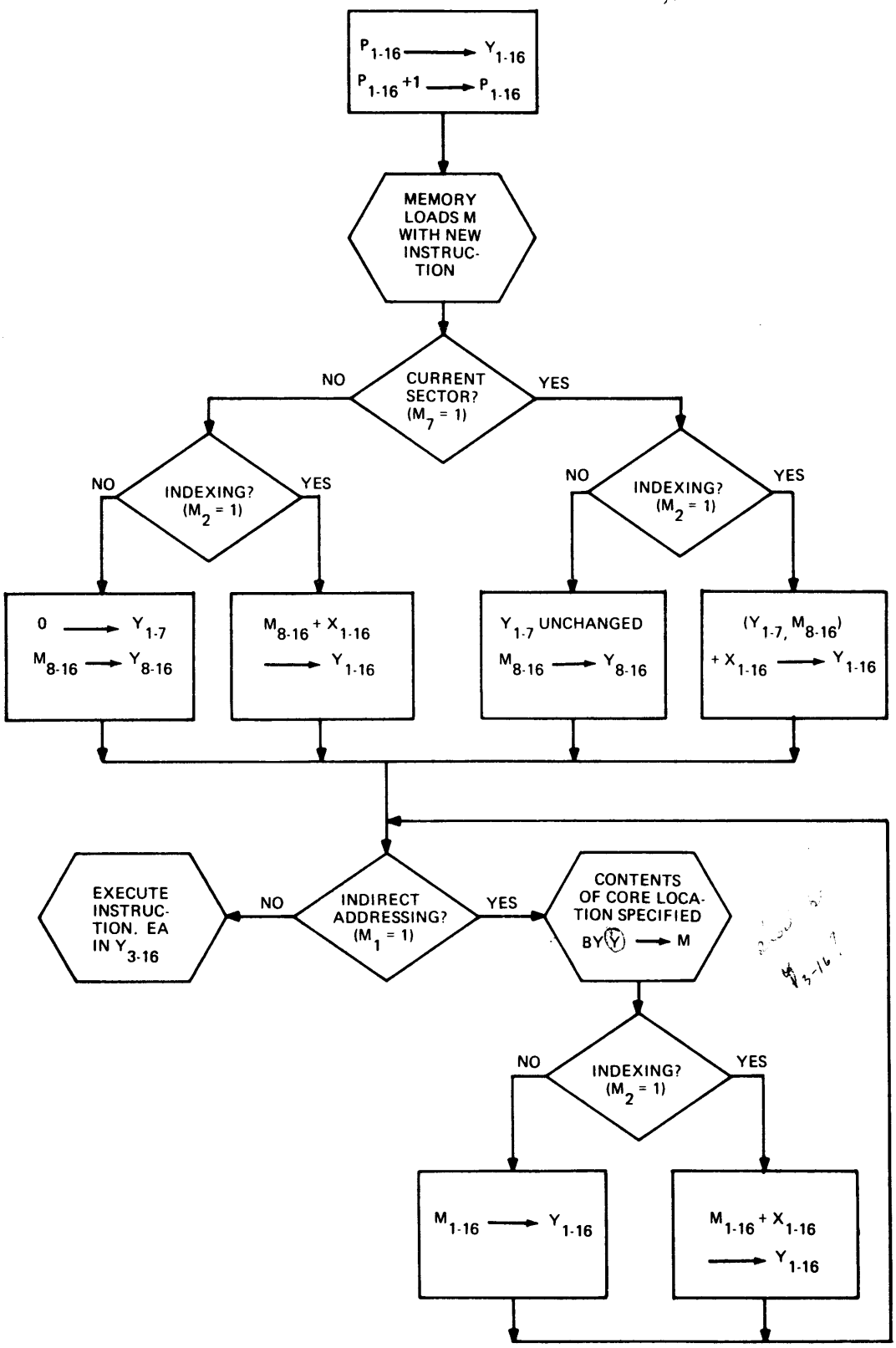


Figure 2-11. Effective Address Formation in the Honeywell 316/516 in Normal Addressing Mode

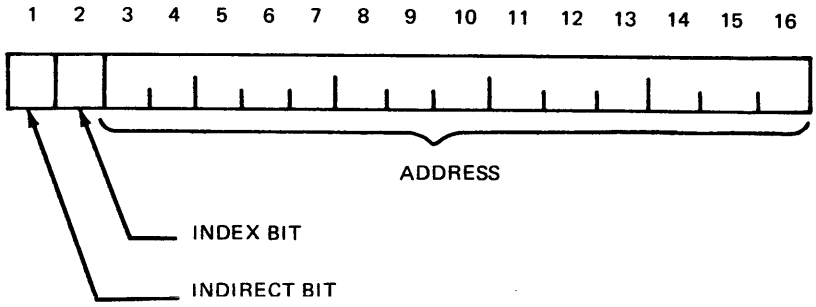


Figure 2-12. Indirect Address Word for Normal Addressing

Addressing Summary.—When extended addressing has been installed, addresses may be formed in either normal or EXTEND mode. Extended mode addressing is almost identical to normal addressing as shown in Figure 2-11. The only difference is that bit 2 of the Y-Register *always* specifies the bank in which the computer is currently operating – 0 for the first (lower) bank (16K) and 1 for the second (upper) bank (addresses above 16K). While operating in normal mode there is no way to alter bit 2 of the Y-Register. The net effect is that the computer seems to have 16K of memory. Unless relocated, the base sector is sector 0 when operating in the lower bank and sector '40 when operating in the upper bank.

Figure 2-13 shows the method of forming effective addresses when operating in the EXTEND mode. This mode may be entered by the generic instruction EXA or by the occurrence of an interrupt. The generic instruction DXA returns processing to the normal mode. The changes in Figure 2-13 caused by the Memory Lockout Option are described in Section IV.

Indirect Addressing.—When indirect addressing is specified by bit 1 of a memory reference instruction, the address specifies the location of an indirect address word rather than the location of the operand. (See Section II for a discussion of indirect addressing in the normal mode.) Figure 2-14 shows the indirect address word-format for extended addressing. Bit 1, the indirect bit, = 0 whenever this word points to the desired address. The indirect bit = 1 whenever this word points to another indirect address word. Continued indirect addressing is possible; each indirect reference adds a cycle to the execution time. When the Memory Lockout Option is also installed and the computer is operating in the RESTRICT mode only eight levels of indirect addressing are allowed. (See Section IV.)

For example, a program operating in sector '42 which must load the contents of a location in sector 6 into the A-Register, might specify an Indirect Load A-Register instruction (LDA* in DAP-16), and place the sector 6 address in the indirect address word referenced by the address portion of the instruction (location '00444 or '42444 would be acceptable, to continue the previous example).

Indexing.—The index register is a 16-bit hardware register whose contents can be added to the direct address of an instruction as part of effective address formation. This action causes no increase in instruction execution time. Indexing is specified by a 1 in bit 2 of a memory reference instruction.

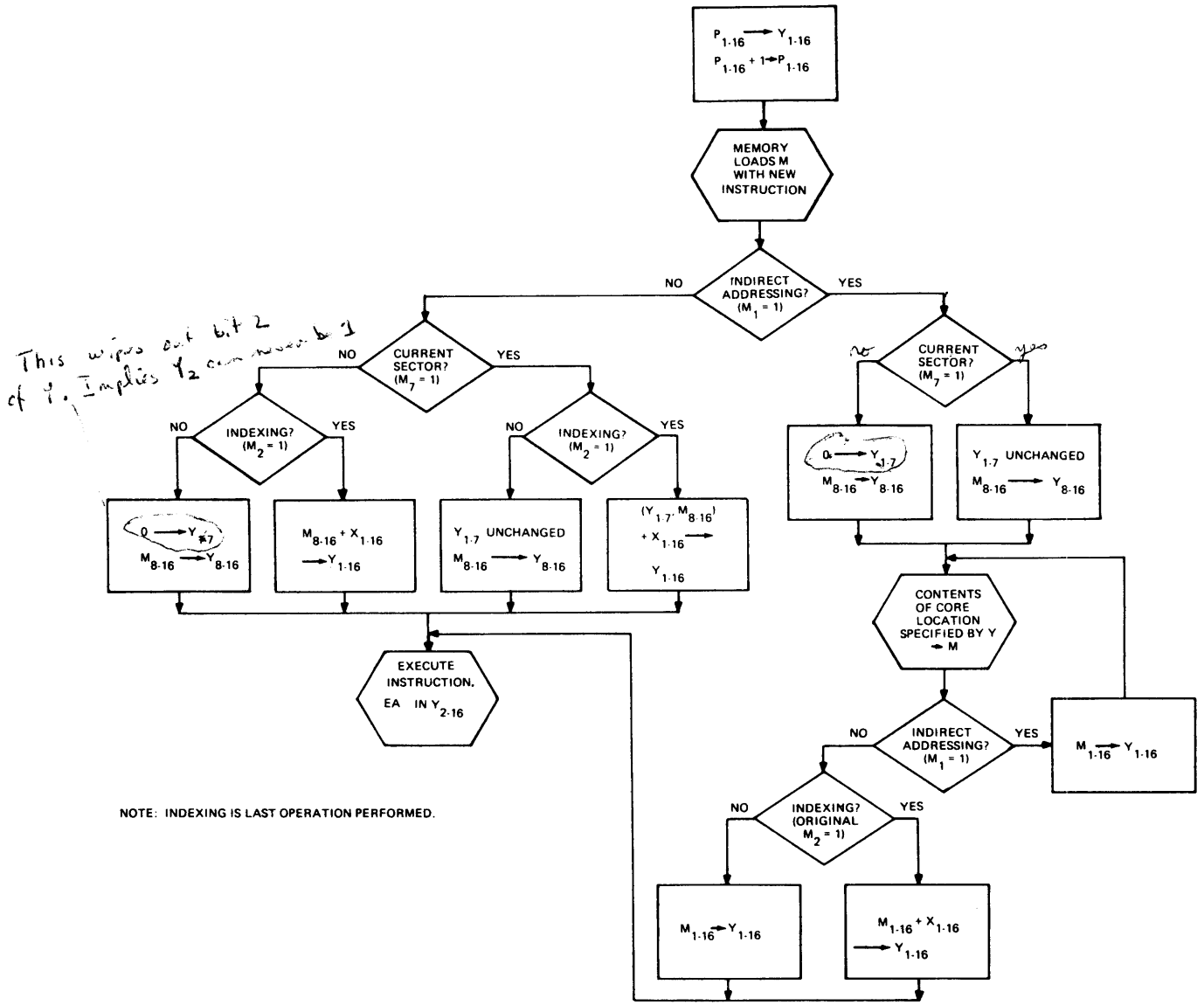


Figure 2-13. Address Formation in EXTEND Mode

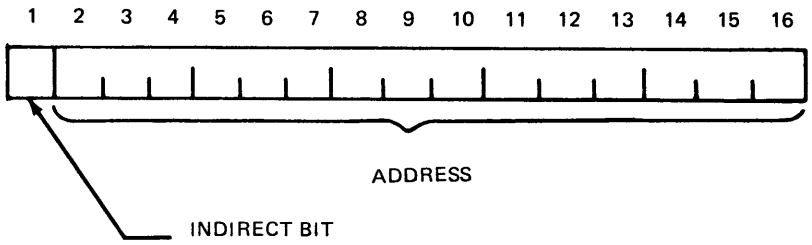


Figure 2-14. Indirect Address Word Format in Extended Addressing

If indexing is specified, the value in the index register is added algebraically to the direct address or the final indirect address. Thus only one level of indexing, a post-index, may be specified in an instruction. The index register can contain either a positive or negative value, although negative values are used more commonly.

For example, if the index register contained -2 ('177776) and the previously mentioned memory reference instruction at '42100 (referencing '42444) were executed, the effective address formed would be at '42444 + '177776 = '42442. Sector boundaries can be crossed with no increase in instruction execution time.

The index register can be loaded or stored directly by means of the load and store index register instructions LDX and STX. In addition, any instruction that addresses memory location 0 of the base sector addresses the index register. The usual method of incrementing the index register is the increment, replace, and skip instruction IRS 0.

INSTRUCTION SEQUENCES

Programs are executed sequentially with the program counter (P-Register) being incremented by 1 during each instruction. Certain instructions (skips, compare, some IO) conditionally increment the program counter by an additional 1 or 2, causing a skip. Others (jump, jump and store) unconditionally load the program counter with the effective address, thereby starting execution of another sequence.

C BIT

Table 2-2 lists all instructions which may change the state of the C-Bit and the necessary conditions. All other instructions have no effect on the C-Bit. See Section III for details on each instruction.

The following discussion illustrates TWOs complement arithmetic and the effect of addition instructions on the C-Bit.

TABLE 2-2. INSTRUCTIONS WHICH AFFECT THE C-BIT

Instruction	Reset Conditions	Set Conditions
ADD, DAD	Addition without overflow[1]	Addition with overflow[1]
SUB, DSB	Subtraction without overflow[2]	Subtraction with overflow[2]
AOA, ACA	Any but '077777+1	'077777+1
DIV	Divisor > Dividend[3]	Divisor ≤ Dividend[3]
All shifts except NRM	The C-Bit assumes the value of the last bit shifted out.	
RCB	Always	Never
SCB	Never	Always
CSA, OTK	$(A_1) = 0$	$(A_1) = 1$

[1]Overflow in addition occurs when the sign of the addend and augend are the same and the sign of the sum is different from that of the addend and augend. Overflow is not possible if the sign of the addend and augend are different.

[2]Overflow in subtraction occurs when the sign of the minuend and subtrahend are different and the sign of the difference is not that of the minuend. Overflow is not possible if the sign of the minuend and subtrahend are the same.

[3]In division the divisor is a 16-bit fraction with binary point between bits 1 and 2, and the dividend is a 32-bit fraction with the binary point between bits 1 and 2, and bit 17 always equal to zero. The magnitudes must be compared on this basis. Notice that multiplication does not affect the C-Bit because the product of two numbers less than one is always less than one.

Consider the A-Register to have only two bits, bits 1 and 2, the number set is then:

−2	10
−1	11
0	00
+1	01

Notice that there is one more negative number than positive number. The results of all additions not involving zero are given in the following table:

Addend		*Augend*	*C Bit*	*Sum*
+1	+	+1	1	−2
+1	+	−1	0	0
+1	+	−2	0	−1
−1	+	−1	0	−2
−1	+	−2	1	+1
−2	+	−2	1	0

The above table applies if the addend and augend are reversed. For subtraction the subtrahend should be TWOs complemented and the rules of addition followed.

DEDICATED LOCATIONS

The first 100 memory locations in sector 0 are dedicated locations used by the computer for special purposes. Table 2-3 shows these locations.

TABLE 2-3. DEDICATED LOCATIONS

Octal Address	Assignment
00000	Index Register
00001-00017	Protected locations for Key-In Bootstrap Loader
00020-00057	DMC Channels 1 through 16 starting and Ending Addresses
00060	Power Failure Interrupt Link
00061	Real-Time Clock Increment Location
00062	Memory Lockout Violation Interrupt Link
00063	Standard Interrupt Link
00064-00143	Optional Priority Interrupt Links Numbers 1 through 48 (also used for Memory Increment Option)

Index Register

The index register and location zero of the base sector are indistinguishable to programs. Any operation that addresses location zero or the index register affects both of them. When the base sector is relocated (Memory Lockout Option 316/516-08 or Relocatable Base Sector Option 316-800), location zero of the relocated base sector becomes the memory cell which duplicates the index register. (See the descriptions of these options for further details.)

