UNI COMP
incorporated



COMP 18

# REFERENCE MANUAL

18219 PARTHENIA STREET, NORTHRIDGE, CALIFORNIA 91324 • (213) 886-7722

# COMP-18

# REFERENCE MANUAL

# TABLE OF CONTENTS

## ILLUSTRATIONS

# COMP-18 SYSTEM SUMMARY

This manual describes the functional operation of the COMP-18 computer. The reader is assumed to be knowledgeable of computer terminology, operation, and programming.

In addition to this manual, the COMP-18 user has a special manual for: optional units, hardware maintenance, UNICAP-II Assembler, BASIC compiler, Debug, Trace, and Utility Programs, and Diagnostic programs.

## COMP-18 CHARACTERISTICS

COMP-18 is a general purpose computer designed expressly for unattended, stand alone operation in a production area environment. The COMP-18 uses high speed and high reliability TTL circuits. The command structure and input/output system of the COMP-18 is ideally suited to the requirement of real time, on line applications.

Figure 1 summarizes COMP-18 specifications.

## SOFTWARE

Programs can be developed for the COMP-18 using the UNICAP-II Assembler or the BASIC Language compiler. UNICAP-II will operate in either a one pass conversational mode or, for exceptionally long programs, in a two pass mode. The BASIC compiler conforms to the original Dartmouth College BASIC Language compiler specification. UNICAP-II and BASIC will operate on a minimum COMP-18 with 4,096 words of memory and an ASR-33 for input and output. UNICAP-II and BASIC are written to operate efficiently with increased memory capacity and both will operate with higher speed input/output devices, such as,punched cards, magnetic tape, line printers, and high speed paper tape readers and punches.

In addition to UNICAP-II and BASIC, there are additional programs available for mathematic operations, program checkout, diagnostic testing,and input/output handlers.

A library of fixed point mathematic subroutines includes: sine, cosine,arctangent, log e, log 10, exponential e, exponential 10, square root, multiply and divide.

The floating point operations are programmed to be executed in an interpretive mode. All conversion, arithmetic, and trigonometric operations are included in the floating point package as part of UNICAP-II.

Input/output data handling subroutines include: input, smoothing, interpolation, conversion to BCD, and real time display.

Diagnostic programs are provided for checking the COMP-18 memory, central processor, and optional peripheral devices.

OPTIONS

The COMP-18 computer system is adaptable, through hardware main frame options and peripheral devices, to assume the exact configuration necessary to guarantee maximum system performance for on line real time processing and control applications.

Following is a list of COMP-18 features and options:

Memory expandable from basic 4,096 words to 262,144 words in increments of 4,096.

Interrupts expandable in groups of eight to a maximum of 64.

Multiply, Divide, and Square Root (MDSR) hardware operations may be added to the COMP-18 instruction set if faster execution times are required.

Real Time Clock (RTC) provides an interrupt after a specified elapsed time. The interval is set under program control and elapsed time may be read under program control prior to, or after, the interrupt.

Power Fail Restart (PFR) will cause the COMP-18 to begin execution of commands from a fixed location upon resumption of operation after a power failure.

Peripheral devices standard to the COMP-18 are:

- Magnetic Tape
- Cassette Magnetic Tape
- Disc
- Drum
- Card Reader
- Card Punch
- Line Printer
- Teletype
- Paper Tape Reader
- Paper Tape Punch

- Cathode Ray Tube Display
- Decimal Display
- Binary Display
- Discrete Input Module (bi-level TTL signal)
- Discrete Output Module (bi-level TTL signal)
- Data Modems (RS-232 interface)
- Analog/Digital input devices
- Digital/Analog output devices
- Communication Terminals (100,200,300 series)

TYPE:
General Purpose Stored Program

MEMORY:
.888 us cycle time
4,096 to 262,144 - 18 bit words

BASIC COMMAND FUNCTIONS:
Arithmetic and Logic
Store
Branching
Shift
Input/Output
Indexing
Halt and Unconditional Jump

OPERAND ADDRESSING MODES:
Direct
Indirect
Indexed
Immediate

INDEX REGISTERS:
Six, Part of Core Memory

EXECUTION SPEED (microseconds):

| | | |
|---|---|---|
| Arithmetic and Logic | | |
| Immediate | 1.77 | |
| Direct Address | 2.22 | |
| Index or Indirect | 3.11 | |
| Jump Commands | | |
| Direct Address | 1.77 | |
| Index or Indirect | 2.66 | |
| Shift Commands | | |
| Direct Address | 2.22 | +.888n (number of shifts) |
| Multiply | 7.2 | |
| Divide | 7.6 | |
| Square Root | 21.6 | |

INTERRUPTS:
Expandable to 64

SIZE:
8-3/4 x 22-3/4 x 19 inches

WEIGHT:
48 pounds

COMP-18 General Specifications

FIGURE 1

# COMPUTER ORGANIZATION

The COMP-18 is a digital computer which performs programmed operations upon binary represented data quantities. The operations that are performed are defined by a sequence of COMP-18 commands. The commands are stored in memory and operate on data quantities stored in memory or read from external devices.

This section describes COMP-18 central processor functions and characteristics of: command formats, command execution, operating registers, data representations, and addressing schemes.

## COMMAND FORMAT

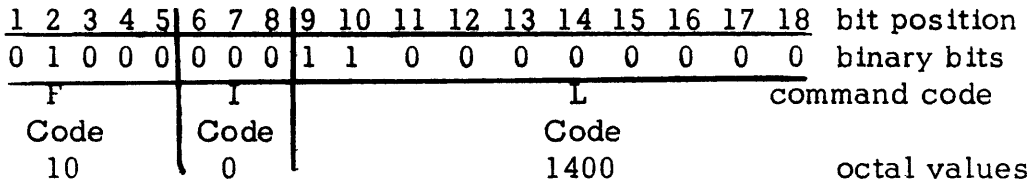
A command is a single COMP-18 program instruction. A command is contained in 18 bits and is stored in a single memory location. The command is made up of three parts: Function, Index, Location.

The Function, or F Code, specifies the type of operation that will be performed. There are 31 Function Codes in the COMP-18 for performing arithmetic, logic, decision making, housekeeping, and input/output operations. The F Code occupies the first five bits of the command word, bit positions 1 through 5. The section, "Function Set", describes each COMP-18 Function.

The Index, or I Code, specifies the type of address creation that will be used in the execution of the command. An Address may be direct, indirect, or indexed. For indexed addressing, the I Code designates the use of one of the six index registers. For certain Functions, the I Code may specify no address, in which case, the operand is taken from the Location portion of the command word. The I Code is three bits of the command word, bit positions 6 through 8.

The Location, or L Code, is used directly, or is modified, to form an address, operand, shift count, or I/O device code. The I Code specifies the modifications, if any, that will be made to the L Code. The Function Code dictates the meaning of the L Code. The L Code occupies the last ten bits of the command word, bit positions 9 through 18.

The following is the command word, in binary and octal, which would load the contents of memory location $1400_8$ into the accumulator:




The octal representation of the F Code for the Load Accumulator Positive Function is $10_8$. An I Code of zero specifies direct addressing. The L Code contains the octal value $1400_8$ and is used to directly address memory.

## EFFECTIVE ADDRESSING

An effective address is developed for each command. The method of developing the effective address is specified by the I Code. The L Code is the effective address or is modified to form the effective address. The effective address is formed in the same manner for all Functions except INC, reference the detailed INC description for differences. The methods of effective address creation are as follows:

| I Code | Effective Address Creation |
|---|---|
| 0 | The L Code is the effective address (direct addressing). |
| 1 - 6 | The contents of the specified index register are added to the L Code to form the effective address (indexed addressing). |
| 7 | The L Code is used to reference a memory location which contains the effective address (indirect addressing). |

The arithmetic Function Codes, with an I value of 6, are an exception to the above. For these Functions an I Code of 6 designates the contents of the L Code as the operand.

The interpretation of the effective address is dependent upon the Function being performed. The following shows the catagories of Functions and the corresponding use that is made of the effective address.

| Function | Effective Address Usage |
|---|---|
| Arithmetic | The effective address is an operand or an operand address. |
| Jump | The effective address is the memory address of the next command to be executed, assuming conditions for jumping are satisfied. |
| Shift | The effective address is the shift count. |
| Input/Output | The effective address is a peripheral device code. |

Since the effective address is a 18 bit unsigned quantity, the highest possible memory address is $262,143_{10}$. Since the L Code is ten bits, the maximum memory address that can be directly referenced is $1777_8$. Referencing locations above that address requires use of indirect or index addressing.

## INDEXING

Six core memory locations have been dedicated to function as index registers. They are locations $101_8$ through $106_8$. The specific location is selected by an I value of 1 through 6, as follows:

| I Value | Specified Memory Location (octal) |
|---|---|
| 1 | 101 |
| 2 | 102 |
| 3 | 103 |
| 4 | 104 |
| 5 | 105 |
| 6 | 106 |

When indexing is specified, the contents of the selected memory location are added to the value of the L Code to form the effective address.

## OPERATING REGISTERS

The execution of a command in the COMP-18 central processor involves five registers. They are:

A register or Accumulator is the main arithmetic and operational register. It is normally the source or destination of information for each command execution.

P register or Program Counter contains the address of the command being executed. During execution of a command, the counter is normally incremented by one to reference the next sequentially addressed command. When a jump command is executed, the counter is set to the specified address.

L register or Location Address is used in creating and holding the effective address during command execution.

F/I register or Function Index holds the Function and Index Codes of the command during execution.

M register or Memory Register is part of each memory module and temporarily stores data being taken from, or placed in, memory.

The F/I register is 8 bits in length. All other registers are 18 bits in length.

The use of the registers in the execution of a command is as follows:

1) The contents of the P register are used as a memory reference address.

2) The contents of the reference memory location are read into the M register.

3) The M register bits 1 through 8 are placed in the F/I register and M register bits 9 through 18 are placed in the L register.

4) The contents of the P register are incremented by one and returned to the P register.

5) The F/I register is interpreted, the effective address formed in the L register, and the command executed.