Key-In Loader

The contents of memory locations 00001 through 00017 may be altered only when the MA/SI/RUN switch is in the MA (Memory Access) position. These 15 locations are used for a bootstrap loader. Tapes which are loaded through this loader are known as self-loading. After the tape is mounted in the appropriate reader execution is started at location 00001. Figure 2-15 shows the code which must be manually loaded into these locations before the computer is first used.

```
                                    HIGH-
                                    SPEED     ASR
LOC.                                READER    READER    COMMENTS
----  --------------------          ------    ------    --------------------

   1  STA    '57                    010057    010057    SAVE A REGISTER
*                                                          FOR START LOCATION
   2  OCP    1/4                    030001    030004    START READER
   3  INA    '1001/'1004            131001    131004    INPUT FIRST FRAME
   4  JMP    *-1                    002003    002003    DELAY UNTIL READY
   5  SNZ                           101040    101040    IGNORE LEADER
   6  JMP    *-3                    002003    002003    LOOP IN LEADER
   7  STA    0                      010000    010000    STORE IN LOCATION 0
  10  INA    '1001/'1004            131001    131004    CLEAR AND INPUT FRAME
  11  JMP    *-1                    002010    002010    DELAY UNTIL READY
  12  LGL    8                      041470    041470    SHIFT TO PACK
  13  INA    1/4                    130001    130004    INPUT FRAME
  14  JMP    *-1                    002013    002013    DELAY UNTIL READY
  15  STA*   0                      110000    110000    INDIRECTLY STORE WORD
  16  IRS    0                      024000    024000    POINT AT NEXT WORD
  17  SZE                           100040    100040    TEST INPUT WORD
```

Figure 2-15. Key-In Loader for Paper Tape

The A-Register may be used to direct the locations in which the final bootstrap will operate. Zero in the A-Register causes the bootstrap to be loaded to the location specified on the tape.

Key-in loaders also exist for card, magnetic tape, disc and drum input.

DMC Starting and Ending Addresses

These locations will be further discussed under the Direct Multiplex Control (DMC) Option. A pair of locations (e.g., '20 and '21) are dedicated to the starting and ending addresses of each of the 16 channels.

Interrupt Links

The locations '00060 and '00062 to '00143 are used as pointers to interrupt-handling routines. These must be loaded by any program which expects interrupts.

Real-Time Clock

Location '00061 is incremented by the Real-Time Clock Option. (See Section IV for details.)

INTERRUPTS AND BREAKS

Memory Access Priority Structure

The various functions that the computer performs are executed in a priority sequence if two or more functions are trying to simultaneously access memory. Table 2-4 shows the relative priorities between the program and breaks and interrupts.

TABLE 2-4. HONEYWELL 316/516 COMPUTER ACCESS-TO-MEMORY PRIORITY STRUCTURE

Relative Priority Level	Option/Function
1	Direct Memory Access Break (DMA) (DDP-516-21)
2	Direct Multiplex Control Break (DMC) (H316-20, 21, DDP-516-20)
3	Power Failure Interrupt (PFI)
4	Real-Time Clock Break (Honeywell 316/516-12)
5	Memory Lockout Violation Interrupt (Honeywell 316/516-08)
6	Standard Interrupt
7	Memory Increment Break (Honeywell 316/516-26)
8	Priority Interrupt (Honeywell 316/516-25)
9	Central Processing Unit (CPU)

Breaks

Certain operations may occur between instructions or between cycles of instructions without effecting the contents of the program counter. When the operations are complete, the program resumes. These actions are called "breaks," and include such operations as DMA or DMC I/O cycles, incrementation of the real-time clock, and memory increment breaks.

Interrupts

Interrupts are breaks which change the order of instruction execution. When an interrupt occurs the next instruction is a computer-generated JST* (indirect jump and store) through a dedicated memory location. Consequently, it is the responsibility of any program which allows an

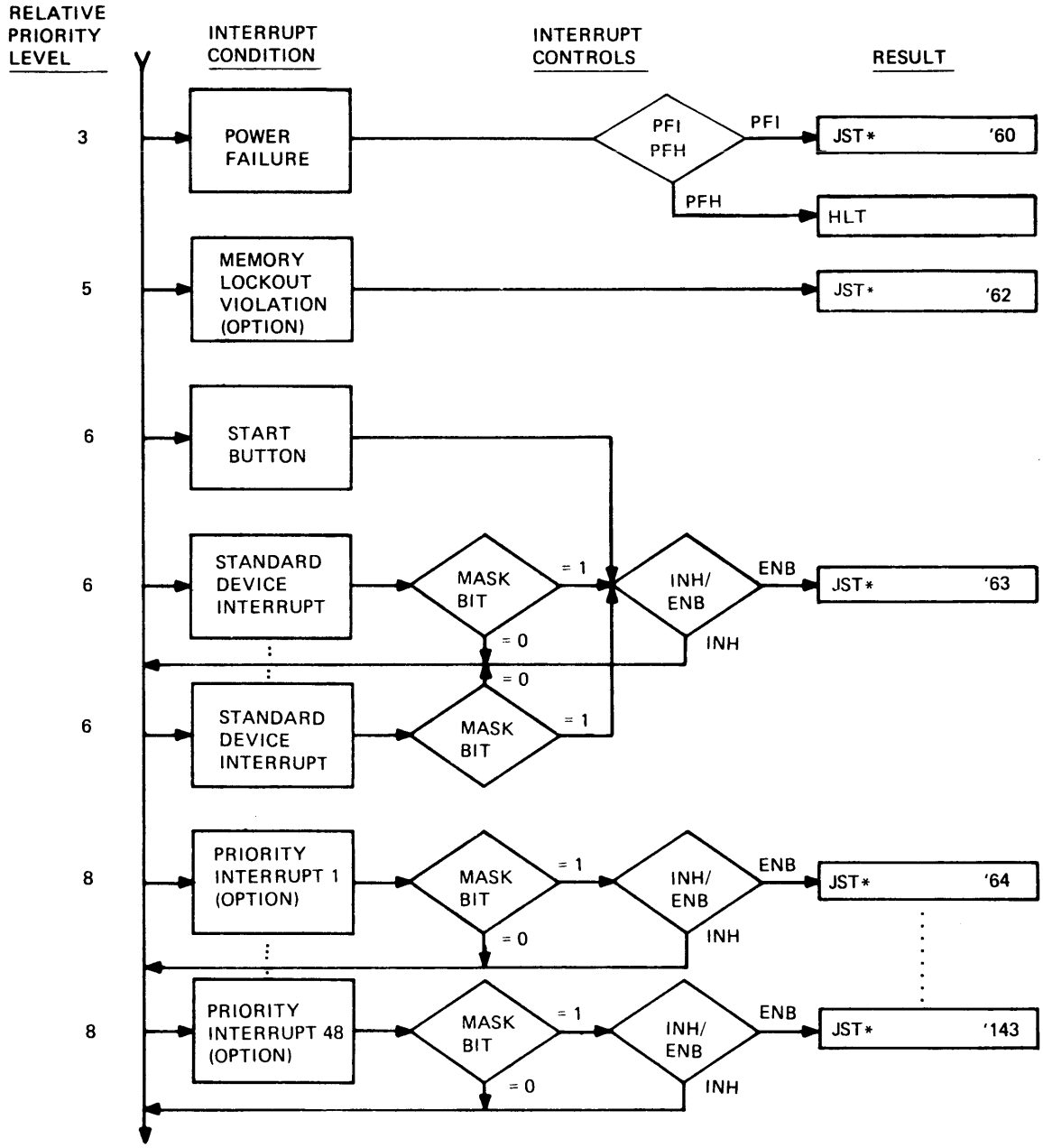
interrupt, to store a pointer to an interrupt-handling routine in the proper dedicated location. Interrupts also place the computer in the EXTEND mode if it is equipped with the Extended Addressing Option.

There are two controls over most interrupts. The primary control is exercised by the INH (inhibit interrupts) and ENB (enable interrupts) instructions. Interrupt requests are also controlled by mask bits which are set or reset by SMK (set interrupt mask) instructions. The mask bits allow an interrupt request to reach the CPU, but an interrupt will actually be generated only if the CPU is in the ENABLE mode due to ENB. Most interrupt requests except those caused by power failure, memory lockout violations, or the START button may be detected with SKS (skip on sense) instructions. MASTER CLEAR puts the computer in INHIBIT mode. All interrupts put the computer in INHIBIT mode automatically (preventing another interrupt from arriving immediately).

The START button interrupt is used to drive the computer to an interrupt routine in some executive systems. This interrupt can neither be tested nor masked.

Figure 2-16 shows interrupt handling. The Memory Increment and Real-Time Clock options are not shown since they are not interrupts, but breaks. The memory increment breaks are controlled by mask bits, but never generate an interrupt. The Real-Time Clock is controlled by specific input/output instructions and may generate a standard interrupt as may any input/output option.

Devices which normally generate standard interrupts may be wired to generate priority interrupts if the computer is appropriately configured.



NOTE: MASK BITS ALLOW INTERRUPT REQUESTS IN THE 1-STATE AND INHIBIT THEM IN THE 0 STATE. ALL INTERRUPTS PUT THE CPU IN THE EXTEND STATE FOR SYSTEMS WITH MORE THAN 16K OF MEMORY. ALL INTERRUPTS SET INH/ENB TO INH, PREVENTING ANOTHER INTERRUPT UNTIL THE FIRST IS HANDLED.

Figure 2-16. Interrupt Control

SECTION III
STANDARD INSTRUCTIONS

The standard instructions are described below and summarized in Tables 3-1 through 3-10. They are grouped for convenience into the following categories:

a. Load and Store
b. Arithmetic
c. Logical
d. Shift
e. Halfword
f. Program Control
g. Computer Control
h. Input/Output

LOAD AND STORE

Description of Instructions

*LDA 02 Load A-Register (MR, 2).**—The contents of the memory location at the effective address are loaded into the A-Register. The previous contents of the A-Register are lost.

STA 04 Store A-Register (MR, 2).—The contents of the A-Register are stored in the memory cell at the effective address. The previous contents of the memory cell are lost. The contents of the A-Register are unaltered.

LDX 15 With Index Bit = 1, Load X-Register (MR, 3).—The contents of the memory location at the effective address are loaded into the X-Register and base sector location 0. The previous contents of the X-Register and base sector location 0 are lost. This instruction cannot be indexed. However, if indirect addressing is called for, the indirect address can be indexed in the usual manner.

STX 15 With Index Bit = 0, Store X-Register (MR, 2).—The contents of the X-Register are stored in the memory cell at the effective address. The previous contents of the memory cell are lost. The contents of the X-Register are unchanged. The contents of base sector location 0 are not involved. This instruction cannot be indexed. However, if indirect indexing is called for, the indirect address can be indexed in the usual manner.

IMA 13 Interchange Memory and A-Register (MR, 3).—The contents of the A-Register and the memory cell at the effective address are interchanged. This instruction does not disturb the shift counter and may be used in interrupt processing.

CRA 140040 Clear A-Register (G, 1).—A computer-generated zero is loaded into the A-Register.

---

*In this and the following descriptions, the instruction mnemonics is shown first, followed by the octal instruction code and the name of the instruction. The type of instruction and execution time in cycles are shown in parentheses. For example, (MR, 2) means memory reference instruction, 2-cycle execution time.

TABLE 3-1. GLOSSARY OF SYMBOLS

Symbol	Definition
MR	Memory Reference Instruction
G	Generic Instruction
SH	Shift Instruction
IO	Input-Output Instruction
A	A-Register (16 bits)
B	B-Register (16 bits
C	C-bit (1 bit)
M	M-Register (16 bits)
P	Program Counter (16 bits)
X	Index Register (16 bits)
ADB	Address Bus
INB	Input Bus
OTB	Output Bus
EA	Effective operand address; the address from which the operand is obtained. This is determined only after all selection of sectors, indexing, and indirect addressing have been performed.
( )	Contents of a hardware register (e.g., (A) = contents of A-Register)
[ ]	Contents of core location specified (e.g., [EA] = contents of core location specified by EA)
n	Specified number of shifts to be performed
N	TWOs complement of the number of shifts to be performed.
→	Replaces
⇄	Is exchanged with
↴	Is discarded
Λ	Logical AND
V	Logical OR
∀	Exclusive OR
+	Algebraic Addition
-	Algebraic subtraction
DP Mode	Double Precision Mode associated with Honeywell 316/516-11
EXTMD	Extended Mode Indicator - associated with Extended Addressing (Honeywell 516-05, 06 and 316-06)
PMI	Previous Mode Indicator - associated with Extended Addressing (Honeywell 516-05, 06 and 316-06)
index bit	Index Bit (bit 2 of instruction word)

IAB 000201 Interchange A- and B-Registers (G, 1).—The contents of the A- and B-Registers are interchanged.

TABLE 3-2. LOAD AND STORE INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
LDA[1]	MR	02	Load A	[EA] → (A)	2	1.92	3.2
STA[1]	MR	04	Store A	(A) → [EA]	2	1.92	3.2
LDX	MR	15 (index bit = 1)	Load X	[EA] → (X) [EA] → [00000]	3	2.88	4.8
			NOTE This instruction cannot be indexed. However, if indirect addressing is called for, the indirect address can be indexed in the usual manner.				
STX	MR	15 (index bit = 0)	Store X	(X) → [EA]	2	1.92	3.2
			NOTE This instruction cannot be indexed. However, if indirect addressing is called for, the indirect address can be indexed in the usual manner.				
IMA	MR	13	Interchange Memory and A	(A) ⇄ [EA]	3	2.88	4.8
CRA	G	140040	Clear A	0 → (A)	1	0.96	1.6
IAB	G	000201	Interchange A and B	(A) ⇄ (B)	1	0.96	1.6

[1]These instructions may have alternate meanings when the High-Speed Arithmetic Unit Option is installed. (See Section IV for details.)

ARITHMETIC

Description of Instructions

ADD 06 Add (MR, 2).—The contents of the memory cell at the effective address are added to the contents of the A-Register and the sum left in the A-Register. See Table 2-2 for the effect of ADD on the C-Bit.

SUB 07 Subtract (MR, 2).—The contents of the memory cell at the effective address are subtracted from the contents of the A-Register and the difference left in the A-Register. See Table 2-2 for the effect of SUB on the C-Bit.

TCA 140407 TWOs Complement A-Register (G, 1.5).—The contents of the A-Register are replaced with the TWOs complement of the original contents.

ACA 141216 Add C-Bit to A-Register (G, 1).—The C-Bit is added algebraically to the A-Register. See Table 2-2 for the effect of ACA on the C-Bit after the addition.

AOA 141206 Add One to A-Register (G, 1).—A computer-generated '000001 is added algebraically to the A-Register. See Table 2-2 for the effect of AOA on the C-Bit.

TABLE 3-3. ARITHMETIC INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
ADD[1]	MR	06	Add	(A) + [EA] → (A) Overflow status → (C)	2	1.92	3.2
SUB[1]	MR	07	Subtract	(A) – [EA] → (A) Overflow status → (C)	2	1.92	3.2
TCA	G	140407	TWOs Complement A	– (A) → (A)	1.5	1.44	2.4
ACA	G	141216	Add C-Bit to A	(A) + (C) → (A) Overflow status → (C)	1	0.96	1.6
AOA	G	141206	Add one to A	(A) + 1 → (A) Overflow Status → (C)	1	0.96	1.6

[1]These instructions may have alternate meanings when the High-Speed Arithmetic Unit Option is installed. (See Section IV for details.)

LOGICAL

Description of Instructions

ANA 03 AND of Memory with A-Register (MR, 2).—The contents of the memory cell at the effective address are logically ANDed with the contents of the A-Register and the result left in the A-Register. The original contents of the memory cell are unchanged. The final value of any A-Register bit is 1 if and only if both the initial A-Register bit in this position and the corresponding bit in the memory location are 1s.

ERA 05 EXCLUSIVE OR of Memory with A-Register (MR, 2).—The contents of the memory cell at the effective address are logically EXCLUSIVE ORed with the contents of the A-Register and the result left in the A-Register. The original contents of the memory cell are unchanged. The value of any A-Register bit is unchanged if the corresponding bit in the memory cell is 0 and inverted if the corresponding bit in the memory cell is 1.

CMA 140401 Ones Complement A-Register (G, 1).—Each bit in the A-Register is logically inverted and the result left in A-Register.

SSP 140100 Set Sign Plus (G, 1).—Bit 1 of the A-Register is cleared to 0.

SSM 140500 Set Sign Minus (G, 1).— Bit 1 of the A-Register is set to 1.

CHS 140024 Complement Sign (G, 1).—Bit 1 of the A-Register is logically inverted.

CSA 140320 Copy Sign and Set Sign Plus (G, 1).—The value of bit 1 of the A-Register is copied into the C-Bit and bit 1 of the A-Register is cleared to 0.

TABLE 3-4. LOGICAL INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
ANA	MR	03	AND Memory With A	(A) $\wedge$ [EA] → (A) 0 0 0 0 1 0 1 0 0 1 1 1	2	1.92	3.2
ERA	MR	05	EXCLUSIVE OR Memory With A	(A) $\veebar$ [EA] → (A) 0 0 0 0 1 1 1 0 1 1 1 0	2	1.92	3.2
CMA	G	140401	Ones Complement A	$(\overline{A})$ → (A)	1	0.96	1.6
SSP	G	140100	Set Sign Plus	0 → $(A)_1$	1	0.96	1.6
SSM	G	140500	Set Sign Minus	1 → $(A)_1$	1	0.96	1.6
CHS	G	140024	Complement Sign	$(\overline{A})_1$ → $(A)_1$	1	0.96	1.6
CSA	G	140320	Copy Sign and Set Sign Plus	$(A)_1$ → (C) 0 → $(A)_1$	1	0.96	1.6

SHIFT

Description of Instructions

Logical Shifts.—The following four instructions shift the contents of the A or A- and B-Registers without regard to the sign bits. Zeros are supplied to the vacated bits of the word or doubleword. Bits shifted out are lost except that the last bit shifted is saved in the C-Bit. The previous value of the C-Bit is lost. The number of bit positions shifted is represented by n.

LGR 0404 Logical Right Shift (SH, 1 + n/2)

LRL 0400 Long Logical Right Shift (SH, 1 + n/2)

LGL 0414 Logical Left Shift (SH, 1 + n/2)

LLL 0410 Long Logical Left Shift (SH, 1 + n/2)

Arithmetic Shifts.—The following four instructions shift the contents of the A or A- and B-Registers keeping track of the sign bit. On right shifts bits equal to the sign bit are supplied to the vacated high-order bits. Bits shifted out are lost except the last bit is saved in the C-Bit. The previous value of the C-Bit is lost. On left shifts the C-Bit is 0 unless the sign changes value during the shift. Bit 1 of the B-Register does not participate in arithmetic shifts. The number of bit-positions shifted is represented by n.

ARS 0405 Arithmetic Right Shift (SH, 1 + n/2)

LRS 0401 Long Arithmetic Right Shift (SH, 1 + n/2)

ALS 0415 Arithmetic Left Shift (SH, 1 + n/2)

LLS 0411 Long Arithmetic Left Shifts (SH, 1 + n/2)

Rotates. –The following four instructions rotate the contents of the A or A- and B-Registers without losing or gaining any bits. The last bit shifted around the end is left in the C-Bit. The number of bit positions shifted is represented by n.

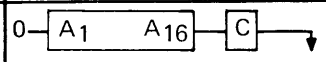
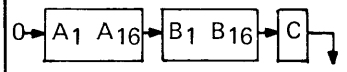
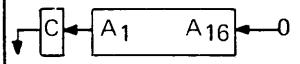
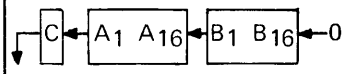
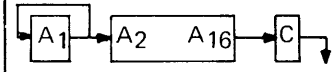
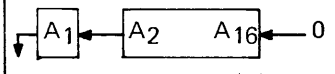
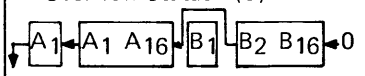
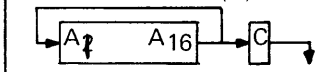
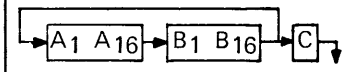
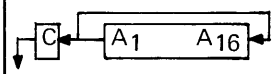
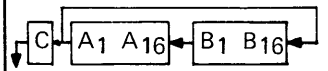
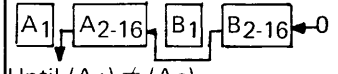
ARR 0406 Right Rotate (SH, 1 + n/2)

LRR 0402 Long Right Rotate (SH, 1 + n/2)

ALR 0416 Left Rotate (SH, 1 + n/2)

LLR 0412 Long Left Rotate (SH, 1 + n/2)

TABLE 3-5. SHIFT INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description
LGR	SH	0404	Logical Right Shift	
LRL	SH	0400	Long Logical Right Shift	
LGL	SH	0414	Logical Left Shift	
LLL	SH	0410	Long Logical Left Shift	
ARS	SH	0405	Arithmetic Right Shift	
LRS	SH	0401	Long Arithmetic Right Shift	
ALS	SH	0415	Arithmetic Left Shift	 Overflow Status → (C)
LLS	SH	0411	Long Arithmetic Left Shift	 Overflow Status → (C)
ARR	SH	0406	Right Rotate	
LRR	SH	0402	Long Right Rotate	
ALR	SH	0416	Left Rotate	
LLR	SH	0412	Long Left Rotate	
NRM[1]	G	000101	Normalize	 Until $(A_1) \neq (A_2)$ Number of shifts → (SC)

[1]Optional instruction. See High-Speed Arithmetic Unit Option.

HALFWORD

Description of Instructions

CAL 141050 Clear A-Register, Left Half (G, 1).—Bits 1 through 8 of the A-Register are cleared to 0. Bits 9 through 16 are unchanged.

CAR 141044 Clear A-Register, Right Half (G, 1).—Bits 9 through 16 of the A-Register are cleared to 0. Bits 1 through 8 are unchanged.

ICA 141340 Interchange Characters in A-Register (G, 1).—The contents of bits 1 through 8 of the A-Register are interchanged with the contents of bits 9 through 16. Bit 1 $\rightleftarrows$ bit 9, bit 2 $\rightleftarrows$ bit 10 etc.

ICL 141140 Interchange and Clear Left Half of A-Register (G, 1).—The contents of bits 1 through 8 of the A-Register are interchanged with the contents of bits 9 through 16. Bits 1 through 8 are then cleared to 0.

ICR 141240 Interchange and Clear Right Half of A-Register (G, 1).—The contents of bits 1 through 8 of the A-Register are interchanged with the contents of bits 9 through 16. Bits 9 through 16 are then cleared to 0.

TABLE 3-6. HALFWORD INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
CAL	G	141050	Clear A, Left Half	$0 \rightarrow (A_{1-8})$ (A_{9-16}) are unchanged	1	0.96	1.6
CAR	G	141044	Clear A, Right Half	$0 \rightarrow (A_{9-16})$ (A_{1-8}) are unchanged	1	0.96	1.6
ICA	G	141340	Interchange Characters in A	$(A_{1-8}) \rightleftarrows (A_{9-16})$	1	0.96	1.6
ICL	G	141140	Interchange and Clear Left Half of A	$(A_{1-8}) \rightarrow (A_{9-16})$ $0 \rightarrow (A_{1-8})$	1	0.96	1.6
ICR	G	141240	Interchange and Clear Right Half of A	$(A_{9-16}) \rightarrow (A_{1-8})$ $0 \rightarrow (A_{9-16})$	1	0.96	1.6

PROGRAM CONTROL

Description of Instructions

JMP 01 Unconditional Jump (MR, 1).—The next instruction to be executed is the instruction at the effective address. However, a jump-to-itself instruction (JMP *) is illegal.

JST 10 Jump and Store Location (MR, 3).—The address of the next sequential location is stored in bits 3-16 of the memory cell at the effective address. Bits 1 and 2 are unchanged (see the Extended Addressing Option for an exception). The next instruction executed is at the location following the effective address. The stored address is useful in subroutines accessing arguments and then returning in sequence.

CAS 11 Compare (MR, 3).—The contents of the A-Register are compared to the contents of the memory cell at the effective address. If (A) > [EA], the next sequential instruction is executed. If (A) = [EA], one instruction is skipped. If (A) < [EA], two instructions are skipped.

IRS 12 Increment, Replace, and Skip (MR, 3).—The contents of the memory cell at the effective address are incremented by 1 and the result returned to the same cell. If the result is 0, the next instruction is skipped. Otherwise the next instruction is executed.

Skips.—The following instructions all belong to the skip group of generic instructions. Note that the instructions are in pairs with bit 7 controlling the sense of the skip condition, and bits 8 through 16 defining the condition to be tested.

NOP 101000 No Operation (G, 1).—No operation is performed during the execution of this instruction.

SKP 100000 Unconditional Skip (G, 1).—The next instruction in sequence is skipped.

SZE 100040 Skip if A-Register Zero (G, 1)

SNZ 101040 Skip if A-Register Not Zero (G, 1)

SPL 100400 Skip if A-Register Positive or Zero (G, 1)

SMI 101400 Skip if A-Register Minus (G, 1)

SLZ 100100 Skip if A-Register Bit 16 = 0 (G, 1)

SLN 101100 Skip if A-Register Bit 16 = 1 (G, 1)

SRC 100001 Skip if C-Bit = 0 (G, 1)

SSC 101001 Skip if C-Bit = 1 (G, 1)

Sense Switch Skips.—The following instructions test the sense switches on the control panel and cause a skip if the indicated condition is met. Sense switches on the DDP-516 are reset in the horizontal position and set when tilted upward. Sense switches on the H-316 are reset when tilted upward and set when tilted downward.

SSR 100036 Skip if No Switch is Set (G, 1)

SSS 101036 Skip if Any Switch is Set (G, 1)

SR1 100020 Skip if Switch 1 is Reset (G, 1)

SR2 100010 Skip if Switch 2 is Reset (G, 1)

SR3 100004 Skip if Switch 3 is Reset (G, 1)

SR4 100002 Skip if Switch 4 is Reset (G, 1)

SS1 101020 Skip if Switch 1 is Set (G, 1)

SS2 101010 Skip if Switch 2 is Set (G, 1)

SS3 101004 Skip if Switch 3 is Set (G, 1)

SS4 101002 Skip if Switch 4 is Set (G, 1)

Load C Bit.—The following two instructions control the value of the C-Bit independently.

RCB 140200 Reset C-Bit (G, 1).—The C-Bit will be set to 0.

SCB 140600 Set C-Bit (G, 1).—The C-Bit will be set to 1.

TABLE 3-7. PROGRAM CONTROL INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
JMP	MR	01	Unconditional Jump	EA → (P) Next instruction to be executed is at location EA JMP * is illegal	1	0.96	1.6
JST	MR	10	Jump and Store Location	$(P_{3-16}) \rightarrow [EA_{3-16}]$ $[EA_{1,2}]$ unchanged EA + 1 → (P)	3	2.88	4.8

CAS	MR	11	Compare	Algebraically compare (A) and [EA]
				If (A) > [EA], execute next instruction.
				If (A) = [EA], skip next instruction.
				If (A) < [EA], skip two instructions
					3	2.88	4.8

IRS	MR	12	Increment, Replace, and Skip	[EA] + 1 → [EA] if [EA] + 1 = 0. Skip next instruction.	3	2.88	4.8
NOP	G	101000	No operation	Performs no operation Computer proceeds to next instruction	1	0.96	1.6
SKP	G	100000	Unconditional Skip	Skip next instruction	1	0.96	1.6
SZE	G	100040	Skip if A Zero	If (A) = 0: skip next instruction	1	0.96	1.6
SNZ	G	101040	Skip if A Not Zero	If (A) ≠ 0: skip next instruction	1	0.96	1.6

TABLE 3-7. PROGRAM CONTROL INSTRUCTION REPERTOIRE (Cont.)

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
SPL	G	100400	Skip if A Plus	If $(A_1) = 0$: skip next instruction	1	0.96	1.6
SMI	G	101400	Skip if A Minus	If $(A_1) = 1$: skip next instruction	1	0.96	1.6
SLZ	G	100100	Skip if A Register Bit 16 = 0	If $(A_{16}) = 0$: skip next instruction	1	0.96	1.6
SLN	G	101100	Skip if A Register Bit 16 = 1	If $(A_{16}) = 1$: skip next instruction	1	0.96	1.6
SRC	G	100001	Skip if C Bit = 0	If (C) = 0: skip next instruction	1	0.96	1.6
SSC	G	101001	Skip if C Bit = 1	If (C) = 1: skip next instruction	1	0.96	1.6
SSR	G	100036	Skip if No Sense Switch is Set	If no Sense Switches are ON: skip next instruction	1	0.96	1.6
SSS	G	101036	Skip if Any Sense Switch is Set	If any Sense Switch is ON: skip next instruction	1	0.96	1.6
SR1	G	100020	Skip if Sense Switch 1 is Reset	If Sense Switch 1 is OFF; skip next instruction	1	0.96	1.6
SR2	G	100010	Skip if Sense Switch 2 is Reset	If Sense Switch 2 is OFF: skip next instruction	1	0.96	1.6
SR3	G	100004	Skip if Sense Switch 3 is Reset	If Sense Switch 3 is OFF: skip next instruction	1	0.96	1.6
SR4	G	100002	Skip if Sense Switch 4 is Reset	If Sense Switch 4 is OFF: skip next instruction	1	0.96	1.6
SS1	G	101020	Skip if Sense Switch 1 is Set	If Sense Switch 1 is ON: skip next instruction	1	0.96	1.6
SS2	G	101010	Skip if Sense Switch 2 is Set	If Sense Switch 2 is ON: skip next instruction	1	0.96	1.6
SS3	G	101004	Skip if Sense Switch 3 is Set	If Sense Switch 3 is ON: skip next instruction	1	0.96	1.6
SS4	G	101002	Skip if Sense Switch 4 is Set	If Sense Switch 4 is ON: skip next instruction	1	0.96	1.6
RCB	G	140200	Reset C Bit	0 → (C)	1	0.96	1.6
SCB	G	140600	Set C Bit	1 → (C)	1	0.96	1.6

COMPUTER CONTROL

Description of Instructions

HLT 000000 Halt (G, 1.5).—All computer operations cease. The computer may be restarted only by depressing the START button. The instruction in the M-Register (except HLT) will be executed before any are fetched from memory.

ENB 000401 Enable Program Interrupt (G, 1).—The computer enables standard and priority interrupts. The enable does not become effective until the completion of the instruction following the ENB.

INH 001001 Inhibit Program Interrupt (G, 1).—The computer inhibits standard and priority interrupts from occurring. The inhibit status takes effect immediately. Interrupt requests are not disturbed.

INK 000043 Input Keys (G, 1).—The contents of the C-Bit are loaded into bit 1 of the A-Register; the contents of the Double Precision Mode Indicator (see the High-Speed Arithmetic Unit Option) are loaded into bit 2; the Previous Mode Indicator (see the Extended Addressing Option) is loaded into bit 3; bits 4 through 11 are cleared to zero; and the Shift Count is loaded into bits 12 through 16. The Shift Count is useful only for recovering the result of the optional normalize instruction NRM of the High-Speed Arithmetic Unit. The Double Precision Mode Indicator and Previous Mode Indicator will be loaded as 0 for CPUs without these options. This instruction does not change the present value of the C-Bit, the Double Precision Mode Indicator, the Previous Mode Indicator, or the Shift Count.