## NUMERIC FORMAT

The numeric representation of the eighteen bit COMP-18 binary word may be either a signed or unsigned value.

For an unsigned value, bit position 1 is the most significant bit position and bit position 18 is the least significant bit position. The maximum value that can be contained in an unsigned 18 bit word is $262,144_{10}$, expressed as:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | bit position |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | binary bits |

For a signed value, bit position 1 is the sign bit, bit position 2 is the most significant bit position, and bit position 18 is the least significant bit position. A zero sign bit indicates a positive number, a one sign bit indicates a negative value. The maximum value that can be expressed in a signed 18 bit word is $131,072_{10}$, expressed as:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | bit position |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | binary bits |

The 2's complement is formed by inverting each bit and adding one to the least significant bit. The 2's complement of $131,072_{10}$, the largest possible 18 bit signed, negative quantity, is expressed as:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | bit position |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | binary bits |

## ARITHMETIC OPERATIONS

The COMP-18 adds, bit-by-bit, 18 bit quantities. When executing a programmed Add, the contents of the 18 bit accumulator are added to the operand and the sum placed in the accumulator.

The COMP-18 performs a programmed subtraction by converting the subtrahend to 2's complement and adding.

When an arithmetic operation results in the carry of a one out of bit position 1, the Exchange Bit is set to one. The Exchange Bit is set to zero when no carry occurs.

Testing for carry is accomplished by shifting the Exchange Bit into the accumulator and performing the appropriate conditional test.

When an arithmetic operation results in the loss of a significant bit from bit position 2, the overflow indicator is set. When overflow occurs, the sign bit contains the most significant bit of the result of the arithmetic operation; however, the sign is lost from the accumulator and enters the Exchange Bit. For addition, overflow occurrence is indicated by a sign change in the accumulator after adding like signed values. For subtraction, overflow occurrence is indicated by a sign change in the accumulator after subtracting different signed values. Overflow is also indicated when an arithmetic left shift results in a significant bit being shifted out of bit position 2.

In the following examples, the four bits represent the contents of bit positions 1, 2, 3, and 4 of the 18 bit accumulator. Negative values are in 2's complement format. Examples of addition, and the resultant Exchange Bit and overflow indicator settings, are:

| | | |
|---|---|---|
| Exchange Bit 0, overflow set | 0 1 1 1 accumulator<br>0 0 1 0 operand<br>1 0 0 1 | $7 + 2 = 11_8$ |
| Exchange Bit 1, overflow not set | 0 1 1 1 accumulator<br>1 1 1 0 operand<br>0 1 0 1 | $7 + (-2) = 5_8$ |
| Exchange Bit 1, overflow set | 1 0 1 1 accumulator<br>1 0 1 1 operand<br>0 1 1 0 | $-5 + (-5) = -12_8$ |
| Exchange Bit 0, overflow not set | 0 1 0 0 accumulator<br>0 0 1 1 operand<br>0 1 1 1 | $4 + 3 = 7_8$ |

Examples of subtraction, and the resultant Exchange Bit and overflow indicator settings are:

| | | |
|---|---|---|
| Exchange Bit 1, overflow not set | 0 1 1 1 accumulator<br>1 1 0 0 operand<br>0 0 1 1 | $7 - 4 = 3_8$ |
| Exchange Bit 0, overflow set | 0 1 1 1 accumulator<br>0 1 0 0 operand<br>1 0 1 1 | $7 - (-4) = 13_8$ |

| | | |
|---|---|---|
| Exchange Bit 1, overflow set | 1 0 1 1 accumulator<br>1 0 1 0 operand<br>0 1 0 1 | $-5\ -6=-13_8$ |
| Exchange Bit 0, overflow not set | 1 0 1 1 accumulator<br>0 0 1 1 operand<br>1 1 1 0 | $-5\ -(-3)=-2_8$ |

## RESERVED MEMORY LOCATIONS

Certain COMP-18 memory locations are dedicated to hardware operations. The hardware operation has priority for the usage of these locations. Care should be taken when using these locations for program operations that a hardware function does not result in altering either data values or the processing sequence.

Figure 2 shows the reserved memory locations and their use.

| LOCATION (octal) | USAGE |
|---|---|
| 0000 | A switch action at the control panel will cause execution of the contents of this location. |
| 0001 | When an interrupt occurs, the Program Counter is set to this address and the command stored in this location is executed. |
| 0100 | The contents of the Program Counter are stored in this location when an interrupt occurs. |
| 0101 | Index Register 1 |
| 0102 | Index Register 2 |
| 0103 | Index Register 3 |
| 0104 | Index Register 4 |
| 0105 | Index Register 5 |
| 0106 | Index Register 6 |
| 0107 | Execution of JPR Function causes the contents of the Program Counter to be stored in this location. |
| 2000 | Switch action at the control panel will cause execution of the contents of this location. |

RESERVED MEMORY LOCATIONS

FIGURE 2

## FUNCTION SET

The COMP-18 has 31 hardware decoded Functions. Each Function is a single instruction available to the programmer for the development of COMP-18 processing routines. The Functions are grouped into four catagories of processing operations: arithmetic and logic, jump, shift, and operational.

This section describes the command word format, detailed description of Function operation,and execution timing. Execution timing is given in cycles. A single cycle is .888 microseconds.

Figure 3 presents a summary of the Function Codes.

| F CODE | MNEMONIC | OPERATIONAL DESCRIPTION |
|---|---|---|
| 00 | HLT | Jump to Effective Address and if the Halt Enable switch is ON, stop computing. |
| 01 | INT | Jump to Fixed Address if Interrupted.(Not available to programmer.) |
| 02 | INC | Increment the designated I register and if it is negative, Jump to the location whose address is the sum of the A register and Program Counter. (L is the increment value.) |
| 03 | JPR | Store the contents of the Program Counter in location $107_8$ and Jump to the effective address. |
| 04 | SPC | Store the contents of the Program Location Counter in Effective Address. |
| 05 | STA | Store (ACC) in Effective Address. |
| 06 | RIN | Read in Data. Put Effective Address on external lines and read input lines into Accumulator. |
| 07 | OUT | Output Data. Put (ACC) and Effective Address on external lines. |

FUNCTION SET

FIGURE 3

| F CODE | MNEMONIC | OPERATIONAL DESCRIPTION |
|---|---|---|
| 10 | LAP | Load Accumulator with Operand. |
| 11 | LAN | Load Accumulator Negatively with Operand. |
| 12 | ADD | Add Operand to (ACC) and put the sum in Accumulator. |
| 13 | SUB | Subtract Operand from (ACC) and store the difference in the Accumulator. |
| 14 | AND | Logical AND of (ACC) with Operand. |
| 15 | ANI | Logical AND of (ACC) with Operand inverted. |
| 16 | LOR | Logical OR of (ACC) with Operand. |
| 17 | EXO | Exclusive OR of (ACC) with Operand. |
| 20 | JAZ | Jump to Effective Address if (ACC) is zero. |
| 21 | JNZ | Jump to Effective Address if (ACC) is non-zero. |
| 22 | JAP | Jump to Effective Address if (ACC) is positive. |
| 23 | JAN | Jump to Effective Address if (ACC) is negative. |
| 24 | JEP | Jump to Effective Address if Accumulator has an even number of one's. |
| 25 | JOP | Jump to Effective Address if Accumulator has an odd number of one's. |
| 26 | JOF | Jump to Effective Address if Overflow has been set and reset overflow. |

FUNCTION SET (CONTD)

FIGURE 3

| F CODE | MNEMONIC | OPERATIONAL DESCRIPTION |
|---|---|---|
| 27 | JMP | Jump to Effective Address Unconditionally. |
| 30 | SLA | Shift (ACC) Left Arithmetically, a number of places specified by the Effective Address, (Sign unchanged). |
| 31 | SLX | Shift (ACC) left, through Exchange Bit. |
| 32 | SLL | Shift (ACC) left Logically, a number of places specified by Effective Address. (Sign shifts also). |
| 33 | SLE | Shift (ACC) Left End-Around, a number of places specified by the Effective Address. |
| 34 | SRA | Shift (ACC) Right Arithmetically, a number of places specified by the Effective Address, (sign copies into next bit, but remains unchanged). |
| 35 | SRX | Shift (ACC) right, through Exchange Bit. |
| 36 | SRL | Shift (ACC) Right Logically, a number of places specified by the Effective Address, (sign is copied into next bit and set to zero.) |
| 37 | SRE | Shift (ACC) Right End-Around, a number of places specified by the Effective Address. |

FUNCTION SET (CONTD)

FIGURE 3

## ARITHMETIC AND LOGIC FUNCTIONS

These Functions combine an operand with the accumulator according to the arithmetic or logic rules associated with the individual Function.

For these Functions, the effective address is either the operand, direct operand address, indirect operand address, or indexed operand address. Figure 4 shows the development and use of the effective address for this group of Functions.

The mode of addressing determines the number of machine cycles to execute the command. The following gives the number of machine cycles for each address mode.

| Operand Address Mode | Machine Cycles |
|---|---|
| L Code is operand | 2 |
| Direct Operand Address | 2.5 |
| Indirect or Indexed Operand Address | 3.5 |

Allowable F Codes (octal): 10,11,12,13,14,15,16,17

| I Code (Octal) | Effective Address Usage |
|---|---|
| 0 | The L Code is the direct address of the operand. |
| 1,2,3,4,5 | The L Code is added to the contents of location 101,102,103,104, or 105 to form the operand address. |
| 6 | The L Code is the operand. |
| 7 | The L Code is used to address a memory location whose contents are used as the operand address. |

ARITHMETIC AND LOGIC FUNCTION
EFFECTIVE ADDRESS USAGE
FIGURE 4

EXAMPLES:

| Location | Contents or Command Word | Result |
|---|---|---|
| 101 | 000500 | |
| . | | |
| . | | |
| 200 | 767 | |
| . | | |
| . | | |
| 500 | LAP 6 0030 | Accumulator contains 000030 |
| 501 | ADD 1 0077 | Accumulator contains 000063 (000030+ 000033) |
| 502 | ADD 7 0200 | Accumulator contains 000064 (000063+ 000001) |
| . | | |
| . | | |
| 577 | 000033 | |
| . | | |
| 767 | 000001 | |

ARITHMETIC AND LOGIC FUNCTION
EFFECTIVE ADDRESS USAGE
(CONTD)

FIGURE 4

LOAD ACCUMULATOR POSITIVE, LAP, F = 10

The operand, taken from memory or the L portion of the command word, is placed in the accumulator. Execution of this command replaces the previous contents of the accumulator.

Register affected: Accumulator.

LOAD ACCUMULATOR NEGATIVE, LAN, F = 11

This Function is identical to the Load Accumulator Positive except the operand is converted to 2's complement, and the sign reversed, prior to loading. Execution of this command replaces the previous contents of the accumulator.

A zero operand results in all accumulator bits being set to zero.

Register affected: Accumulator.

ADD, ADD, F = 12

The contents of the accumulator and the operand are added together. The sum replaces the previous contents of the accumulator. If addition results in a carry out of bit position 1, the Exchange Bit is set to a one. If no carry occurs, the Exchange Bit is set to zero. If addition results in the loss of a significant bit from bit position 2, the overflow indicator is set.

Register affected: Accumulator, Exchange Bit, overflow indicator.

SUBTRACT, SUB, F = 13

The operand is subtracted from the contents of the accumulator and the remainder placed in the accumulator. Subtraction is accomplished by taking the 2's complement and adding. If the subtraction results in a carry out of bit position 1, the Exchange Bit is set to one. If no carry occurs, the Exchange bit is zero. If the subtracting results in the loss of a significant bit from bit position 2, the overflow indicator is set.

Register affected: Accumulator, Exchange Bit, overflow indicator.

LOGICAL AND, AND, F = 14

The contents of the accumulator are ANDed bit-by-bit with the operand. The result replaces the previous contents of the accumulator.

The four possible ANDing combinations are:

| Operand | Accumulator | Result |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Register affected: Accumulator.

LOGICAL AND INVERTED, ANI, F = 15

This Function is similar to the Logical AND except the operand is converted to 1's complement prior to the AND operation.

Register affected: Accumulator.

LOGICAL OR, LOR, F = 16

The contents of the accumulator are ORed with the operand. The result replaces the previous contents of the accumulator.

The four possible ORing combinations are:

| Operand | Accumulator | Result |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Register affected: Accumulator.

EXCLUSIVE OR, EXO, F = 17

The contents of the accumulator are exclusive-ORed with the operand. The result replaces the previous contents of the accumulator.

The four possible exclusive-ORing combinations are:

| Operand | Accumulator | Result |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Register affected: Accumulator.

JUMP FUNCTIONS

These Functions provide for setting the Program Counter to a particular memory address. All but one of the Functions first tests for a particular condition and, dependent upon test results, will jump to the specified address. For conditional jumps, if the condition is not satisfied, the next sequential command will be executed.

Figure 5 shows how the effective address is used as a jump address.

Execution time for a directly addressed jump Function is 2 cycles. Three cycles are required to execute an indirect or index addressed jump Function.

Allowable F Codes (octal): 20,21,22,23,24,25,26,27

| I Code | Effective Address Usage |
|---|---|
| 0 | The L Code is the jump address. |
| 1,2,3,4 | The L Code is added to the specified index |
| 5,or 6 | register to form the jump address. |
| 7 | The L Code addresses a memory location whose contents are the jump address. |

EXAMPLES:

| Location | Contents or Command Word | Result |
|---|---|---|
| 101 | 000370 | |
| . | | |
| 200 | 002776 | |
| . | | |
| . | | |
| 500 | JMP 1 0117 | After execution the Program Counter would contain $000507_8$. |
| 501 | JMP 0 0001 | After Execution, the Program Counter would contain $000001_8$. |
| 502 | JMP 7 0200 | After Execution, the Program Counter would contain $002776_8$. |

JUMP FUNCTION EFFECTIVE ADDRESS USAGE

FIGURE 5

JUMP IF ACCUMULATOR ZERO, JAZ, F = 20

If all accumulator bits are zero, the jump address is placed in the Program Counter and the command stored in that location is executed.

Register affected: Program Counter.

JUMP IF ACCUMULATOR NOT ZERO, JNZ, F = 21

If an accumulator bit is a one, the jump address is placed in the Program Counter and the command stored in that location is executed.

Register affected: Program Counter.

JUMP IF ACCUMULATOR POSITIVE, JAP, F = 22

If bit position 1 is zero, the jump address is placed in the Program Counter and the command stored in that location is executed.

Register affected: Program Counter.

JUMP IF ACCUMULATOR NEGATIVE, JAN, F = 23

If bit position 1 is a one, then the jump address is placed in the Program Counter and the command stored in that location is executed.

Register affected: Program Counter.

JUMP IF EVEN PARITY, JEP, F = 24

If the accumulator contains an even number of one bits, the jump address will be placed in the Program Counter and the command stored in that location will be executed.

Register affected: Program Counter.

JUMP IF ODD PARITY, JOP, F = 25

If the accumulator contains an odd number of one bits, the jump address will be placed in the Program Counter and the command stored in that location will be executed.

Register affected: Program Counter.

JUMP IF OVERFLOW, JOF, F = 26

If the overflow indicator is on, the jump address will be placed in the Program Counter and the command stored in that location will be executed.

The overflow indicator is set when an arithmetic computation or shift results in the loss of a significant bit from bit position 2 (reference page 12). Execution of the command clears the overflow indicator.

Register affected: Program Counter, Overflow indicator.

JUMP, JMP, F = 27

Unconditionally, the effective address is placed in the Program Counter and the command stored in that location will be executed.

Register affected: Program Counter.

SHIFT FUNCTIONS

These Functions provide for various ways of shifting and circulating the contents of the accumulator.

The effective address is the shift count. Figure 6 shows the creation and use of the effective address as a shift count.

Execution time is dependent upon the method of specifying the shift count and the number of places shifted. The following shows shift count specification and the corresponding machine cycles for execution.

| Shift Count Specification | Machine Cycles |
|---|---|
| Contained in L Code | 2.5 +n* |
| Indirect or index address referenced | 3.5 +n* |

*n is the number of positions shifted.

Allowable F Codes (octal): 30, 31,32,33,34,35,36,37

| I Code (Octal) | Effective Address Usage |
|---|---|
| 0 | The L Code contains the shift count. |
| 1,2,3,4 5, or 6 | The L Code plus the contents of the specified index register is the shift count. |
| 7 | The L Code addresses a memory location whose contents are the shift count. |

EXAMPLE:

| Location | Contents or Command Word | Result |
|---|---|---|
| 101 | 000005 | |
| . . | | |
| 200 | 000004 | |
| . . | | |
| 500 | SLA 0 0003 | Arithmetic left shift of 3. |
| 501 | SRA 1 0000 | Arithmetic right shift of 5. |
| 502 | SLL 7 0200 | Logical left shift of 4. |

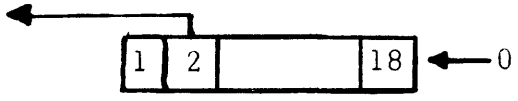
SHIFT FUNCTION EFFECTIVE ADDRESS USAGE

FIGURE 6

SHIFT LEFT ARITHMETICALLY, SLA, F = 30

Accumulator bit positions 2 through 18 are shifted left. Bit position 1 is unchanged. Bits shifted out of bit position 2 are lost. Zeros are shifted into bit position 18.

Shifting a one bit of a positive number out of bit position 2 will set the overflow indicator. Shifting a zero bit of a negative number out of bit position 2 will set the overflow indicator.

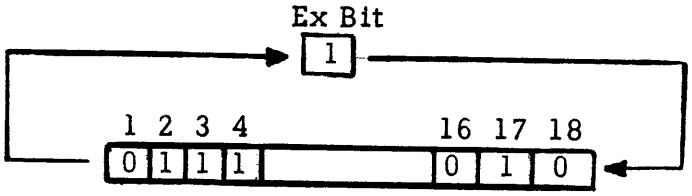



Register affected: Accumulator.

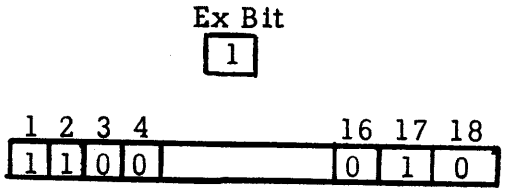
SHIFT LEFT THROUGH EXCHANGE BIT, SLX, F = 31

The accumulator bit and Exchange Bit are circulated left. The Exchange Bit enters the accumulator at bit position 18. The contents of bit position 1 enters the Exchange Bit.

If the Exchange Bit and accumulator bits are,




and a shift count of 2 is executed, the result would be,




Register affected: Accumulator and Exchange Bit.

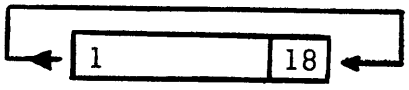
SHIFT LEFT LOGICALLY, SLL, F = 32

All accumulator bits are shifted left. Bits shifted out of bit position 1 are lost. Zeros are shifted into bit position 18.


Register affected: Accumulator.

SHIFT LEFT END-AROUND, SLE, F = 33

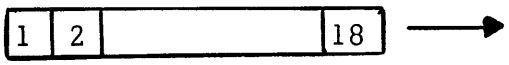
The accumulator bits are circulated left to right. The content of bit position 1 is shifted into bit position 18.




Register affected: Accumulator.

SHIFT RIGHT ARITHMETICALLY, SRA, F = 34

Accumulator bit positions 2 through 18 are shifted right. Bits shifted out of bit position 18 are lost. Bits shifted into bit position 2 repeat the sign bit.

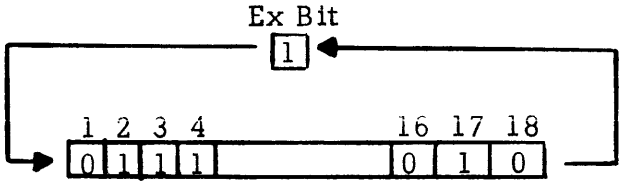



Register affected: Accumulator.