OTK 171020 Output Keys (I0, 2).—Bit 1 of the A-Register is loaded into the C-Bit; Bit 2 of the A-Register acts as an SGL or DBL instruction (see the High-Speed Arithmetic Unit Option); Bit 3 of the A-Register acts as an EXA or DXA instruction (see the Extended Addressing Option); and bits 12 through 16 of the A-Register are loaded into the Shift Counter. This instruction may also be called OTA '1020 or SMK '1020.

TABLE 3-8. COMPUTER CONTROL INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
HLT	G	000000	Halt	Sets machine to halt mode. No further instructions or interrupts are serviced until the console START button is pressed, at which time normal execution resumes.	1	0.96	1.6
ENB	G	000401	Enable Program Interrupt	Set machine status to permit interrupt. The permit interrupt status does not take effect until the instruction immediately following ENB is completed. (PI indicator lights.)	1	0.96	1.6

TABLE 3-8. COMPUTER CONTROL INSTRUCTION REPERTOIRE (Cont.)

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
INH	G	001001	Inhibit Program Interrupt	Resets "permit interrupt status" to prohibit standard or priority interrupts. (PI indicator is extinguished.)	1	0.96	1.6
INK	G	000043	Input Keys	(C) → $(A)_1$ (DP Mode) → $(A)_2$ (PMI) → $(A)_3$ 0 → $(A)_{4-11}$ Shift Count → $(A)_{12-16}$	1	0.96	1.6
OTK	G	171020	Output Keys	$(A)_1$ → (C) $(A)_2$ → (DP Mode) $(A)_3$ → (EXTMD) $(A)_{12-16}$ → Shift Count	2	1.92	1.6

INPUT/OUTPUT

Input and output operations are conducted with a series of busses and control lines collectively known as the I/O bus. This bus runs in daisy-chain fashion from the CPU to the control unit for each I/O device or pseudo-device in the I/O bus system.

The following is a brief description of the sequence of events occurring during any input/output instruction. Figure 3-1 shows how the instruction word is interpreted.

a. Bits 7 through 16 of any instruction word (including non-I/O instructions) are placed on the 10-bit Address Bus (ADB) of the I/O Bus. By convention bits 11 through 16 identify the device controller addressed by the instruction, and bits 7 through 10 (in conjunction with the signals on the control lines of the I/O bus) identify exactly the function of the addressed controller is to perform.

b. When the instruction is an I/O instruction the CPU identifies it to the controllers via signals on the I/O bus control lines.

c. The device controller addressed by bits 11 through 16 of the Address Bus and the CPU takes appropriate action. All other control units take no action. If a signal must be sent back from the control unit to the CPU (necessary for input, output, and status tests) the device ready line (DRLIN, one of the control lines of the I/O bus) is used. The CPU may send signals to the unit on either the Output Control Pulse Line (OCPLS) or the Reset Ready Line (RRLIN).

d. Addresses '20 and '24 are reserved for special CPU functions. For instance, the instruction SMK '0020 causes the control lines to signal a mask-setting operation to most of the standard device control units, even though none are specifically addressed by the Address Bus.

The proper device function and address codes and the mask bit assigned for each input/output device are listed in the appropriate Programmers Reference Manuals.

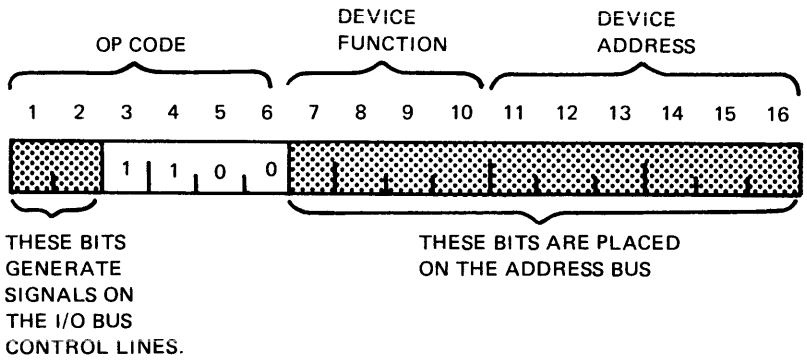


Figure 3-1. Input/Output Instruction Word Format

Description of Instructions

OCP 14 Output Control Pulse (I0, 2).—The central processor issues a control pulse which the controller uses to perform a function. Typically an OCP is used to turn a device on or off or select a particular operating mode. An OCP to set an *on* condition will normally clear the ready indicator in an input device controller and set the ready indicator in an output device controller. Interrupt requests may result from the latter. An OCP to set an *off* condition will normally clear the ready indicator regardless of its state and reset any interrupt request.

SKS 34 Skip on Sense Line (I0, 2).—This instruction is used to test the status of the addressed control unit or of the device it controls. The device function code identifies the particular test and the control unit forwards the result to the CPU via the ready line. Typical tests include whether the device is ready for data transfer, busy, or operational. An SKS is implied by an INA or an OTA, and the response determines the result of the INA or OTA.

INA 54 Input to A-Register I0, 2).—This instruction may have any of three results:

a. If the device is not ready the next instruction will be executed and no data will be input (i.e., the implied SKS does not skip).

b. If the device is ready and bit 7 of the instruction word is 0, the data on the Input Bus will be ORed with the present contents of the A-Register, and the next instruction will be skipped. Notice that some devices do not transmit a full 16 bits on input. The remaining bits will always be 0s.

c. If the device is ready and bit 7 of the instruction word is 1, the A-Register will be cleared and then the data on the Input Bus will be loaded into the A-Register. The next instruction will be skipped.

Most options have their ready indicators reset by the "success" of an INA (which is said to "fall through").

OTA 74 Output from A-Register I0, 2).—This instruction may have either of two results:

a. If the device is not ready the next instruction will be executed and no data will be output (i.e., the implied SKS does not skip).

b. If the device is ready the contents of the A-Register will be placed on the Output Bus and the next instruction will be skipped. The contents of the A-Register are not altered by an OTA. The device controller will receive the data from the output bus and take appropriate action. For devices which use fewer than 16 bits the excess bits will be ignored.

Most options have their ready indicators reset by the "success" of an OTA (which is said to "fall through").

SMK 74 Set Mask (10, 2).—SMK instructions are OTAs with device addresses of '20 or '24. Each possible SMK instruction loads the contents of the A-Register into one of 16 possible receiving registers, which may be up to 16 bits long. Some of these registers are made of smaller registers or individual bits distributed throughout the computer. Usually they are used for control purposes. The instruction following an SMK is never skipped.

The standard interrupt masks are set with SMK '0020. Table 3-9 lists the bit assignment for these masks. SMK '1020 is OTK (Output Keys). Other SMK instructions are used for setting masks associated with options.

TABLE 3-9. STANDARD INTERRUPT MASK ASSIGNMENT

Bit No.	Device	Bit No.	Device
1	Mag Tape Control Unit No. 1	9	Paper Tape Reader
2	Mag Tape Control Unit No. 2	10	Paper Tape Punch
3	Lab Display	11	ASR-33/35
4	Moving Head Disc File	12	Card Reader
5	I/O Channel No. 1	13	Card Reader/Punch
6	I/O Channel No. 2	14	Line Printer
7	I/O Channel No. 3	15	Memory Parity
8	Fixed Head Store	16	Real-Time Clock

TABLE 3-10. INPUT/OUTPUT INSTRUCTION REPERTOIRE

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
OCP	IO	14	Output Control Pulse	$(M)_{7-16} \rightarrow (ADB)_{7-16}$ GENERATE OCP CONTROL PULSE	2	1.92	3.2
SKS	IO	34	Skip on Sense Line	$(M)_{7-16} \rightarrow (ADB)_{7-16}$ SKIP CONDITION MET? NO: EXECUTE NEXT INSTRUCTION; YES: SKIP NEXT INSTRUCTION	2	1.92	3.2
INA	IO	54	Input to A	$(M)_{7-16} \rightarrow (ADB)_{7-16}$ CONTROLLER READY? NO: EXECUTE NEXT INSTRUCTION (M) = 1? 7 NO: (A) V (INB) → (A); YES: 0 → (A), INB → (A) SKIP NEXT INSTRUCTION	2	1.92	3.2
OTA	IO	74	Output from A	$(M)_{7-16} \rightarrow (ADB)_{7-16}$ CONTROLLER READY? NO: EXECUTE NEXT INSTRUCTION; YES: (A) → (OTB), SKIP NEXT INSTRUCTION	2	1.92	3.2
SMK	IO	74 (OTA with address '20 or '24)	Set Mask	The contents of the A-Register are loaded into a mask register.	2	1.92	3.2

SECTION IV
STANDARD MAINFRAME OPTIONS

This section describes the standard options of the DDP-516 and H316. These options greatly expand the power of the basic computer.

Figures 4-1 and 4-2 are snapshots showing all these options and their interface with the computer. The standard interrupt mask assignments and memory access priority structure which have been presented before are also included.

HIGH-SPEED ARITHMETIC UNIT OPTION (HONEYWELL 316/516-11)

This option provides hardware multiply and divide capability, hardware implementation of double-precision load, store, add, and subtract, and normalization capability.

Instructions which reference double precision operands must produce even effective addresses (after any indexing or indirect addressing). An odd effective address causes the instruction to be executed as if it had the next lower even effective address for double precision load, add, or subtract. An odd effective address in a double precision store causes the B-Register content to be stored in the specified location without affecting any other location. This requirement is automatically taken care of for direct, unindexed instructions when programs are written in DAP-16 and loaded with the linking loaders.

Added Instructions and Their Descriptions

The following instructions are added when the High-Speed Arithmetic Unit is installed. (See Table 4-1.)

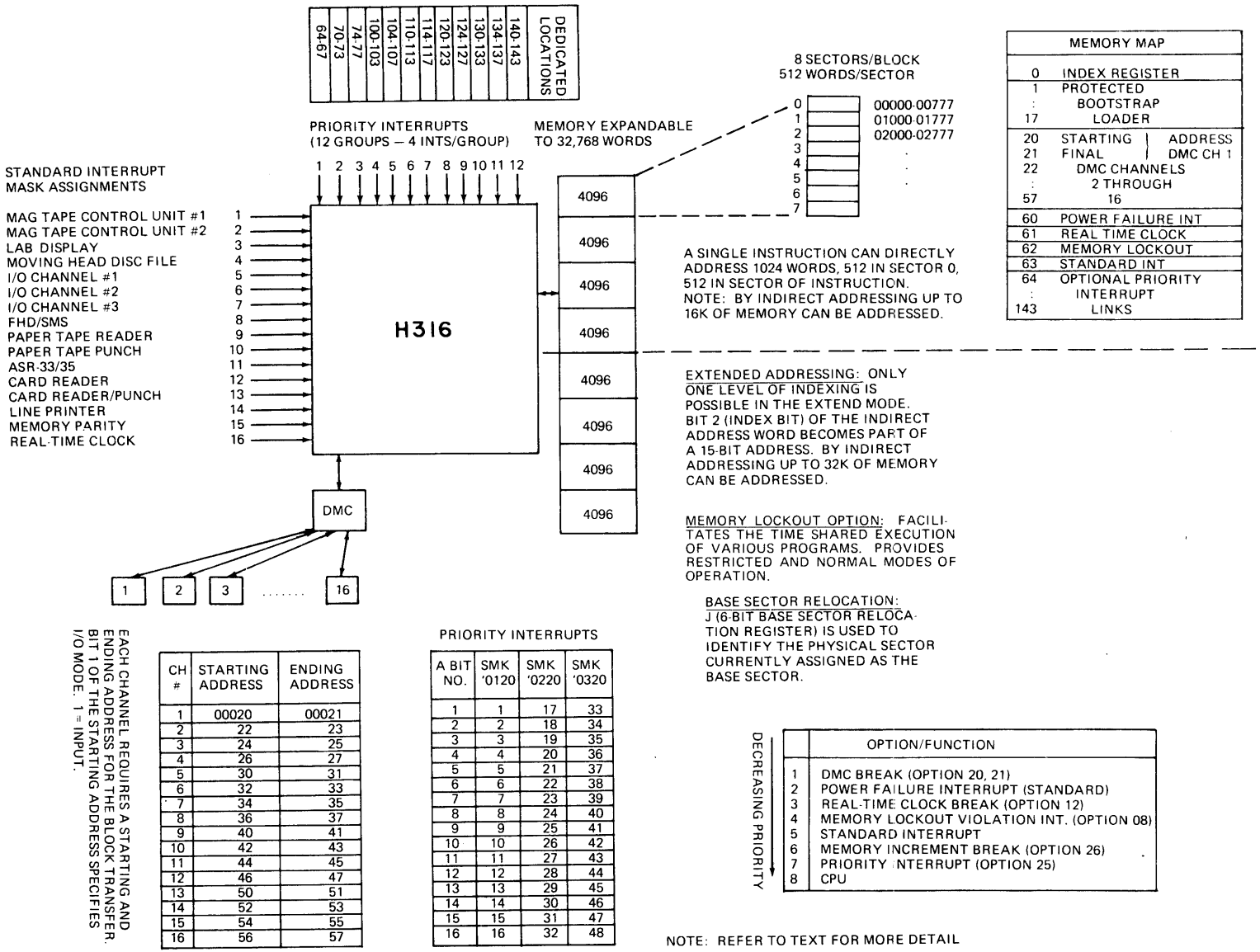
DBL 000007 Enter Double Precision Mode (G, 1).—LDA, STA, ADD, and SUB will be executed as DLD, DST, DAD, and DSB respectively until SGL (enter single-precision mode) is executed or the computer is MASTER CLEARed.

SGL 000005 Enter Single-Precision Mode (G, 1).—LDA, STA, ADD, and SUB will be executed in the normal fashion.

DLD 02 Double Precision Load (MR, 3).—The contents of the memory location at the effective address are loaded into the A-Register and the contents of the memory location at the effective address +1 are loaded into the B-Register. The previous contents of these registers are lost.

DST 04 Double Precision Store (MR, 3).—The contents of the A-Register are stored into the memory location at the effective address and the contents of the B-Register are stored into the memory location at the effective address +1. The previous contents of these memory locations are lost. The contents of the A- and B-Registers are unchanged.

DAD 06 Double Precision Add (MR, 3).—The contents of the memory cell at the effective address and at the effective address +1 (treated as a double precision number) are added to the contents of the A- and B-Registers (treated as a double precision number) and the sum is left in the A- and B-Registers. The C-Bit will be one if the addition resulted in overflow; otherwise it will be zero. An invalid sum will result if bit 1 of either the B-Register or the second memory cell was not 0 initially.