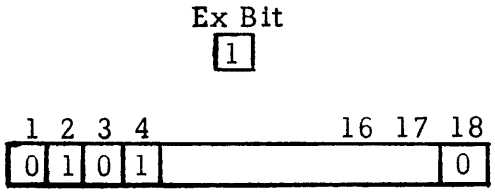
SHIFT RIGHT THROUGH EXCHANGE BIT, SRX, F = 35

The accumulator bits and Exchange Bit are circulated right. The Exchange Bit enters the accumulator at bit position 1. The contents of bit position 18 enters the Exchange Bit.

If the Exchange Bit and accumulator bits are,



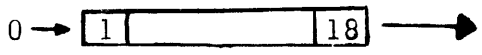

and a shift count of 2 is executed, the result would be,




Register affected: Accumulator and Exchange Bit.

SHIFT RIGHT LOGICALLY, SRL F = 36

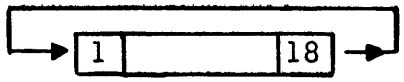
All accumulator bits are shifted right. Bits shifted out of bit position 18 are lost. Zeros are shifted into bit position 1.




Registers affected: Accumulator.

SHIFT RIGHT END-AROUND, SRE, F = 37

The accumulator bits are circulated right.




Register affected: Accumulator

OPERATIONAL FUNCTIONS

The Functions in this group are miscellaneous operations associated with hardware operations, data transfer, processing, and housekeeping.

Figure 7 shows the creation and meaning of the effective address for input and output Functions. Figure 8 shows the creation and meaning of the effective address for SPC and STA Functions. The effective address conventions for the HLT and JPR Function are identical to jump Functions.

Execution time for the INC command is 3.5 cycles. With the Halt Enable off, the execution time for HLT is 2 cycles for direct addressing and 3 cycles for indirect or indexed addressing. Execution time for all other Functions in this group is either 2.5 cycles for direct address or 3.5 cycles for indirect or indexed addressing.

HALT, HLT, F = 00

Operation of this Function is dependent upon the setting of the HALT ENABLE switch. If the switch is off, the Function is executed as an unconditional jump. If the switch is on, the COMP-18 halts after executing the jump.

The effective address is the memory address for the jump.

Register affected: Program Counter.

Allowable F Codes (octal): 06, 07

| I Code (Octal) | Effective Address Usage |
|---|---|
| 0 | The L Code contains the device code. |
| 1,2,3,4, 5,or 6 | The L Code plus the contents of the index register is the device code. |
| 7 | The L Code addresses a memory location whose contents are used as the device code. |

EXAMPLE:

| Location | Contents or Command Word | Result |
|---|---|---|
| 101 | 000100 | |
| . | | |
| . | | |
| 300 | 000102 | |
| . | | |
| . | | |
| 500 | RIN 0 0101 | The device code is 101. |
| 501 | OUT 1 0000 | The device code is 100. |
| 502 | RIN 7 0300 | The device code is 102. |

INPUT/OUTPUT COMMAND EFFECTIVE ADDRESS USAGE

FIGURE 7

Allowable F Codes (octal): 04, 05

| I Code (Octal) | Effective Address Usage |
|---|---|
| 0 | The L Code is a memory address. |
| 1,2,3,4, 5, or 6 | The L Code is added to the specified index register to form a memory address. |
| 7 | The L Code addresses a memory location whose contents are used as the memory address. |

EXAMPLE:

| Location | Contents or Command Word | Result |
|---|---|---|
| 101 | | |
| . | | |
| . | | |
| 400 | SPC 0 0101 | Location 101 will contain $401_8$ (Contents of the Program Counter). |
| 401 | STA 1 0007 | The contents of the accumulator will be stored in location $410_8$. |

SPC AND STA COMMAND EFFECTIVE ADDRESS USAGE

FIGURE 8

INCREMENT INDEX, INC., F = 02

The L Code is added to the contents of the index register specified by the I Code and the result stored in the specified index register. The I Code may have a value, 0 through 7, which will reference memory locations $100_8$ through $107_8$.

If the result of the computation is negative, a jump will be executed to an address formed by adding the contents of the accumulator to the Program Counter. The new address will replace the previous contents of the Program Counter.

Register affected: Specified memory location ($100_8$ through $107_8$) and Program Counter.

JUMP RETURN, JPR, F = 03

The contents of the Program Counter are stored in location $107_8$. The jump address is placed in the Program Counter and the command stored in that location is executed.

This Function provides a single command linkage to a closed subroutine. Location $107_8$ will contain the address of the next sequential command after the JPR. An indirectly addressed jump to location $107_8$ would place the contents of $107_8$ in the Program Counter and the command immediately following the JPR would be executed.

Register affected: Program Counter and location $107_8$.

STORE PROGRAM COUNTER, SPC, F = 04

The contents of the Program Counter are stored in the memory location referenced by the effective address. The Program Counter will contain the address of the SPC command plus one, at the time the Function is executed.

The Function is used to set an index register with a reference address. It is generally the first command of a program. The following program uses the index register address to form memory addresses relative to the reference address. For example, the following program would execute correctly, regardless of memory location:

| Location | Command | Comment |
|---|---|---|
| X | SPC 0 0101 | Contents of the Program Counter +1, X+1, are stored in index register one. |
| X 1 | LAP 1 0010 | The contents of the memory address referenced by index register one, plus $10_8$, would be placed in the accumulator ($000467_8$). |
| X 2 | ADD 1 0011 | The contents of the memory address referenced by index register one plus $11_8$, would be added to the accumulator ($000577_8$). |
| X 3 | STA 1 0012 | The contents of the accumulator would be stored in the memory location whose address is the contents of index register one plus $12_8$. |
| . | | |
| . | | |
| . | | |
| X 11 | 000467 | |
| X 12 | 000577 | |
| X 13 | | |

Register affected: Specified memory location.

STORE ACCUMULATOR, STA, F=05

The contents of the accumulator are stored in the memory location referenced by the effective address. The effective address is a memory address.

Register affected: Specified memory location.

READ IN, RIN, F=06

Data from the Input/Output Data bus is placed in the accumulator. The effective address is used as the peripheral device code.

Register affected: Accumulator.

OUTPUT, OUT, F=07

The contents of the accumulator are placed on the Input/Output Data bus. The effective address is used as the peripheral device code.

Register affected: None in COMP-18.

## MULTIPLY, DIVIDE, SQUARE ROOT

A hardware multiply, divide, square root, (MDSR) capability may be added to the COMP-18 as an option.

MDSR will perform signed or unsigned multiply and divide and will take the square root of a 36 bit, (assumed positive) value.

The MDSR consists of three 18 bit registers: Product Numerator (PN), Multiplier Quotient (MQ), and Multiplicand Divisor (ID). The use of these registers for each operation is shown in Figure 9.

All operands and results are treated as if they were integers. In double precision numbers, the radix point is considered to follow the least significant bit of the least significant word.

Numbers can be treated as if they were signed or unsigned. Bit 1 of unsigned numbers is treated as the most significant bit. When the numbers are signed, bit 1 is the sign. Negative numbers are handled in 2's complement form. Bit 1 of the least significant half of a double precision number is always treated as a significant bit and not as a sign.

The remainder in division (as well as square root) is always positive. A negative multiplicand or negative divisor is made positive during multiplication or division, respectively. These operands remain in the designated registers, after completion of either operation, in positive form regardless of the original sign.

A divide operation must commence with the absolute value of the PN register being less than the absolute value of the ID register. If such is not the case, the result will be in error.

Operands and functions to be performed are transmitted to the MDSR using the OUT Function. MDSR results are input to the COMP-18 using the RIN Function. All MDSR operations are controlled by the COMP-18 programmed commands.

| | PN | MQ | ID |
|---|---|---|---|
| Multiply | Product (Most significant half of double precision product) | Multiplicand | Multiplier and Product (Least significant half product) |
| Divide | Dividend (Most significant half of double precision dividend) and Remainder | Dividend (Least significant half) and Quotient | Divisor |
| Square Root | Operand (Most significant half of double precision operand) and Remainder | Operand (Least significant half) | Square Root |

MDSR REGISTER OPERATION

FIGURE 9

Operands to be transmitted to the MDSR are placed in the accumulator and an OUT Function, with the appropriate MDSR register device code, is executed. To input the contents of the MDSR register into the accumulator, a RIN Function, with appropriate device code, is executed. The MDSR register device codes are:

| Register | Device Code |
|---|---|
| MQ | 0024 |
| ID | 0025 |
| PN | 0026 |

Each MDSR operation and corresponding function code is shown in Figure 10.

| FUNCTION | CODE | OPERATION |
|---|---|---|
| Square Root | 33 | Computes the 18 bit square root of a 36 bit operand. The Operand is assumed to be positive with the sign bit treated as the most significant bit. |
| Multiply (Unsigned) | 34 | Two 18 bit quantities are multiplied forming a 36 bit unsigned product. |
| Multiply (Signed) | 35 | Two 17 bit signed quantities are multiplied forming a 34 bit signed product. |
| Divide (Unsigned) | 36 | A 36 bit unsigned dividend is divided by a 18 bit unsigned divisor to form an 18 bit unsigned quotient and 18 bit unsigned remainder. |
| Divide (Signed) | 37 | A 34 bit signed dividend is divided by a 17 bit signed divisor to form a 17 bit signed quotient and a 17 bit positive remainder. |

MDSR FUNCTION CODES

FIGURE 10

Typical MDSR timing to perform each function is:

| Function | Cycles* |
|---|---|
| Square Root | 27 |
| Multiply (unsigned) | 9 |
| Multiply (signed) | 11 |
| Divide (unsigned) | 9-1/2 |
| Divide (signed) | 11-1/2 |

* The exact execution time varies for signed and unsigned values.

The following programming example would compute:

$$\frac{X \cdot Y}{Z}$$

and store the result in T and R:

```
LAP   0    X
OUT   0 0024
LAP   0    Y
OUT   0 0025
OUT   0 0034
SLL   0 0006
RIN   0 0025
OUT   0 0024
LAP   0    Z
OUT   0 0025
OUT   0 0036
SLL   0 0007
RIN   0 0024
STA   0    T
RIN   0 0026
STA   0    R
```

The SLL Functions are to delay until the MDSR has completed the operations.

The following programming example would take the square root of Z and store the result in S and R:

```
LAP   0    Z
OUT   0 0026
LAP   6 0000
OUT   0 0024
OUT   0 0033
SLL   0 0022
RIN   0 0025
STA   0    S
RIN   0 0026
STA   0    R
```

# INPUT/OUTPUT

The COMP-18 communicates with all peripheral devices over a common Input/Output Data Bus and Output Address Bus. The data bus is used to transmit 18 bit data quantities either to, or from, the COMP-18 and a peripheral device. The Output Address Bus is used to transmit a device code from the COMP-18 to a peripheral device. The Input Address Bus is used with the Direct Memory Access Controller to input the memory location that would be accessed during the data transfer. Figure 11 shows the COMP-18 communication buses.

## INTERFACE MODULE

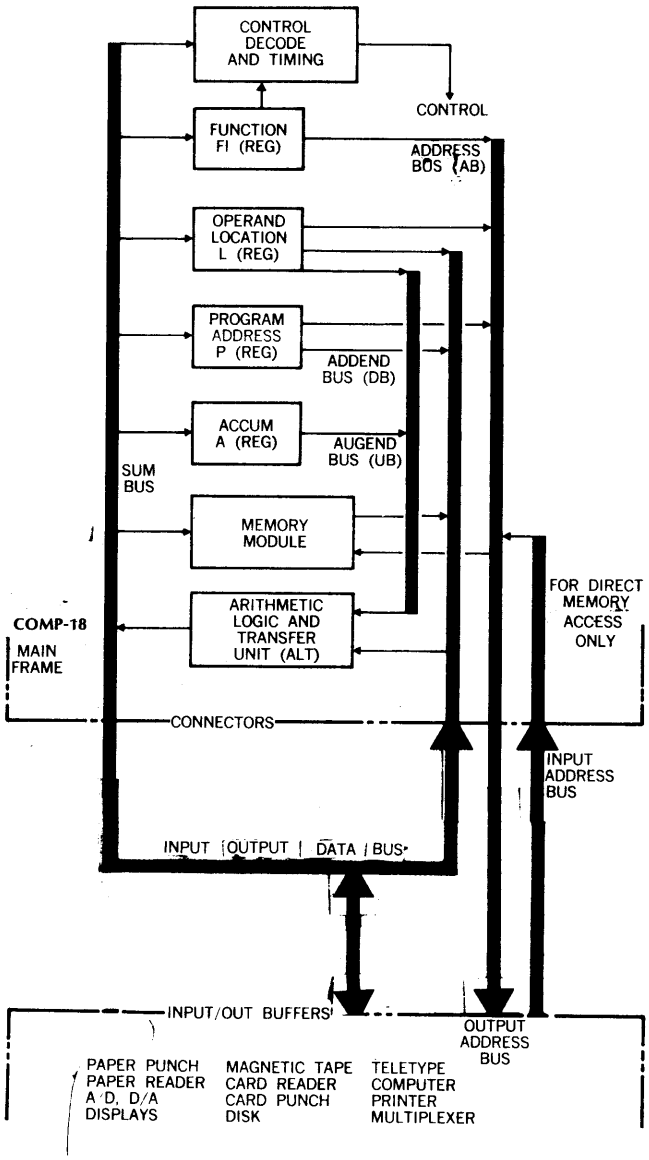
An interface module, called a buffer or controller, is used to match the input/output characteristics of each peripheral device with those of the COMP-18. Additionally, the interface module will respond to unique device codes on the Output Address bus. The form of the response is dependent upon device characteristics and the direction in which data is to be transferred. A device code is generally used to turn on or off the unit, request status, send data, or receive data.

Figure 12 shows the assignment of device codes to peripheral units. Device codes may be any bit length, up to 18 bits. However, since the effective address of a RIN or OUT Function is the device code, it is convenient to limit the code to ten bits and include it in the L Code of the command word.

## DATA TRANSFERS

Data transfers can be accomplished on a word-by-word basis under program control or a block basis. Single word transfers are between the accumulator and the interface module. Block transfers are between memory and a Direct Memory Access (DMA) controller, which is incorporated in the interface module. The DMA makes a memory access by halting the COMP-18 between command executions. This input/output technique is commonly called cycle stealing.

Data transfers between device and accumulator are generally employed when data rates are low or when the COMP-18 can be dedicated to the task of data communication. Direct Memory Access is usually employed when either: data rate is high, length of communication can be specified, data occurrence is random, or the application requires processing concurrent with input/output operations.



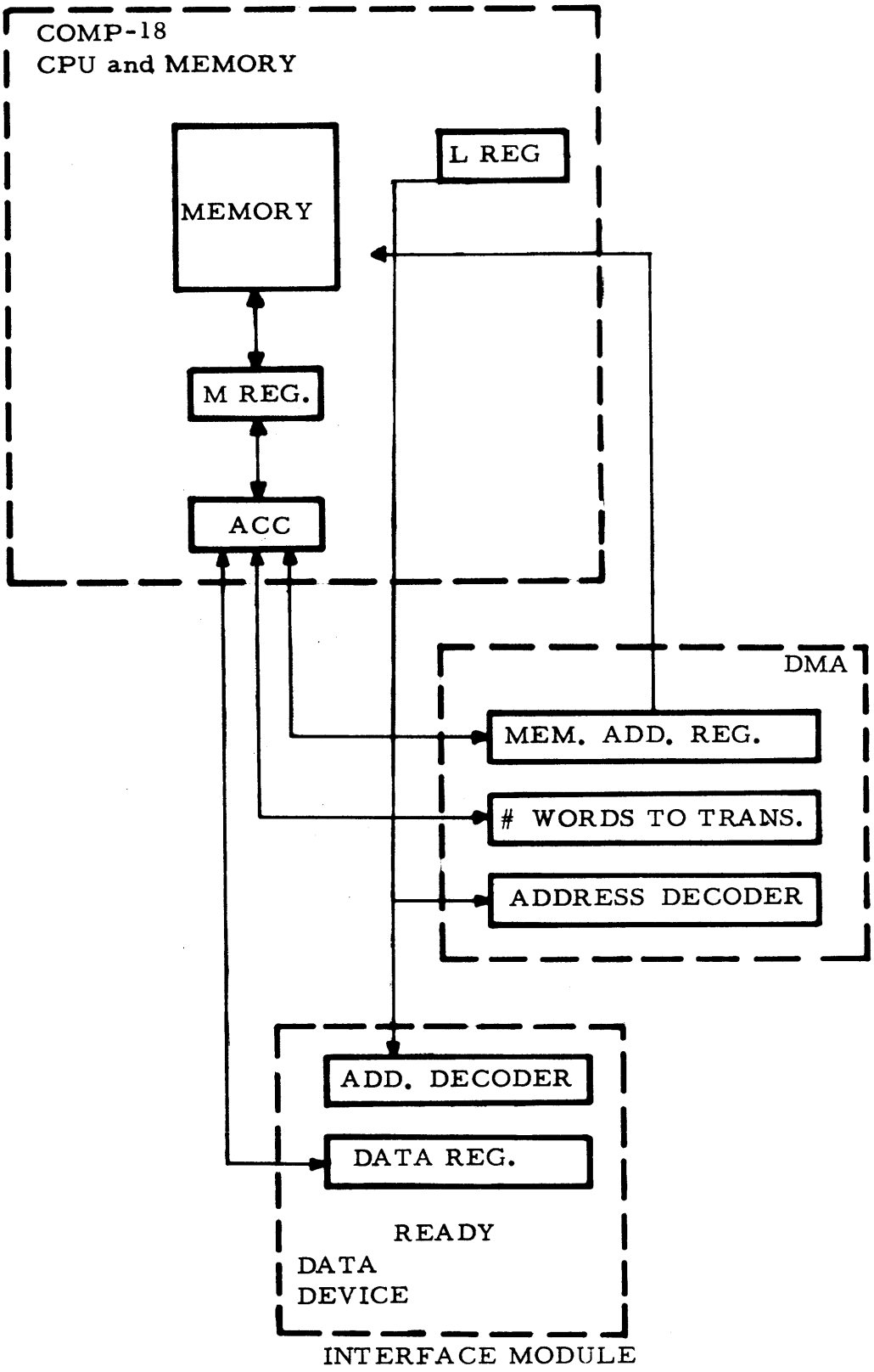

COMP-18 Bus Structure

FIGURE 11

| ADDRESS (Octal) | DEVICE |
|---|---|
| 0000-0003 | RTC |
| 0004-0007 | Special Timing Devices |
| 0010-0017 | Special Devices |
| 0020-0023 | Interrupt |
| 0024-0027 | MDSR Registers |
| 0030-0037 | MDSR Functions |
| 0040-0047 | UniMux Interrupt Status |
| 0050-0057 | UniMux Interrupt Status |
| 0060-0067 | Special Arithmetic Units |
| 0070-0077 | Special Arithmetic Units |
| 0100-0103 | Teletype Console |
| 0104-0107 | Paper Tape Punch |
| 0110-0117 | Card Reader |
| 0120-0127 | Paper Tape Reader |
| 0130-0137 | Printer |
| 0140-0147 | Disc Controller |
| 0150-0157 | Disc Controller |
| 0160-0167 | Magnetic Tapes |
| 0170-0177 | Magnetic Tapes |
| 0200-0207 | Special Devices |
| 0210-0217 | Special Devices |
| 0220-0227 | Special Devices |
| 0230-0237 | Special Devices |
| 0240-0247 | Special Devices |
| 0250-0257 | Special Devices |
| 0260-0267 | Special Devices |
| 0270-0277 | Special Devices |
| 0300-0377 | UniMux Buffer Addressing |

I/O DEVICE CODES

FIGURE 12




INPUT, OUTPUT, DATA FLOW

FIGURE 13

Figure 13 shows the data paths between the COMP-18 registers and memory and the registers of a Word Transfer Interface Module and DMA Interface Module.

A single word data transfer is accomplished using the RIN or OUT command. The RIN is used to read data from a device into the COMP-18 accumulator. The OUT is used to output the contents of the accumulator.