CH #	STARTING ADDRESS	ENDING ADDRESS
1	00020	00021
2	22	23
3	24	25
4	26	27
5	30	31
6	32	33
7	34	35
8	36	37
9	40	41
10	42	43
11	44	45
12	46	47
13	50	51
14	52	53
15	54	55
16	56	57

A BIT NO.	SMK '0120	SMK '0220	SMK '0320
1	1	17	33
2	2	18	34
3	3	19	35
4	4	20	36
5	5	21	37
6	6	22	38
7	7	23	39
8	8	24	40
9	9	25	41
10	10	26	42
11	11	27	43
12	12	28	44
13	13	29	45
14	14	30	46
15	15	31	47
16	16	32	48

Figure 4-1. H316 and its Standard Options

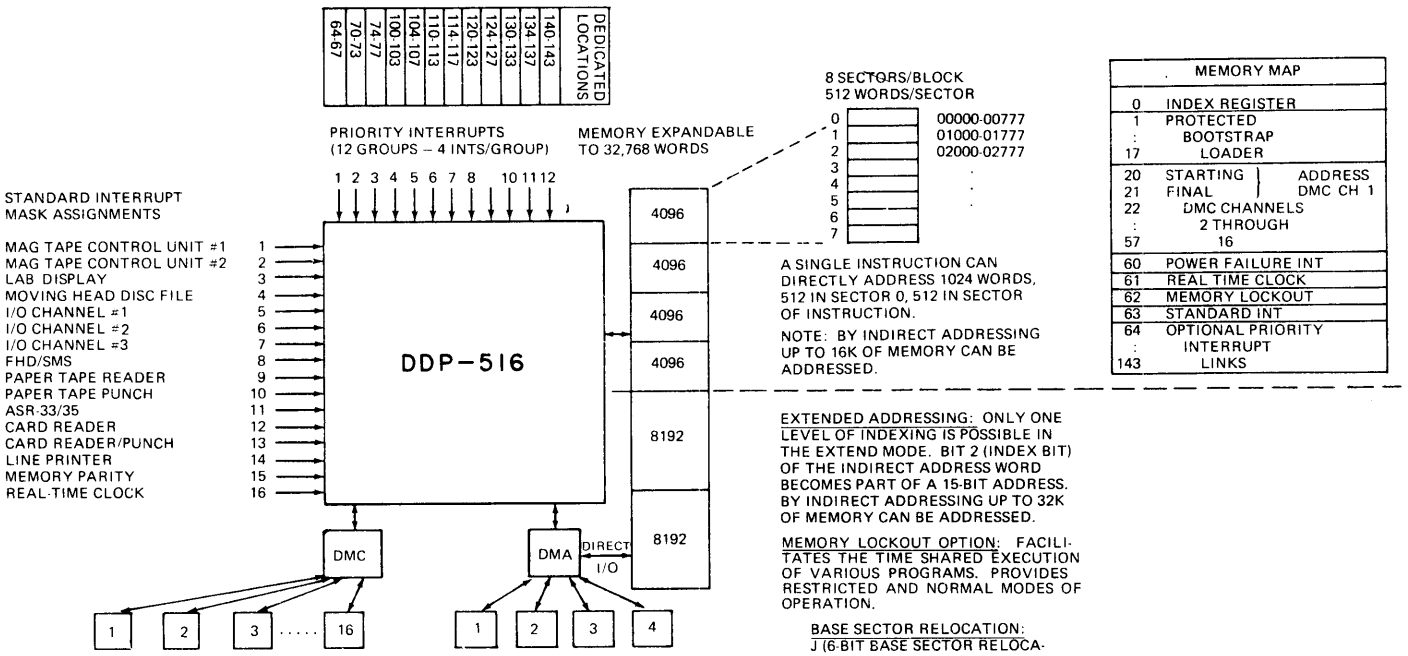
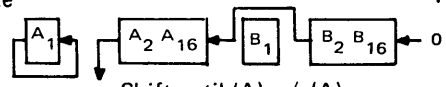


Figure 4-2. DDP-516 and its Standard Options

TABLE 4-1. HIGH-SPEED ARITHMETIC UNIT INSTRUCTIONS

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
DBL*	G	000007	Enter Double Precision Mode	Execute LDA, STA, ADD, and SUB as DLD, DST, DAD and DSB, respectively, until SGL is executed or MASTER CLEAR is depressed	1	0.96	1.6
SGL*	G	000005	Enter Single Precision Mode	Execute LDA, STA, ADD, and SUB in normal single precision	1	0.96	1.6
DLD	MR	02	Double Precision Load	[EA] → (A); [EA + 1] → (B)	3	2.88	4.8
DST	MR	04	Double Precision Store	(A) → [EA] : (B) → [EA + 1]	3	2.88	4.8
DAD	MR	06	Double Precision Add	(A, B) + [EA, EA + 1] → (A, B) Overflow Status → (C) If $[EA + 1]_1 \neq (B)_1$, an invalid sum results	3	2.88	4.8
DSB	MR	07	Double Precision Subtract	(A, B) - [EA, EA + 1] → (A, B) Overflow Status → (C) If $[EA + 1]_1 \neq (B)_1$, an invalid difference results	3 3	2.88 2.88	4.8 4.8
MPY	MR	16	Multiply	(A) x [EA] → (A, B)	5.5	5.28	8.8
DIV	MR	17	Divide	(A, B) ÷ [EA] → (A) Remainder → (B) Overflow Status → (C) If initial magnitude of dividend is ⩾ magnitude of divisor overflow occurs	10.0 or 10.5 or 11.0	9.60 or 10.08 or 10.56	16.0 or 16.8 or 17.6
NRM	G	000101	Normalize	 Shift until $(A)_2 \neq (A)_1$; number of shifts required stored as Shift Count	1 + n/2	0.96+ 0.48n	1.6+ 0.8n
SCA*	G	000041	Shift Count to A	Shift Count → $(A)_{11-16}$ 0 → $(A)_{1-10}$ The shift count is valid if no IAB, MPY, DIV, OTK, shift, or double precision instruction has been executed since the last NRM instruction was executed.	1	0.96	1.6

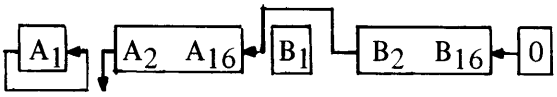
*See OTK, INK instructions Table 3-8

DSB 07 Double Precision Subtract (MR, 3).—The contents of the memory cell at the effective address and at the effective address + 1 (treated as a double precision number) are subtracted from the contents of the A- and B-Registers (treated as a double precision number) and the difference is left in the A- and B-Registers. The C-Bit will be 1 if the subtraction resulted in overflow; otherwise it will be 0. An invalid difference will result if bit 1 of either the B-Register or the second memory cell was not initially zero.

MPY 16 Multiply (MR, 5.5).—The contents of the memory cell at the effective address are multiplied by the contents of the A-Register and the result is left as a double precision number in the A- and B-Registers.

DIV 17 Divide (MR, 10, 10.5, or 11).—The contents of the memory cell at the effective address are divided into the contents of the A- and B-Registers (treated as a double precision number). The quotient is left in the A-Register and the remainder is left in the B-Register. If the initial magnitude of the dividend is greater than or equal to the initial magnitude of the divisor, overflow will result and the C-Bit will be 1; otherwise it will be 0.

NRM 000101 Normalize (G, 1 + n/2 where n is the number of shifts required).—The A- and B-Registers are shifted left as shown until bit A_1 and bit A_2 are not equal. The number of shifts required is stored as the shift count.



SCA 000041 Shift Count to A-Register (G, 1).—The shift count is loaded into the A-Register. This count is set by NRM (Normalize) instructions and remains valid until another NRM, any shift instruction, any double precision instruction, IAB, MPY, DIV, or OTK is executed. The shift count is a positive number.

Modified Instructions and Their Descriptions

The following instructions are altered when the High-Speed Arithmetic Unit is installed.

OTK 171020 Output Keys (IO,2).—Bit 2 of the A-Register acts as a DBL instruction if it is 1 and as an SGL instruction if it is 0. The shift count is input to the A-Register bits 11-16.

INK 000043 Input Keys (G, 1).—Bit 2 of the A-Register will be 1 if the CPU was in DBL mode and 0 if it was in SGL mode. Bits 11-16 are stored in the shift count.

MEMORY PARITY OPTION (316/516-07)

The memory parity option allows generation of a parity bit for each 16-bit memory word during memory write-cycles and the checking of this bit on memory read cycles (except console memory read cycles). A memory parity error indicator is set when a memory parity error is detected. This indicator may be tested and reset under program control. A light on the control panel shows the state of the memory parity indicator. MASTER CLEAR resets the indicator.

A memory parity error can cause an interrupt through the standard interrupt location '63. Bit 15 of the A-Register is used to mask interrupt requests on or off by the execution of SMK '0020. (See *Interrupts and Breaks.*)

Added Instructions and Their Descriptions

The following instructions (summarized in Table 4-2) are added when the memory parity option is installed.

TABLE 4-2. MEMORY PARITY INSTRUCTIONS

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
RMP	G	000021	Reset Memory Parity Error	0→ memory parity indicator	1	0.96	1.6
SPN	G	100200	Skip if No Memory Parity Error	If memory parity indicator = 0, skip next instruction	1	0.96	1.6
SPS	G	101200	Skip if Memory Parity Error	If memory parity indicator = 1, skip next instruction	1	0.96	1.6

RMP 000021 Reset Memory Parity Error (G, 1).—The memory parity indicator is reset.

SPN 100200 Skip if No Memory Parity Error (G, 1).—The next instruction will be skipped if the memory parity indicator is not set. Otherwise the next instruction will be executed.

SPS 101200 Skip if Memory Parity Error (G, 1).—The next instruction will be skipped if the memory parity indicator is set. Otherwise the next instruction will be executed.

Set Interrupt Mask (SMK '0020; bit 15 set).—The state of the standard interrupt mask is made equal to bit 15 of the A-Register (which must contain the entire mask word) by executing SMK '0020.

EXTENDED MEMORY AND EXTENDED ADDRESSING

DDP-516 or H316 systems with extended memory (more than 16K) require the Extended Addressing Option, which is also available without the added memory. Memory is divided into two banks, the first 16K and the remaining 4, 8, 12, or 16K. All operations within one bank are performed normally. A special mode EXTEND is available for moving from one bank to the other or for connecting the two banks together to permit addressing up to 32K. This mode is controlled by two added instructions and is encoded in bit 2 of the KEYS.

Operation

Figure 2-13 shows address formation in the EXTEND mode. Notice that bit 2 (used as the index bit in normal addressing) is used as an address bit. This allows only one level of indexing, controlled by the index bit of the instruction word and performed as the final step in effective address formation (post-indexing). Programmers are cautioned that the mode of operation has substantial implications at the program design and code stages. If the program may possibly be operated in EXTEND mode, it is strongly recommended that it be written to work in either mode. The handling of indexed instructions is the most important difference between the modes. Programmers should also be aware that recovery from interrupts is difficult for programs operating in EXTEND mode. If the following restrictions are followed, however, no difficulty will be encountered.

a. All EXA instructions should be followed immediately by a JMP instruction, with no memory reference instructions intervening. This restriction ensures that an interrupt immediately following the EXA cannot cause any memory reference instructions to be executed in the wrong mode.

b. The interrupt return routine should end with exactly the following sequence:

```
INH
LDA   <location of stored keys>
OTK
IMA   <location of stored A-Register>
ENB
JMP*  <return pointer>
```

Programs may operate in either bank in the normal mode in exactly the same fashion assuming that the program was loaded in the normal addressing mode. Care should be taken to ensure that the index register and location 0 (or '40000) are brought into agreement prior to use.

Added Instructions and Their Descriptions

The following two instructions (Table 4-3) are added when Extended Addressing is installed.

EXA 000013 Enable Extended Addressing (G, 1). The computer is placed in the EXTEND mode at the end of the execution of this instruction.

DXA 000011 Disable Extended Addressing (G, 1).—The computer prepares to be placed in the normal mode. The mode change does not become effective until the next JMP instruction is completed. This allows a return in EXTEND mode to a program in normal mode. Any number of instructions may be executed between DXA and the JMP.

TABLE 4-3. EXTENDED ADDRESSING INSTRUCTIONS

Mnemonic	Type	Op Code	Definition	Description	No. of Cycles	Time (μs)	
						DDP-516	H316
EXA	G	000013	Enable Extended Addressing	1 → EXTMD	1	0.96	1.6
DXA	G	000011	Disable Extended Addressing	0 → EXTMD	1	0.96	1.6
JST*	MR	10	Jump and Store Location	Modified to Store 15 bits When EXTMD = 1	3	2.88	4.8

*See INK and OTK Instructions Table 3-8.

Modified Instructions and Their Descriptions

The following instructions are altered when the Extended Addressing Option is installed.

OTK 171020 Output Keys (IO, 2).—Bit 3 of the A-Register acts as an EXA instruction if it is 1 and as a DXA instruction if it is 0.

INK 000043 Input Keys (G, 1).—An interrupt causes the addressing state to be stored in the Previous Mode Indicator, which is accessed by bit 3 during an INK. Bit 3 of the A-Register is 1 following an INK if the last interrupt occurred when the computer was in EXTEND mode and 0 if it was in normal mode.

JST 10 Jump and Store Location (MR, 3).—This instruction is altered in EXTEND mode to store bits 2 through 16. In normal mode it continues to store only bits 3 through 16. Note that subroutines in the upper bank which have been entered in EXTEND mode must not be reentered in normal mode without considering the effect of bit 2.

MEMORY LOCKOUT OPTION (316/516-08)

This option facilitates having more than one program in memory with lessened danger that they will interfere with each other. A program may be restricted to access only certain sectors, one of which (not necessarily sector zero) is the base sector for the program. An additional feature of the option is a mode of operation, RESTRICT mode, which enables unverified programs to run with minimum danger of interference with other programs. Use of this option implies an executuve or monitor program.

An address is stored as a result of the Memory Lockout violation. An analysis of the address often can lead to the cause of the violation. Refer to Debugging Memory Lockout Violations (page 4-12) for more information.

Protected Sector Selection

A lockout mask register (LMR) of 16, 32, 48, or 64 bits is used to identify the protected sectors of memory. Sectors which are protected (that is, may be read from but may not be written into) are identified with 0s in the mask register. Unprotected sectors are identified with 1s in the mask register. The contents of this register are under program control (SMK instructions). MASTER CLEAR protects all sectors, but does not put the computer in RESTRICT mode. Ones should be output with an SMK instruction to unprotect sectors. An attempt to write in a protected sector results in a memory lockout violation. Table 4-4 lists the bit assignments for the memory sectors.

TABLE 4-4. PROTECTED MEMORY RANGES

A-Register Bit	SMK '1420	SMK '1520	SMK '1620	SMK '1720
1	00000-00777	20000-20777	40000-40777	60000-60777
2	01000-01777	21000-21777	41000-41777	61000-61777
3	02000-02777	22000-22777	42000-42777	62000-62777
4	03000-03777	23000-23777	43000-43777	63000-63777
5	04000-04777	24000-24777	44000-44777	64000-64777
6	05000-05777	25000-25777	45000-45777	65000-65777
7	06000-06777	26000-26777	46000-46777	66000-66777
8	07000-07777	27000-27777	47000-47777	67000-67777
9	10000-10777	30000-30777	50000-50777	70000-70777
10	11000-11777	31000-31777	51000-51777	71000-71777
11	12000-12777	32000-32777	52000-52777	72000-72777
12	13000-13777	33000-33777	53000-53777	73000-73777
13	14000-14777	34000-34777	54000-54777	74000-74777
14	15000-15777	35000-35777	55000-55777	75000-75777
15	16000-16777	36000-36777	56000-56777	76000-76777
16	17000-17777	37000-37777	57000-57777	77000-77777

NOTE: Locations 00001-00017 are always protected against all programs, restricted or normal. However, no memory lockout violation interrupt occurs if an attempt is made to write in these locations unless sector 0 is protected and the machine is in the restricted mode.

Base Sector Relocation

The memory lockout option allows the base sector for the currently operating program to be relocated from physical sector 0 to any sector. An additional register, the J-Register, identifies the physical sector currently assigned as the base sector. This is a 6-bit register, corresponding to bits 2 through 7. It is set by the execution of an SMK '1320 instruction. This transfers bits 2 through 7 of the A-Register to the J-Register. The J-Register is cleared by any interrupt or MASTER CLEAR.