Execution of a RIN command causes the following:

- the contents of the L Register (the effective address) is placed on the Output Address Bus, bit position 1 refers to bus position 1, and so on.

- a control signal is placed on the RIN signal (out) line.

- the contents of the Input/Output Data Bus is read into the accumulator, bit position 1 refers to bus position 1, and so on.

All interface modules receive the address and RIN control signal.

The addressed device responds with data on the Input/Output Data Bus.

Execution of an OUT command causes the following:

- the content of the L Register (the effective address) is placed on the Output Address Bus, bit position 1 refers to bus position 1, and so on.

- a control signal is placed on the OUT signal (out).

- the content of the accumulator is placed on the Input/Output Data Bus, bit position 1 refers to bus position 1, and so on.

All interface modules receive the address and OUT control signal. The addressed device responds by accepting data on the Input/Output Data Bus.

The DMA Controller for a particular device is addressed using the device code and an OUT command. The DMA Controller has two registers. One holds the COMP-18 memory address that will be referenced for the transfer of data. The other holds the number of words that are to be transferred.

The following command sequence would initialize a DMA Controller:

| Command | Comment |
|---|---|
| LAP BUF | BUF is the starting memory address for data transfer. |
| OUT 0 XXXX | XXXX is the device selection code to store BUF in the DMA register. |
| LAP NMWD | NMWD is the number of words to be transferred. |
| OUT 0 ZZZZ | ZZZZ is the device selection code to store NMWD in DMA register. |

Depending upon the logic of the interface module, an additional device code may be required to begin operation of the device for transferring data. No other program commands are required to transfer data or for housekeeping operations during the transfer operations.

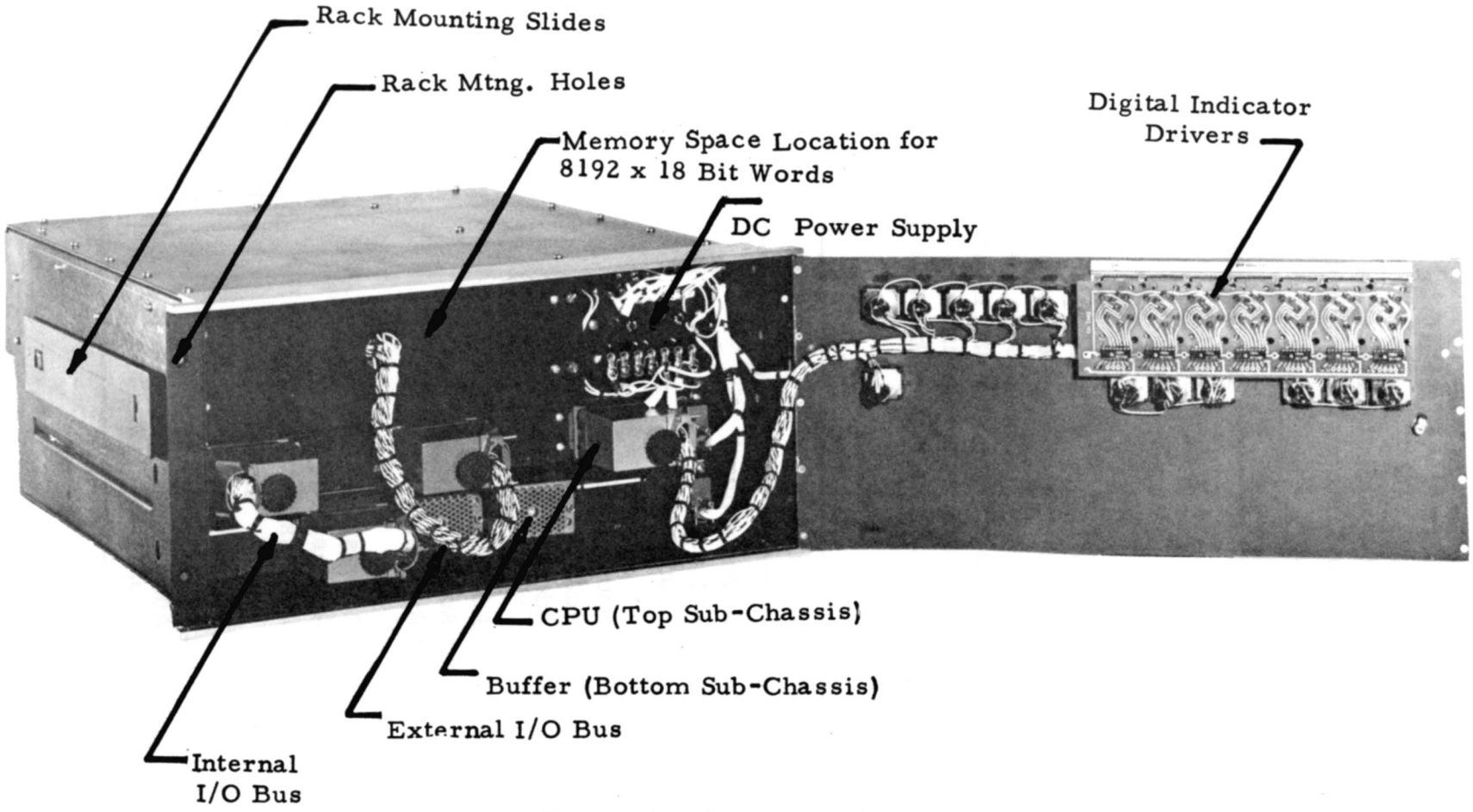
After each data word is transferred, the DMA register containing the memory address is incremented and the register containing the number of transfers is decremented. When the register holding the number of transfers equals zero, transmission halts.

## BASIC I/O CHARACTERISTICS

The basic I/O characteristics are designed for communication between the COMP-18 and buffer controllers housed in the same chassis or immediately adjacent chassis. The COMP-18 pin connector to interface units is located behind the swing out front panel, see Figure 14. The nature of the signals used for such communication limits cable length to approximately six feet.

## PIN CONNECTORS

A single cable connector is available with the COMP-18 main chassis for connecting cable carrying the basic signals. The cable is composed of a twisted pair for each signal. One conductor of each pair is used




Front View (Door Open) Showing
Location of Major Components

FIGURE 14

for an individual ground return for the signal transmitted by the other conductor. However, certain ground returns are grouped by three's and share a common connector pin.

Figure 15 shows the signal name, description, and pin number within the connector.

SIGNAL TIMING

Timing and waveforms associated with execution of the RIN Function are shown in Figure 16. All signals in and out use low true logic levels. The input lines are normally held high by a resistor in the computer, and pulled down to ground to represent a one on any input line. All input line drivers must have open collector drive circuits. Signal levels and impedances are designed to be compatible with TTL and/or DTL circuitry.

| SIGNAL NAME | DESCRIPTION | J1 |
|---|---|---|
| IAB1* | Input Address Bus 1 | A |
| IAB2* | Input Address Bus 2 | B |
| IAB3* | Input Address Bus 3 | C |
| IA RETURN 1-3 | | D |
| IAB4* | Input Address Bus 4 | E |
| IAB5* | Input Address Bus 5 | F |
| IAB6* | Input Address Bus 6 | H |
| IA RETURN 4-6 | | J |
| IAB7* | Input Address Bus 7 | K |
| IAB8* | Input Address Bus 8 | L |
| IAB9* | Input Address Bus 9 | M |
| IA RETURN 7-9 | | N |
| IAB10* | Input Address Bus 10 | P |
| IAB11* | Input Address Bus 11 | R |
| IAB12* | Input Address Bus 12 | S |
| IA RETURN 10-12 | | T |
| Spare | | U |
| IAB13* | Input Address Bus 13 | V |
| IAB14* | Input Address Bus 14 | W |
| IAB15* | Input Address Bus 15 | X |

*Low True Signals

I/O PIN LABEL

FIGURE 15

| SIGNAL NAME | DESCRIPTION | J1 |
|---|---|---|
| IA RETURN 13-15 | | Y |
| IAB16* | Input Address Bus 16 | Z |
| IAB17* | Input Address Bus 17 | AA |
| IAB18* | Input Address Bus 18 | AB |
| IA RETURN 16-18 | | AC |
| OAB1* | Output Address Bus 1 | AD |
| OAB2* | Output Address Bus 2 | AE |
| OAB3* | Output Address Bus 3 | AF |
| OA RETURN 1-3 | | AH |
| OAB4* | Output Address Bus 4 | AJ |
| OAB5* | Output Address Bus 5 | AK |
| OAB6* | Output Address Bus 6 | AL |
| OA RETURN 4-6 | | AM |
| Spare | | AN |
| OAB7* | Output Address Bus 7 | AP |
| OAB8* | Output Address Bus 8 | AR |
| OAB9* | Output Address Bus 9 | AS |
| OA RETURN 7-9 | | AT |
| OAB10* | Output Address Bus 10 | AU |
| OAB11* | Output Address Bus 11 | AV |
| OAB12* | Output Address Bus 12 | AW |
| OA RETURN 10-12 | | AX |
| OAB13* | Output Address Bus 13 | AY |
| OAB14* | Output Address Bus 14 | AZ |
| OAB15* | Output Address Bus 15 | BA |
| OA RETURN 13-15 | | BB |
| OAB16* | Output Address Bus 16 | BC |
| OAB17* | Output Address Bus 17 | BD |
| OAB18* | Output Address Bus 18 | BE |
| OA RETURN 16-18 | | BF |
| Unused | | BH |
| RIN* | Read In Signal (Out) | BJ |
| Return | | BK |
| OUT* | Output Signal (Out) | BL |
| Return | | BM |
| CLOCK* | Clock Signal (Out) | BN |
| Return | | BP |

* Low True Signal

I/O PIN LABEL (Contd)

FIGURE 15

| SIGNAL NAME | DESCRIPTION | J1 |
|---|---|---|
| HOLD ACK* | Hold Acknowledge (Out) | BR |
| Return | | BS |
| HOLD* | Hold Signal (In) | BT |
| Return | | BU |
| INT* | Interrupt Activated | BV |
| Return | | BW |
| INTS* | Interrupt Set (In) | BX |
| Return | | BY |
| READ/WRITE* | Read-Write Signal (Out) | BZ |
| Return | | CA |
| MS* | Memory Strobe (Out) | CB |
| Return | | CC |
| CI | Cycle Initiate (Out) | CD |
| Return | | CE |
| EXR*/W | External Read*/Write (In) | CF |
| Return | | CH |
| Unused | | CJ |
| Unused | | CK |
| EXCI* | External Cycle Initiate (In) | CL |
| Return | | CM |
| Unused | | CN |
| ADDR2* | 3rd Mem. Bank Select (Out) | CP |
| ADDR3* | 4th Mem. Bank Select (Out) | CR |
| ADDR4* | 5th Mem. Bank Select (Out) | CS |
| ADDR RET 2-4 | | CT |
| ADDR5* | 6th Mem. Bank Select (Out) | CU |
| ADDR6* | 7th Mem. Bank Select (Out) | CV |
| ADDR7* | 8th Mem. Bank Select (Out) | CW |
| ADDR RET 5-7 | | CX |
| IODB1* | In/Out Data Bus 1 | CY |
| IODB2* | In/Out Data Bus 2 | CZ |
| IODB3* | In/Out Data Bus 3 | DA |
| IO RETURN 1-3 | | DB |
| IODB4* | In/Out Data Bus 4 | DC |
| IODB5* | In/Out Data Bus 5 | DD |
| IODB6* | In/Out Data Bus 6 | DE |

* Low True Signal

I/O PIN LABEL (Contd)

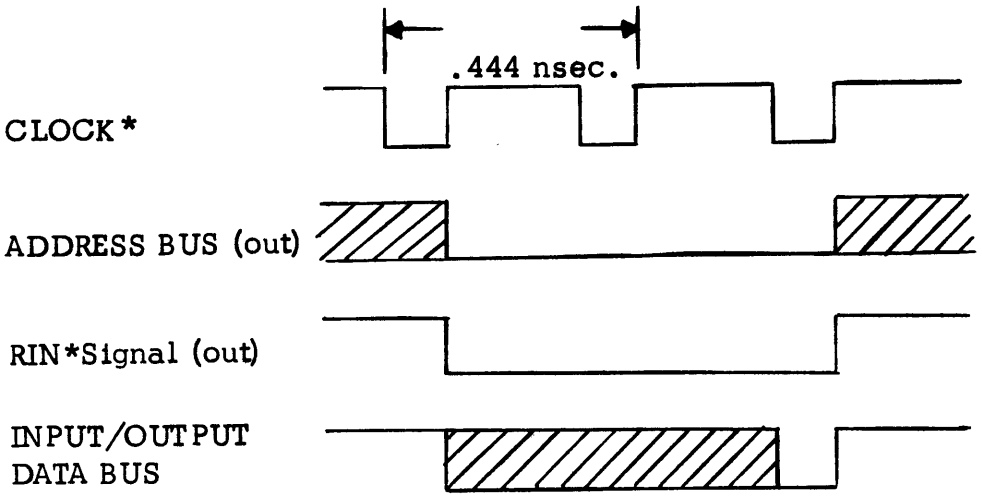
FIGURE 15

| SIGNAL NAME | DESCRIPTION | J1 |
|---|---|---|
| IO RETURN 4-6 | | DF |
| Spare | | DH |
| IODB7* | In/Out Data Bus 7 | DJ |
| IODB8* | In/Out Data Bus 8 | DK |
| IODB9* | In/Out Data Bus 9 | DL |
| IO RETURN 7-9 | | DM |
| IODB10* | In/Out Data Bus 10 | DN |
| IODB11* | In/Out Data Bus 11 | DP |
| IODB12* | In/Out Data Bus 12 | DR |
| IO RETURN 10-12 | | DS |
| IODB13* | In/Out Data Bus 13 | DT |
| IODB14* | In/Out Data Bus 14 | DU |
| IODB15* | In/Out Data Bus 15 | DV |
| IO RETURN 13-15 | | DW |
| IODB16* | In/Out Data Bus 16 | DX |
| IODB17* | In/Out Data Bus 17 | DY |
| IODB18* | In/Out Data Bus 18 | DZ |
| IO RETURN 16-18 | | EA |
| GND | | EB |
| ADDR0 | 1st Mem. Bank Select (Out) | EC |
| ADDR1 | 2nd Mem. Bank Select (Out) | ED |
| K RETURN 4-8 | | EE |
| SMSK* | Set Mask | EF |
| SMRET | | EH |
| RESET* | Reset | EJ |
| RESRET | | EK |
| Spare | | EL |

* Low True Signal

I/O PIN LABEL (Contd)

FIGURE 15




Fall times: 10 to 20 nsec.
Rise times: 15 to 30 nsec.
Input Data Bus Signals must be
gated on and off with RIN Signal.

INPUT WAVEFORMS

FIGURE 16

The timing and waveforms associated with the OUT Function are shown in Figure 17.

DIRECT MEMORY ACCESS SIGNALS

The DMA controller makes use of the RIN and OUT associated communication buses and signals, one other bus and control signals, as follows:

- Input Address Bus
- Read/Write Signal (in)
- External Cycle Initiate (in)
- Hold Signal (in)
- Hold Acknowledge Signal (out)

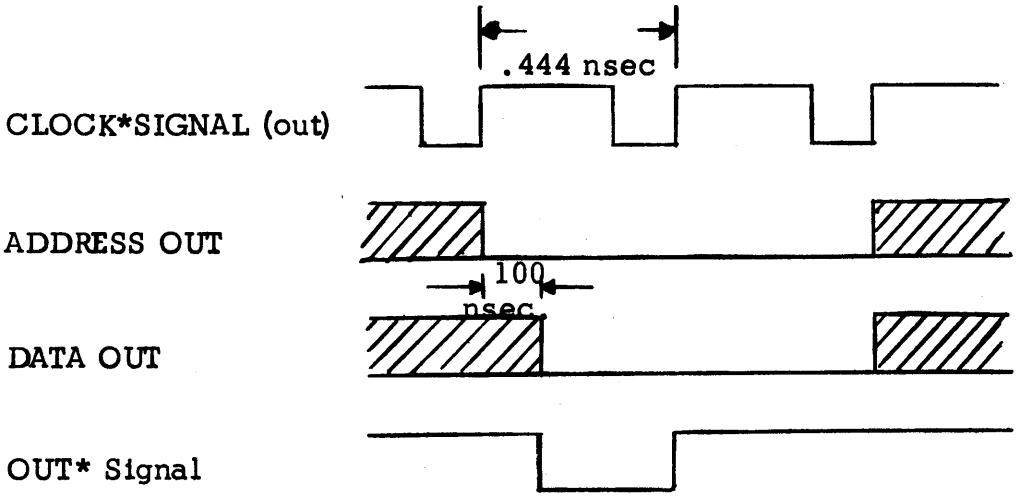
The use of all of the DMA signals are illustrated in Figure 18, and described in the following sequence. When access to the COMP-18 memory is required, the following steps are synchronized with the 2.25 MegaHertz CPU clock:

1) The controller indicates to the CPU that it wishes an access to the memory by pulling down (from +5 volts to ground) on the HOLD* Signal. This signal is driven in parallel by all controllers using DMA and should be driven by an open collector circuit.

2) The CPU will then indicate that it has temporarily stopped operations by lowering the Hold Acknowledge Signal (HOLD ACQ*) from +5 to ground. This indicates that the controller can make access to the memory.

3) As soon as the HOLD ACQ* signal is low, the controller then pulls down on the external Cycle Initiate and on the EXR*/W Signal if the access is to be a read operation. (If the access is a write operation, the EXR*/W Signal is left high). It also supplies the address of the memory location to be accessed on the Input Address Bus with the 1's being low and 0's being high.

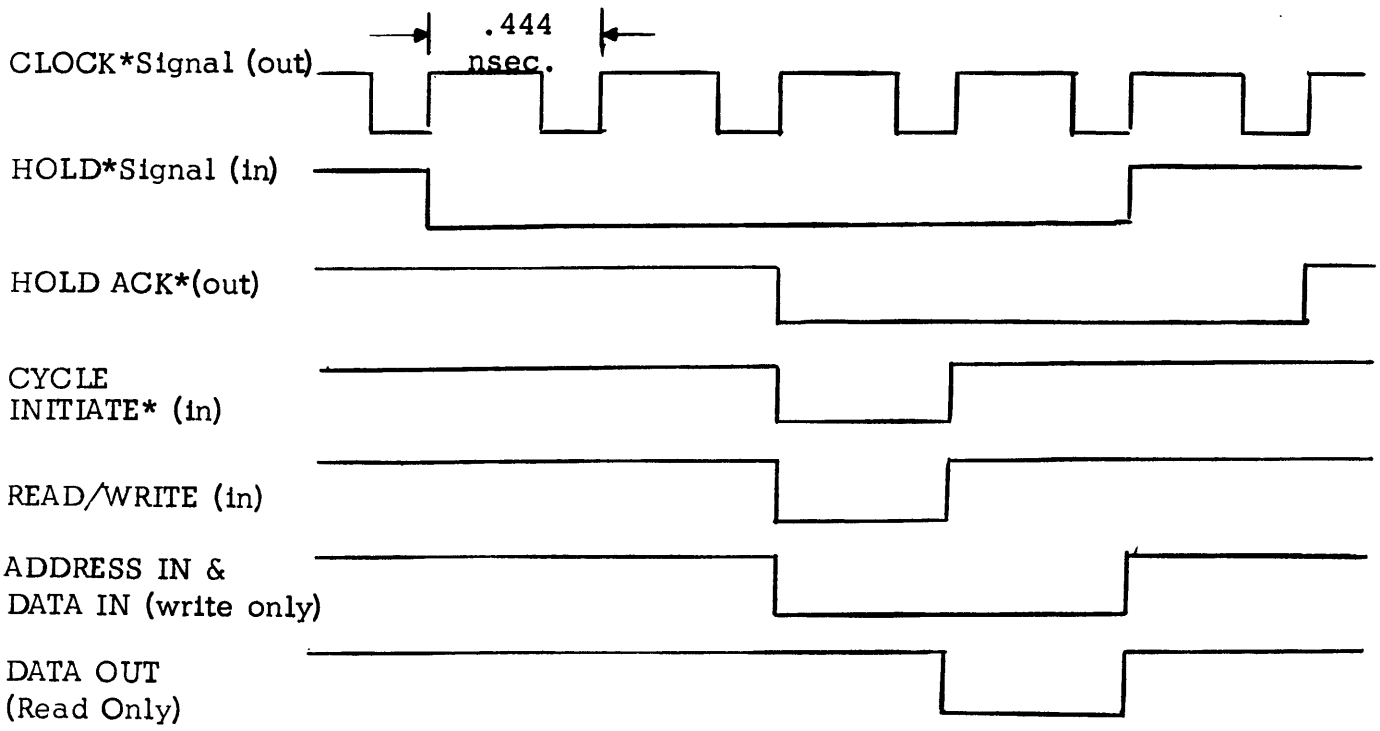
4) The Cycle Initiate and Read*/Write are kept low for one clock period only. HOLD* Signal is held low for one clock period after the HOLD ACQ goes low. The other signals must be held steady for this same clock period.