When the sector bit, bit 7 of a memory reference instruction, is a 1, the address (bits 8-16) is in the same sector as the instruction being performed. This is not a change from standard address formation. When the sector bit is a 0 the effective address is in the sector specified by the J-Register. If sector 0 is called for as a result of indirect addressing, it is relocated to the sector specified in the J-Register. If sector 0 is called for as a result of indexing, the access is made to physical 0 (i.e., relocation does not apply). Base sector relocation does not affect memory references caused by breaks or interrupts. The JST instruction continues to store a true 14- or 15-bit address without regard to the J-Register.

Location 0 in the physical sector which is the current base sector normally has contents identical to those of the X-Register. However, when the base sector is relocated the contents of location 0 in that physical sector and the X-Register will probably not agree. Any memory reference instruction which loads either the X-Register or location 0 will bring the two into step. This operation must also be considered when the base sector is returned to sector 0. The programmer must decide whether location 0 should be made to agree with the X-Register (accomplished by STX 0) or the reverse (LDX 0).

Address Formation

Figure 4-3 shows address formation with the memory lockout option. This figure closely resembles Figure 2-11 except for the handling of references to sector 0.

Figure 4-4 shows address formation with the memory lockout option when the computer is in EXTEND mode. This figure closely resembles Figure 2-13 except for the handling of references to sector 0.

RESTRICT Mode

There are two modes of operation available with the Memory Lockout Option. In normal (unrestricted) mode all operations proceed normally. The base sector may be relocated, but no memory sectors are protected.

The following restrictions are placed on a program operating in RESTRICT mode:

a. Instructions which write into memory locations (STA, DST, STX, LDX, IMA, IRS, and JST) are prevented from writing in protected sectors and cause a memory lockout violation if they attempt to do so.

b. Certain instructions are performed differently and cause a memory lockout violation. All input/output instructions (INA, OTA, OTK, OCP, SKS, and SMK) are treated as if they were the corresponding SKS. It is assumed that all I/O considerations are handled by a program in normal mode which is entered in response to interrupts. HLT is treated as if it were NOP, but causes a violation. INH is treated normally, but also causes a violation. (INH is therefore the conventional method used by a program to request a memory lockout violation interrupt.)

c. Indirect addressing is limited to eight levels after which a violation interrupt occurs.

The following sections describe the effect each of these instructions has when it causes a memory lockout violation. Notice that the first group (those which cause a violation because of attempting to write into a protected sector) is not altered in RESTRICT mode unless it causes a violation.

STA (Store A-Register) and DST (Double Precision Store A-Register).—Either of these instructions is treated as an NOP (No operation).

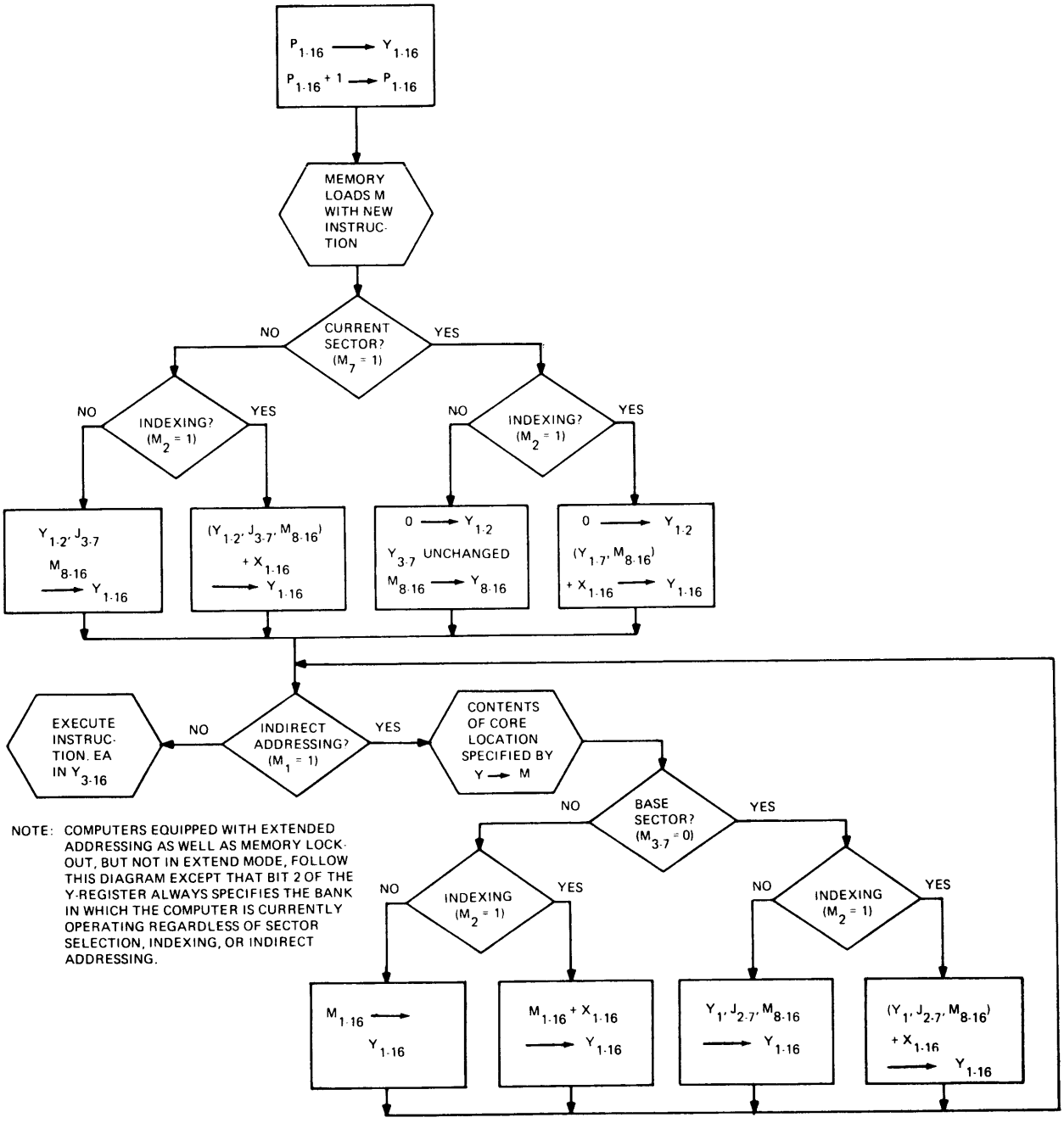


Figure 4-3. Address Formation in Computers Equipped With the Memory Lockout Option and Not in EXTEND Mode

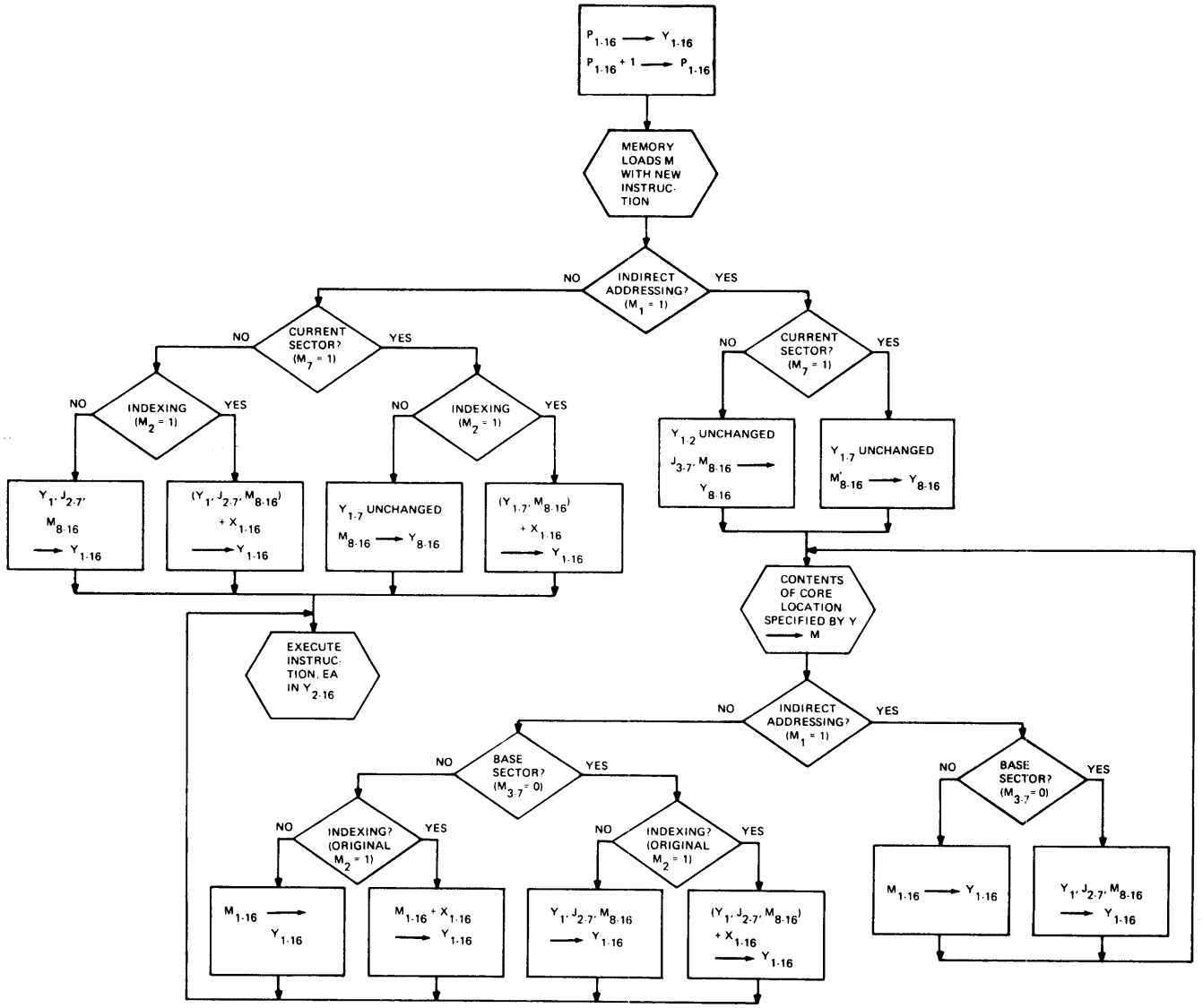


Figure 4-4. Address Formation in Computers Equipped With the Memory Lockout Option and Operating in EXTEND Mode.

STX (Store X-Register).—STX loads the X-Register with the contents of the addressed memory cell but does not change the contents of that cell or the contents of base sector location 0. Notice that an STX that attempts to store the X-Register in a protected location actually causes the contents of that location to be loaded into the X-Register.

LDX (Load X-Register).—LDX can cause a memory lockout violation only if the base sector is protected. In this case the contents of base sector location 0 are loaded into the X-Register, but no reference is made to the location that was to be loaded into the X-Register.

IMA (Interchange Memory and A-Register).—This instruction is treated as an LDA.

IRS (Increment, Replace, and Skip).—This instruction is treated as an IRS except that the incremented value is not returned to the protected location. This has the effect of an NOP unless the incremented value of the contents of the protected location equals 0. (See *Memory Lockout Violation Interrupt* for a further description of this case.)

JST (Jump and Store).—This instruction is treated as a JMP to the effective address + 1. That is, it is treated normally except that the return address is not stored.

INA, OTA, OTK, OCP, SKS, and SMK (Input/Output Instructions).—All of these instructions are treated as though they were SKS (Skip on Sense Line). If bits 7 through 16 of the instruction reference a condition that would cause an SKS to skip, the skip occurs. In addition, if the violating instruction was an INA and the resultant SKS causes a skip, the A-Register is loaded with all 1s. (See Memory Lockout Violation Interrupt.)

HLT (Halt).—This instruction is treated as an NOP.

INH (Inhibit Interrupts).—This instruction functions normally.

Memory Lockout Violation Interrupt

If any of the above instructions are attempted in RESTRICT mode, a Memory Lockout Violation Interrupt may occur through location '00062. Table 4-5 shows the effect of each violation instruction, when the memory lockout violation interrupt occurs, and what address is stored as a result of the interrupt. The violation returns the computer to the unrestricted mode.

Debugging Memory Lockout Violations

A programmer attempting to discover the source of a memory lockout violation should first check the four instructions preceding the address stored as a result of the violation. In most cases a violating instruction will be found there. If the address stored is in protected memory, the violating instruction may have been a JST located somewhere else. It is possible, in fact, that a JMP (which does not cause a violation) from the program being debugged to a second program was the original trouble and that the second program finally caused a violation by attempting a restricted instruction.

Added Instructions and Their Descriptions

The following instructions are added when the Memory Lockout Option is installed. Table 4-6 lists all the instructions which apply to the memory lockout option.

ERM 001401 Enter RESTRICT Mode (G, 1).—The computer is placed in RESTRICT mode immediately and interrupts are enabled at the end of the instruction following the ERM. RESTRICT mode continues until any program interrupt, either a standard, priority, or memory lockout violation interrupt occurs. The use of an INH is recommended to exit from RESTRICT mode.

TABLE 4-5. MEMORY LOCKOUT VIOLATIONS

Violating Instruction	Treated as	Can It Cause a Skip	When Will The Interrupt Occur	Address Stored If No Skip	Address Stored If Skip
STA	NOP	no	See Note 1 below		
DST	NOP	no	immediately	A+1	
STX	See text	no	See Note 1 below		
LDX	See text	no	See Note 1 below		
IMA	LDA	no	immediately	A+1	
IRS	See text	yes	immediately	A+1	A+2
JST	JMP to EA+1	no	immediately	EA+1	
INA	SKS (see text for possible input)	yes	immediately	A+1	A+2
OTA, OTK, OCP, SKS, or SMK	SKS	yes	immediately	A+1	A+2
HLT	NOP	no	See Note 1 below		
INH	INH	no	See Note 1 below		

A = the address of the violating instruction

EA = the effective address formed from the address field of the violating instruction

NOTE 1. For any of these instructions the following table determines when the interrupt occurs and the address that is stored. Any DMA or H316 DMC breaks are executed before this table goes into effect.

Next Clock Cycle The Interrupt Occurs After This Cycle or Instruction	Address Stored When no Standard or Priority Interrupt is Pending	Address Stored When a Standard or Priority Interrupt is Pending[2]
Real-Time Clock break, Memory Increment break, DMC break on DDP-516	A+1	A+1
JMP	EA	no memory lockout violation interrupt occurs
JST	EA+1	
INA, OTA, SKS, IRS, with no skip	A+2	
INA, OTA, SKS, IRS, and all skips, with skip	A+3	
CAS with [A] > (EA)	A+2	
CAS with [A] = (EA)	A+3	
CAS with [A] < (EA)	A+4	
Any instruction not listed above	A+2	A+2

A = the address of the violating instruction

EA = the effective address formed from the address field of the instruction following the violating instruction.

[A] = the contents of the A-Register

(EA) = the value stored at the effective address

NOTE 2. No interrupt can be pending if the violating instruction was an INH.

TABLE 4-6. MEMORY LOCKOUT INSTRUCTIONS

Mnemonic	Type	Instruction Word	Definition	Description	No. of Cycles	Time (μs) DDP-516	Time (μs) H316
ERM	G	001401	Enter Restricted Mode.	Enables program interrupt and puts computer in restricted mode operation. Restricted operation continues until any program interrupt occurs. Does not take effect until after the next instruction is completed.	1	0.96	1.6
SMK '1320	I/O	171320	Set Relocation Register.	$(A)_{2-7} \rightarrow (J)_{2-7}$ Defines physical location of all address references to base sector until another SMK '1320 is executed or MASTER CLEAR is activated, or any interrupt occurs.	2	1.92	3.2
SMK '1420	I/O	171420	Set Lockout Mask 1.	$(A)_{1-16} \rightarrow (LMR)_{1-16}$	2	1.92	3.2
SMK '1520	I/O	171520	Set Lockout Mask 2.	$(A)_{1-16} \rightarrow (LMR)_{17-32}$	2	1.92	3.2
SMK '1620	I/O	171620	Set Lockout Mask 3.	$(A)_{1-16} \rightarrow (LMR)_{33-48}$	2	1.92	3.2
SMK '1720	I/O	171720	Set Lockout Mask 4.	$(A)_{1-16} \rightarrow (LMR)_{49-64}$	2	1.92	3.2

Set Relocation Register (SMK '1320).—Bits 2 through 7 of the A-Register are loaded into the six bits of the J-Register, defining the physical sector that will serve as the base sector. MASTER CLEAR or any interrupt will reset this register to all 0s.

Set Lockout Mask Register (SMK '1420, SMK '1520, SMK '1620 and SMK '1720).—The contents of the A-Register are loaded into the proper 16 bits of the mask register in accordance with Table 4-6.

REAL-TIME CLOCK OPTION (HONEYWELL 316/516-12)

The Real-Time Clock (Table 4-7) automatically increments dedicated location '00061 at a constant rate. When this location is incremented from '177777 to '000000 an interrupt request is generated through the standard interrupt location '00063 if the mask bit is set.

TABLE 4-7. REAL-TIME CLOCK INSTRUCTIONS

OCP '0020	Start Real-Time Clock
OCP '0220	Stop Real-Time Clock
SKS '0020	Skip if Real-Time Clock not interrupting
SMK '0020 (bit 16 set)	Set interrupt mask

The clock frequency for the DDP-516 is one increment per cycle of the power source. This is a 16.7-ms period with 60-Hz power lines. The clock is locked to the power frequency and is therefore as accurate as this frequency.

The clock period for the H316 may be varied by hardware adjustment between 4 ms and 20 ms independent of the power line frequency. The stability is ±2%.

Programming the Real-Time Clock

Any program which desires real-time clock interrupts or interval timing will normally load location '00061 with the TWOs complement of the number of clock periods after which it wishes to be interrupted. The clock should then be started with OCP '0020, the mask bit set, and interrupts enabled.

Description of Instructions

Start Real-Time Clock (OCP '0020).—This instruction resets any real-time clock interrupt request present and allows the clock to start or continue incrementing location '00061. If the clock is being started the first break for incrementing location '00061 will occur at any time up to one clock period after the clock is started. If the clock was already going its rhythm will not be disturbed by this instruction.

Stop Real-Time Clock (OCP '0220).—This instruction resets the real-time clock interrupt request and stops the clock. If the clock is ready to interrupt while OCP '0220 is executing, that interrupt will occur following the instruction.

Skip if Real-Time Clock Not Requesting Interrupt (SKS '0020).—If the Real-Time Clock mask bit is set and location '00061 has incremented from '177777 to '00000 the real-time clock will request an interrupt and SKS '0020 will not skip.

Set Interrupt Mask (SMK '0020; bit 16 set).—The state of the standard interrupt mask is made equal to bit 16 of the A-Register (which must contain the entire mask word) by executing SMK '0020.

DIRECT MULTIPLEX CONTROL (DMC) OPTION

There are three versions of DMC available. Model 516-20 operates on the DDP-516. Model 316-20 is the standard DMC for the H316. Model 316-21 is the high-speed DMC for the H316. Separate sections describe the differences between the three models.

The DMC is used for data transfers between the memory and peripheral devices over the standard I/O bus. Each peripheral device using the DMC option must have a DMC subchannel in its control logic. This subchannel monitors DMC transfers on the I/O bus and sends requests for DMC cycles using one of the 16 DMC control channels (two of the 16 in the case of the Auto Switch Option). The DMC responds to requests for a cycle in priority order, channel 1 having the highest priority and channel 16 the lowest priority.

DMC Dedicated Locations

Each channel is controlled by the contents of two dedicated memory locations. Figure 4-5 shows the interpretation of these locations, often called the DMC starting and ending addresses. The starting address is actually the address of the next memory location to be accessed for the two standard DMCs. For the high-speed DMC it is the starting location, at least until the transfer is complete. Table 4-8 lists the dedicated locations for each channel.

A read/write flag has been shown as an option in the DMC ending location. This flag is not required except under special circumstances (to be described later) when using the high-speed DMC.

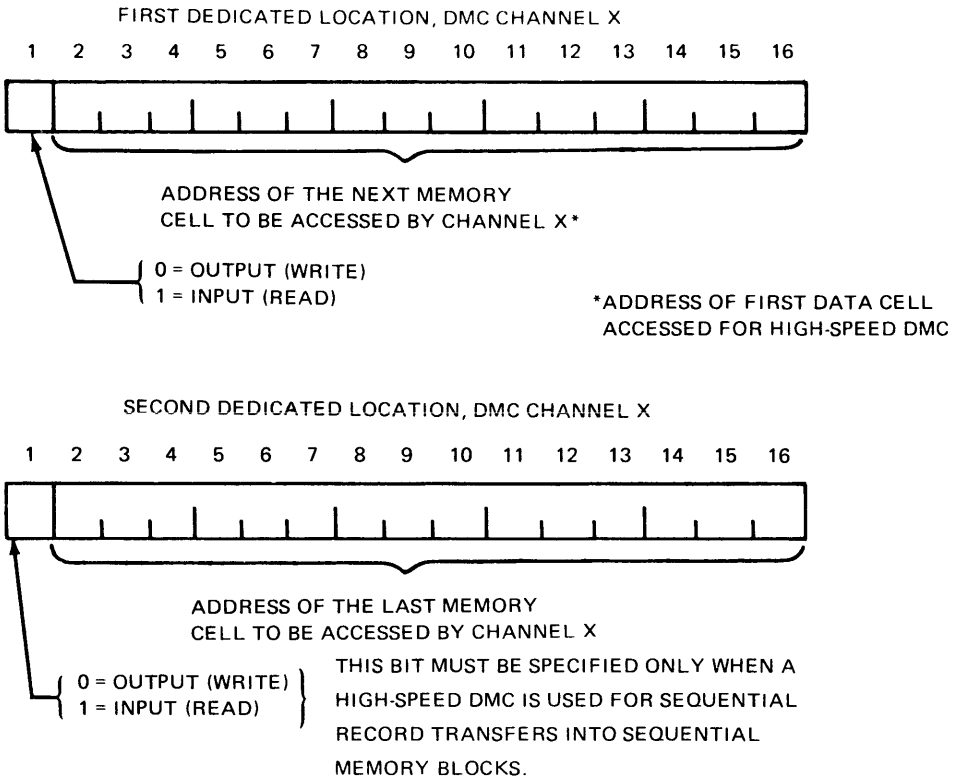


Figure 4-5. Interpretation of DMC Dedicated Locations

TABLE 4-8. DMC STARTING AND ENDING ADDRESSES

Channel Number	Starting Address	Ending Address
1	00020	00021
2	00022	00023
3	00024	00025
4	00026	00027
5	00030	00031
6	00032	00033
7	00034	00035
8	00036	00037
9	00040	00041
10	00042	00043
11	00044	00045
12	00046	00047
13	00050	00051
14	00052	00053
15	00054	00055
16	00056	00057

DMC Operation

The DMC services data requests by subchannels on a priority basis. It does not relinquish control to the CPU until all break requests that it has received are serviced.

When the DMC senses that the established range is exhausted it sends an end-of-range signal to the associated subchannel. The subchannel usually uses the end-of-range signal to initiate an interrupt request on behalf of the input/output device it services. The contents of the dedicated locations during a transfer and after the end-of-range are discussed separately for each type of DMC.

Auto Switch Option

Some subchannels may be equipped with an Auto Switch. This option allows the subchannel to be connected to two DMC channels and to switch between them automatically when end-of-range is reached. The channel switch is accomplished in less than one DMC cycle. This option allows long continuous transfers to or from shorter segments of core. It is available on the magnetic tape, and parallel input/output channels.

DDP-516 Standard DMC Timing

The DMC can effect a transfer following any instruction, providing that a request for a cycle is transmitted to the DMC 0.6 μs or more before the end of the instruction. A request occurring later than that will be serviced following the next instruction.

The data transfer is completed 1.74 μs into the DMC cycle for an input, and 3.0 μs for an output. The longest wait between a request and its execution can be calculated using the following formula:

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

where T_{wc} is the worst case waiting time in μs, T_{li} is the length of the longest instruction (which may be considered to be a 32-place shift, 16.32 μs), M is the number of higher priority DMC requests which may occur during T_{wc}, and N is the number of DMA (Direct Memory Access) requests, if any, which may occur during T_{wc}.

Each DMC cycle requires four memory cycles (3.84 μs) during which computation is suspended. If another DMC request is received 0.6 μs before the end of the DMC cycle or earlier another DMC cycle will follow immediately. Otherwise instruction execution will resume.

The maximum transfer rate of a single DMC channel (which allows no CPU operation) is one word every four cycles or 260 kHz.

DDP-516 Standard DMC Dedicated Locations

During the transfer the first dedicated location contains the address of the next location to be accessed. After the last piece of data has been transferred, the first dedicated location contains the address plus 1 of the last data word transferred.

H316 Standard DMC Timing

The DMC can effect a transfer following any instruction and following memory cycles inside many instructions, providing that a request for a cycle is transmitted to the DMC 1.0 μs or more before the next available break. A request occurring later than that will be serviced at the next possible break.

The data transfer is completed 4.635 μs into the DMC cycle for an input, and 5.685 μs for an output. The longest wait between a request and its execution can be calculated using the following formula:

$$T_{wc} = T_{li} + 6.4M + \begin{matrix} 4.635 \text{ (Input)} \\ 5.685 \text{ (Output)} \end{matrix}$$

where T_{wc} is the worst case waiting time in μs, T_{li} is the longest time during which an instruction may not be interrupted for a DMC break (4.8 μs plus 1.6 μs for each level of indirect addressing), and M is the number of higher priority DMC requests which may occur during T_{wc}.

Each DMC cycle requires four memory cycles (6.4 μs) during which computation is suspended. If another DMC request is received 1.0 μs or more before the end of the DMC cycle another DMC cycle will follow immediately. Otherwise instruction execution will resume.

The maximum transfer rate of a single DMC channel (which allows no CPU operation) is one word every four cycles or 156 kHz.

H316 Standard DMC Dedicated Locations

During the transfer the first dedicated location contains the address of the next location to be accessed. After the last piece of data has been transferred, the first dedicated location contains the address plus 1 of the last data word transferred.

H316 High-Speed DMC Timing

The DMC can effect a transfer following any instruction and following memory cycles inside many instructions, providing that a request for a cycle is transmitted to the DMC 1.0 μs or more before the next available break. A request occurring later than that will be serviced at the next possible break.

The data transfer is completed 4.635 μs into the first DMC cycle for input, and 5.685 μs for output. Data transfers for succeeding cycles are completed 2.635 μs into the cycle for input and 3.285 μs into the cycle for output. The longest wait between a request and its execution can be calculated using the following formula:

$$T_{wc} = T_{li} + 6.4M + 3.2N + \begin{matrix} 4.635 \text{ (first cycle, input)} \\ 2.635 \text{ (succeeding cycles, input)} \\ 5.685 \text{ (first cycle, output)} \\ 3.285 \text{ (succeeding cycles, output)} \end{matrix}$$

Where T_{wc} is the worst case waiting time in μs, T_{li} is the longest time during which an instruction may not be interrupted for a DMC break (4.8 μs plus 1.6 μs for each level of indirect addressing), M is the number of higher priority DMC requests requiring first cycle service which may occur during T_{wc}, and N is the number of higher priority DMC requests not requiring first cycle service which may occur during T_{wc}.

The first DMC cycle of a data transfer requires four memory cycles (6.4 μs) during which computation is suspended. All succeeding cycles require only two memory cycles (3.2 μs). If another DMC request is received 1.0 μs before the end of the DMC cycle or earlier another DMC cycle will follow immediately. Otherwise instruction execution will resume.

A subchannel may request that the starting and ending addresses be replaced in core at any time. Many peripherals do this following the last transfer. This adds an additional four memory cycles (6.4 μs) when performed at the end of the established range, or an additional two memory cycles (3.2 μs) when requested before the established range is exhausted. (See *H316 High-Speed DMC Dedicated Locations.*

The maximum transfer rate of a single DMC channel (which allows no CPU operation) is one word every two cycles or 312 kHz.

H316 High-Speed DMC Dedicated Locations

The contents of the dedicated locations are not incremented during the transfer as they are for the other DMCs. However, a subchannel may send a signal, known as ENDOP, to the DMC that causes it to load 1 plus the address of the last transferred word into the first dedicated location (starting address), giving the programmer the means to determine how many words were actually transferred. However, not all subchannels issue the ENDOP signal. The specifications of the subchannel should be checked in individual cases to see if this feature is included.

Normal end-of-range with subchannels that issue ENDOP leaves the last address plus 1 in the first dedicated location and the last address in the second dedicated location. The read/write flag has been shown as part of the second dedicated location in Figure 4-5 because this is the only way in which the flag in the first word can be saved when ENDOP is issued. This simplifies the programming required for reading from sequential records into sequential memory locations.

There is an awkward case which can sometimes arise. If no data transfers take place before the subchannel signals the end-of-operation, the last address plus 1 will be loaded into the first dedicated location.

DIRECT MEMORY ACCESS (DMA) OPTION (DDP-516)

The direct memory access option is designed for data transfers to and from the DDP-516 at high speed using a special data bus (not the standard I/O bus). DMA cycles are concluded in one memory cycle (0.96 μs). The DMA has first priority to memory and can break at the end of any memory cycle, whether or not it is the end of an instruction or DMC cycle. Each peripheral device using the DMA option must have a DMA subchannel in its control logic. This subchannel monitors transfers over the DMA bus and sends requests for DMA cycles using one of the four DMA control channels (two of the four in the case of the Auto Switch Option). The DMA responds to requests for a cycle in priority order, channel 1 having the highest priority and channel 4 the lowest priority.

DMA Address and Range

Each channel is controlled by the contents of two counters located in the DMA, the address and range counters. These counters are loaded by SMK instructions. The ranges may be read into the A-Register with INA instructions.

Address.—Bit 1 of the address specifies whether input or output transfers are desired. Zero means output; 1 means input. Bits 2 through 16 specify the next memory cell to be accessed. This counter is incremented during each DMA cycle.

Range.—The least significant 15 bits of the TWOs complement of the block size are stored in this counter by the program. This counter is also incremented during each DMA cycle. When it is incremented from '077777 to '100000 the DMA sends an end-of-range signal to the associated subchannel. The subchannel usually uses the end-of-range signal to initiate an interrupt request on behalf of the input/output device it services.

DMA Instructions

Table 4-9 lists the SMK and INA instructions used to load the address and range counters and read the range counters. (A) means the contents of the A-Register; (ACX) (X = 1 to 4) means the contents of the address counter; and (RCX)(X= 1 to 4) means the contents of the range counter.

TABLE 4-9. DIRECT MEMORY ACCESS INSTRUCTIONS

Mnemonic	Type	Instruction Code	Definition	Description	No. of Cycles	Time (μs)
SMK '0124	I/O	170124	Load Address Counter Channel 1	$(A)_{1\text{-}16} \rightarrow (AC1)_{1\text{-}16}$ $0 \rightarrow (RC1)_{1\text{-}16}$	2	1.92
SMK '0224	I/O	170224	Load Address Counter Channel 2	$(A)_{1\text{-}16} \rightarrow (AC2)_{1\text{-}16}$ $0 \rightarrow (RC2)_{1\text{-}16}$	2	1.92
SMK '0324	I/O	170324	Load Address Counter Channel 3	$(A)_{1\text{-}16} \rightarrow (AC3)_{1\text{-}16}$ $0 \rightarrow (RC3)_{1\text{-}16}$	2	1.92
SMK '0424	I/O	170424	Load Address Counter Channel 4	$(A)_{1\text{-}16} \rightarrow (AC4)_{1\text{-}16}$ $0 \rightarrow (RC4)_{1\text{-}16}$	2	1.92
SMK '1124	I/O	171124	Load Range Counter Channel 1	$(A)_{2\text{-}16}$ V $(RC1)_{2\text{-}16}$ $\rightarrow (RC1)_{2\text{-}16}$	2	1.92
SMK '1224	I/O	171224	Load Range Counter Channel 2	$(A)_{2\text{-}16}$ V $(RC2)_{2\text{-}16}$ $\rightarrow (RC2)_{2\text{-}16}$	2	1.92
SMK '1324	I/O	171324	Load Range Counter Channel 3	$(A)_{2\text{-}16}$ V $(RC3)_{2\text{-}16}$ $\rightarrow (RC3)_{2\text{-}16}$	2	1.92
SMK '1424	I/O	171424	Load Range Counter Channel 4	$(A)_{2\text{-}16}$ V $(RC4)_{2\text{-}16}$ $\rightarrow (RC4)_{2\text{-}16}$	2	1.92
INA '1124	I/O	131124	Read Range Counter Channel 1	If end-of-range, INA = NOP; otherwise, $1 \rightarrow (A)_1$ $(RC1)_{2\text{-}16} \rightarrow (A)_{2\text{-}16}$ and skip next instruction	2	1.92
INA '1224	I/O	131224	Read Range Counter Channel 2	If end-of-range, INA = NOP; otherwise, $1 \rightarrow (A)_1$ $(RC2)_{2\text{-}16} \rightarrow (A)_{2\text{-}16}$ and skip next instruction	2	1.92
INA '1324	I/O	131324	Read Range Counter Channel 3	If end-of-range, INA = NOP; otherwise, $1 \rightarrow (A)_1$ $(RC3)_{2\text{-}16} \rightarrow (A)_{2\text{-}16}$ and skip next instruction	2	1.92
INA '1424	I/O	131424	Read Range Counter Channel 4	If end-of-range, INA = NOP; otherwise, $1 \rightarrow (A)_1$ $(RC4)_{2\text{-}16} \rightarrow (A)_{2\text{-}16}$ and skip next instruction	2	1.92

Programming Sequence

The following steps should be followed in setting up a transfer via a DMA channel.

a. Load the address counter for the channel to be used. This also clears the associated range counter.

b. Load the range counter for the channel to be used with the TWOs complement of the block size.

c. Activate the device.

Upon completion the range residue may be checked to verify that the transfer took place as expected.

DMA Timing

The DMA can effect a transfer following any memory cycle, providing that the request is transmitted to the DMA 0.57 μs or more before the end of a memory cycle. A later request will be serviced following the next cycle. All computation except shifts, MPY (multiply, DIV (divide), and NRM (normalize) is suspended during the execution of a DMA cycle.

The first DMA cycle of a transfer requires 1.20 μs. All succeeding cycles require exactly one memory cycle, 0.96 μs. The longest wait between a request and its execution can be calculated using the following formula:

$$T_{WC} = 1.20M + 0.96N + \begin{matrix} 1.89 \text{ (input)} \\ 2.64 \text{ (output)} \end{matrix}$$

where T_{WC} is the worst case waiting time in μs, M is the number of higher-priority DMA cycles requiring first-cycle service which may occur during T_{WC}, and N is the number of higher-priority DMA cycles not requiring first-cycle service which may occur during T_{WC}.

Auto Switch Option

Some subchannels may be equipped with an Auto Switch. This option allows the subchannel to be connected to two DMA channels and to switch between them automatically when end-of-range is reached. The channel switch is accomplished in less than one DMA cycle. This option allows long continuous transfers to or from shorter segments of core. It is available on the magnetic tape, and parallel input/output channels.

PRIORITY INTERRUPT OPTION (316/516-25)

The Priority Interrupt Option allows an interrupt source to be uniquely identified without going through the program test required with all sources ORed onto the standard interrupt line (location '00063). The option will service up to 48 distinct sources, each of which will cause a JST* (indirect jump and store) through a different dedicated core location. Locations '64 through '143 are used for Priority Interrupts 1 through 48 (Table 4-10). See *Interrupts and Breaks* for a discussion of interrupts and their handling. Priority interrupt lines are installed in groups of four.

Priority Interrupt Control

Program interrupts requested by Priority Interrupt lines are individually controlled by mask bits associated with each group of interrupt lines. In addition, all Priority Interrupt lines are controlled by the INH and ENB instructions. Priority interrupt is inhibited until an ENB instruction has been executed. Following the execution of an ENB instruction, an interrupt is accepted on any interrupt line having its associated mask bit set (1). Interrupt remains enabled until an INH instruction is executed or an interrupt occurs on any enabled line (forced INH). Following an interrupt or the execution of an INH instruction, interrupts are inhibited until an ENB instruction is executed.

TABLE 4-10. PRIORITY INTERRUPTS AND THEIR MASKS

Priority Interrupt Number	Dedicated Location (Octal)	SMK and Mask Bit	Priority Interrupt Number	Dedicated Location (Octal)	SMK and Mask Bit
	SMK '0120			SMK '0220 (cont.)	
1	64	bit 1	25	114	bit 9
2	65	2	26	115	10
3	66	3	27	116	11
4	67	4	28	117	12
5	70	5	29	120	13
6	71	6	30	121	14
7	72	7	31	122	15
8	73	8	32	123	16
9	74	9			
10	75	10		SMK '0320	
11	76	11			
12	77	12	33	124	bit 1
13	100	13	34	125	2
14	101	14	35	126	3
15	102	15	36	127	4
16	103	16	37	130	5
			38	131	6
	SMK '0220		39	132	7
			40	133	8
17	104	bit 1	41	134	9
18	105	2	42	135	10
19	106	3	43	136	11
20	107	4	44	137	12
21	110	5	45	140	13
22	111	6	46	141	14
23	112	7	47	142	15
24	113	8	48	143	16

The mask bits associated with each group of interrupt lines are controlled by SMK '0X20 instructions. These instructions set the appropriate bit in the mask register if the corresponding bit in the A-Register is a 1 and reset the mask register bit if the corresponding A-Register bit is a 0. Table 4-10 shows the mask assignments for the optional interrupt lines and the SMK instructions that service them.

If an interrupt request occurs during the execution of an SMK instruction disabling that interrupt, the interrupt may or may not be accepted. Therefore the interrupt masks should be changed only when interrupts are inhibited.

MEMORY INCREMENT OPTION (316/516-26)

Groups of four of the dedicated locations for the priority interrupt option may be converted to memory increment locations. A memory increment break takes 3 cycles and performs the following function:

[dedicated location] + 1 → [dedicated location]

There is no overflow indication and no interrupt generated on overflow.

Memory increment breaks are subject to control by the proper mask bits (see *Priority Interrupt Option*) but not subject to control by INH or ENB. These breaks do not cause the CPU to go into EXTEND mode nor do they force an INH.

APPENDIX
LIST OF REFERENCED DOCUMENTS

Document	*Doc. No.*
Programmers Reference Guide	70130072468
Programmers Reference Card	70130071623B
DAP-16 Mod 2 Assembler Manual	70130072442
DAP-16 Reference Manual	70130071629
DDP-516 Programmers Reference Manual	70130071585
316/516 ASR Teletypewriter, High-Speed Paper Tape Reader and High-Speed Paper Tape Punch Programmers Manual	70130072443
DDP-516 Extended Memory and Extended Addressing Option	70130071646
H316 Extended Memory and Extended Addressing Option	70130072276

APPENDIX B
INSTRUCTIONS BY OP CODE

OP Code Octal	Mnemonic	Instruction	Type	Execution Time (μS) DDP-516	H316
01	JMP	Unconditional Jump	MR	0.96	1.6
02	LDA	Load A	MR	1.92	3.2
02	DLD*@	Double Load	MR	2.88	4.8
03	ANA	AND to A	MR	1.92	3.2
04	STA	Store A	MR	1.92	3.2
04	DST*@	Double Store	MR	2.88	4.8
05	ERA	Exclusive OR to A	MR	1.92	3.2
06	ADD	Add	MR	1.92	3.2
06	DAD*@	Double Add	MR	2.88	4.8
07	SUB	Subtract	MR	1.92	3.2
07	DSB*@	Double Subtract	MR	2.88	4.8
10	JST	Jump and Store Location	MR	2.88	4.8
11	CAS	Compare	MR	2.88	4.8
12	IRS	Increment, Replace and Skip	MR	2.88	4.8
13	IMA	Interchange Memory and A	MR	2.88	4.8
14	OCP	Output Control Pulse	IO	1.92	3.2
15	LDX**	Load X	MR	2.88	4.8
15	STX**	Store X	MR	1.92	3.2
16	MPY*	Multiply	MR	5.28	8.8
17	DIV*	Divide	MR	10.56 (max)	17.6
34	SKS	Skip if Ready Line Set	IO	1.92	3.2
54	INA	Input to A	IO	1.92	3.2
74	OTA***	Output from A	IO	1.92	3.2
74	SMK***	Set Mask	IO	1.92	3.2
0400	LRL	Long Right Logical Shift	SH	†	††
0401	LRS	Long Arithmetic Right Shift	SH	†	††
0402	LRR	Long Right Rotate	SH	†	††
0404	LGR	Logical Right Shift	SH	†	††
0405	ARS	Arithmetic Right Shift	SH	†	††
0406	ARR	Logical Right Rotate	SH	†	††
0410	LLL	Long Left Logical Shift	SH	†	††
0411	LLS	Long Arithmetic Left Shift	SH	†	††
0412	LLR	Long Left Rotate	SH	†	††
0414	LGL	Logical Left Shift	SH	†	††
0415	ALS	Arithmetic Left Shift	SH	†	††
0416	ALR	Logical Left Rotate	SH	†	††
000000	HLT	Halt	G	1.44	2.4
000005	SGL*	Enter Single Precision Mode	G	0.96	1.6
000007	DBL*	Enter Double Precision Mode	G	0.96	1.6
000011	DXA*	Disable Extended Addressing	G	0.96	1.6
000013	EXA*	Enable Extended Addressing	G	0.96	1.6
000021	RMP*	Reset Memory Parity Error	G	0.96	1.6
000041	SCA*	Shift Count to A	G	0.96	1.6
000043	INK	Input Keys	G	0.96	1.6
000101	NRM*	Normalize	G	†	††
000201	IAB	Interchange A and B	G	0.96	1.6
000401	ENB	Enable Program Interrupt	G	0.96	1.6
001001	INH	Inhibit Program Interrupt	G	0.96	1.6

OP Code Octal	Mnemonic	Instruction	Type	Execution Time (μS) DDP-516	H316
001401	ERM*	Enter Restricted Mode	G	0.96	1.6
100000	SKP	Unconditional Skip	G	0.96	1.6
100001	SRC	Skip if C Reset	G	0.96	1.6
100002	SR4	Skip if Sense Switch No. 4 Reset	G	0.96	1.6
100004	SR3	Skip if Sense Switch No. 3 Reset	G	0.96	1.6
100010	SR2	Skip if Sense Switch No. 2 Reset	G	0.96	1.6
100020	SR1	Skip if Sense Switch No. 1 Reset	G	0.96	1.6
100036	SSR	Skip if No Sense Switch Set	G	0.96	1.6
100040	SZE	Skip if A Zero	G	0.96	1.6
100100	SLZ	Skip if (A_{16}) is Zero	G	0.96	1.6
100200	SPN*	Skip on No Memory Parity Error	G	0.96	1.6
100400	SPL	Skip is A Sign Plus	G	0.96	1.6
101000	NOP	No Operation	G	0.96	1.6
101001	SSC	Skip if C Bit Set	G	0.96	1.6
101002	SS4	Skip if Sense Switch No. 4 Set	G	0.96	1.6
101004	SS3	Skip if Sense Switch No. 3 Set	G	0.96	1.6
101010	SS2	Skip if Sense Switch No. 2 Set	G	0.96	1.6
101020	SS1	Skip if Sense Switch No. 1 Set	G	0.96	1.6
101036	SSS	Skip if Any Sense Switch Set	G	0.96	1.6
101040	SNZ	Skip if A Non-Zero	G	0.96	1.6
101100	SLN	Skip if (A_{16}) is One	G	0.96	1.6
101200	SPS*	Skip on Memory Parity Error	G	0.96	1.6
101400	SMI	Skip if A Sign Minus	G	0.96	1.6
140024	CHS	Complement A Sign	G	0.96	1.6
140040	CRA	Clear A	G	0.96	1.6
140100	SSP	Set A Sign Plus	G	0.96	1.6
140200	RCB	Reset C Bit	G	0.96	1.6
140320	CSA	Copy Sign And Set Sign Plus	G	0.96	1.6
140401	CMA	Complement A	G	0.96	1.6
140407	TCA	2's Complement A	G	1.44	2.4
140500	SSM	Set A Sign Minus	G	0.96	1.6
140600	SCB	Set C Bit	G	0.96	1.6
141044	CAR	Clear A, Right Half	G	0.96	1.6
141050	CAL	Clear A, Left Half	G	0.96	1.6
141140	ICL	Interchange and Clear Left Half of A	G	0.96	1.6
141206	AOA	Add One to A	G	0.96	1.6
141216	ACA	Add C to A	G	0.96	1.6
141240	ICR	Interchange and Clear Right Half of A	G	0.96	1.6
141340	ICA	Interchange Characters in A	G	0.96	1.6
171020	OTK	Output Keys	G	1.92	3.2

@ CPU must be in double precision mode.

* Optional instruction.

** Instructions STX and LDX have the same operation code (15)
STX has an index bit of 0; LDX has an index bit of 1.

*** Instructions OTA and SMK have the same operations code (74).
SMK has device address D = 20 or 24; OTA has D ≠ 20 or 24.

N Number of shifts

† 0.96 +0.48N

†† 1.6 +0.8N

HONEYWELL INFORMATION SYSTEMS

Technical Publications Remarks Form*

TITLE:	SERIES 16 MODELS 316 AND 516 PROGRAMMERS' REFERENCE MANUAL	ORDER NO:	BX47, REV. 0
		DATED:	NOVEMBER, 1970

ERRORS IN PUBLICATION:

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION:

(PLEASE PRINT)

FROM: NAME _____ DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

_____

CUT ALONG LINE

*Your comments will be promptly investigated by appropriate technical personnel, action will be taken as required, and you will receive a written reply. If you do not require a written reply, please check here. ☐

CUT ALONG LINE

FOLD ALONG LINE

POSTAGE STAMP NOT NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL: FIRST CLASS PERMIT NO. 39531, NEWTON HIGHLANDS, MASS.

POSTAGE WILL BE PAID BY:

HONEYWELL INFORMATION SYSTEMS

60 WALNUT STREET
WELLESLEY HILLS, MASS. 02181

ATTN: U.S. GROUP PUBLICATIONS, MS 050

FOLD ALONG LINE

Honeywell

The Other Computer Company:

Honeywell

HONEYWELL INFORMATION SYSTEMS

2592
5971
Printed in U.S.A.

BX47, Rev. 0