Fall times: 10 to 20 nsec.
Rise times: 15 to 30 nsec.

OUTPUT WAVEFORMS

FIGURE 17




NOTES: 1) All signals should be controlled by the rise of CLOCK* Signal, and should reach steady state within 50 nanoseconds.

2) Rise time of output signals is 15 to 30 nsec., fall time is 10 to 20 nsec.

3) Signal levels are 0 to+.5v. and 4.4 +0.7 v.

DMA WAVEFORMS

FIGURE 18

5) If the access is a write operation, the word to be written is put on the Input/Output Data Bus, with 1's being low and 0's being high. These signals must be steady for one clock period.

6) If the access is a read operation, the word being read from memory is available on the Input/Output Data Bus during the clock period following the Cycle Initiate, and can be copied by means of the clock signal occurring then.

7) The CPU will continue operations one clock period after the HOLD* is removed.

All input signals to the memory and CPU, driven by the controllers, must be driven by open collector circuits; a resistor pull-up is used on each signal at its destination.

The HOLD ACQ* signal is a "Daisy Chained"priority signal initially supplied by the CPU to indicate that the memory is available for access; however, it is also passed from one DMA controller to the next in such a manner that each one in turn locks out all controllers further down the line until it is done with the memory. In that way, any number of controllers can use DMA without interference among controllers. Thus, each controller receives a HOLD ACQ* (in) and supplies a HOLD ACQ* (out).

## ASR-33 INTERFACE

Communication between the ASR-33 and the COMP-18 is through an interface buffer. The interface buffer holds the character being transferred, responds to the COMP-18 signals, and indicates to the COMP-18, via status bits, the ASR-33 availability for data transmission.

Two device codes are used for COMP-18 ASR-33 interface communication. They are:

| Codes | Meaning |
|---|---|
| 0101 | Place the ASR status bits in accumulator bit position 1 and 2. The code is used only with the RIN Function. |
| 0100 | Transfer data between the interface buffer and bits 11 through 18 of the accumulator. The code is used with either a RIN or OUT Function. |

The ASR status bits indicate when a data transfer may be completed. Bit 1 of the ASR status word must be a one for a character to be input from the ASR. Both bit 1 and 2 of the ASR status word must be zero to output a character to the ASR-33.

The following commands would test ASR-33 status and input one character.

| LOCATION | COMMAND | COMMENT |
|---|---|---|
| 200 | RIN 0 0101 | Read status word. |
| 201 | JAP 0 0200 | If bit 1 is zero, repeat test. |
| 202 | RIN 0 0100 | Read character into accumulator. |

The following commands would test the ASR-33 status word and output one character.

| LOCATION | COMMAND | COMMENT |
|---|---|---|
| 203 | RIN 0 0101 | Read status word. |
| 204 | JNZ 0 0203 | |
| 205 | LAP DATA | When status word is zero, load |
| 206 | OUT 0 0100 | character in accumulator and output. |

In the above example, if an attempt were made to input from the ASR-33, the status word would be set, such that, the program would hang in an endless loop. The following commands test status and also time when the ASR-33 will be ready for the next character. If the status word does not indicate that a character can be outputed from the COMP-18 when the time for transmission has elapsed, then a character is assumed to be waiting for input. A read command is executed and the program returns to output the next character.

| LOCATION | COMMAND | COMMENT |
|---|---|---|
| 207 | LAP DATA | Character to be |
| 210 | OUT 0 0100 | output in accumulator. |
| 211 | LAP CON | Set counter to |
| 212 | STA 0 0104 | time when next |
| 213 | RIN 0 0101 | character can be |
| 214 | JAZ CHAR | output. |
| | | Jump to CHAR to |
| 215 | LAN 6,4 | get next character |
| 216 | INC 4,1 | for output. |
| 217 | RIN 0 0100 ;CHAR | Read character. |
| 220 | JMP NEXT | Return to get next |
| . | | character for output. |
| . | | |
| . | | |
| 230 | -40000 ;CON | |

BOOTSTRAP LOADER

The bootstrap loader is permanently wired into the computer in the form of diodes. The bootstrap will accept a start loading address from the keyboard or paper tape reader. The bootstrap will also accept a start execution address. For a detailed description of bootstrap loader usage, reference the section, "Operation".

Figure 19 contains the bootstrap loader program listing.

```
00000  0400101  SPC 0 101
00001  1060100  LAP 6 100              BOOTSTRAP PROGRAM COMP-18
00002  0500105  STA 0 105              FOR NEW INDEX COMMAND
00003  1160017  LAN 6 017  ;B0
00004  0300021  JPR ANY
00005  0500106  STA 0 106
00006  0300016  JPR THREE  ;B1
00007  3700010  SRE 0 010
00010  0500103  STA 0 103
00011  0300020  JPR FOUR   /JPR THREE FOR COMP-16
00012  1600103  LOR 0 103
00013  0570106  STA 7 106
00014  0260001  INC 6 001
00015  2700006  JMP B1
00016  1160011  LAN 6 011  ;THREE
00017  2700021  JMP ANY
00020  1160014  LAN 6 014  ;FOUR
00021  0500102  STA 0 102  ;ANY
00022  1060000  LAP 6 000
00023  0500104  STA 0 104
00024  0750002  OUT 5 002  ;L1
00025  0650001  RIN 5 001  ;L2
00026  2200025  JAP L2
00027  0650000  RIN 5 000
00030  1460177  AND 6 177
00031  1360052  SUB 6 052         CHECK IF *
00032  2000003  JAZ B0
00033  1260030  ADD 6 030         CHECK IF TAPE
00034  2100037  JNZ L3
00035  0250020  INC 5 020
00036  0750001  OUT 5 001
00037  1360056  SUB 6 056  ;L3    CHECK IF @
00040  2100044  JNZ L4
00041  1160017  LAN 6 017
00042  0300021  JPR ANY
00043  0070104  HLT 7 104
00044  1260010  ADD 6 010  ;L4
00045  2200024  JAP L1
00046  1260010  ADD 6 010
00047  2300024  JAN L1
00050  3720025  SRE 2 025   /SRE 2 023 FOR COMP-16
00051  1600104  LOR 0 104
00052  0500104  STA 0 104
00053  1160031  LAN 6 031
00054  0220003  INC 2 003
00055  1000104  LAP 0 104
00056  2770107  JMP 7 107
```

BOOTSTRAP LOADER LISTING

FIGURE 19

## INTERRUPTS

The hardware interrupts provide a means for a peripheral device to signal the COMP-18 that an external, non-program controlled event has occurred. The event is dependent upon the peripheral device or the application. Some examples of interrupt usage are:

- signal completion of a DMA operation.
- indicate a change of status of a bi-level condition.
- indicate device readiness for data transmission.
- indicate an elapsed time interval.

Interrupts are added to the COMP-18 in modules of 8 interrupt lines. Associated with each interrupt module are two 8 bit registers: Interrupt Mask and Interrupt Status. As additional modules are added, the two registers are correspondingly expanded.

Interrupt lines 1 - 8, 17 - 24, 33 - 40, and 49 - 56 correspond to bit positions 1 - 8. Interrupt lines 9 - 16, 25 - 32,57 - 64 correspond to bit positions 11 - 18. Bit positions 9 and 10 are unused when reading the Interrupt Status or when setting the Interrupt Mask.

### INTERRUPT REGISTERS

The Interrupt Mask register contains an enable/disable bit for each interrupt line. When the interrupt line is enabled, the COMP-18 will respond to the interrupt signal. When the interrupt is disabled, the interrupt will have no effect in the COMP-18. A one enables the interrupt. A zero disables the interrupt.

The Interrupt Mask register is set from the COMP-18 by executing an OUT command. The accumulator is set with the appropriate interrupt enable/disable bit pattern. The device code of the command would be either 020, 021, 022, or 023, reference Figure 12.

When an Interrupt Mask is set, the COMP-16 will not respond to an interrupt until a jump command ($20 \leq F \leq 27$) is obeyed. This delay insures the completion of the interrupt processing program.

Whenever an enabled interrupt occurs, the entire Interrupt Mask is set to zero.

The Interrupt Status register contains a bit position for each interrupt line. Whenever an interrupt line is activated, a one is placed in the corresponding bit position. The setting of the Interrupt Status register is independent of the enable/disable setting in the Interrupt Mask register.

Reading the Interrupt Status register by the COMP-18 resets the register to zero.

The device codes are assigned to the interrupt lines as follows:

| Device Code | Interrupt Lines |
|---|---|
| 0020 | 1 through 16 |
| 0021 | 17 through 32 |
| 0022 | 33 through 48 |
| 0023 | 49 through 64 |

The Interrupt Status register is read into the COMP-18 accumulator by execution of a RIN command. The device code references the group of interrupt lines, as stated above.

Bits 9 and 10 do not correspond to any interrupt lines. These locations are zero after reading Interrupt Status.

## CENTRAL PROCESSOR OPERATION

An interrupt stops the sequential execution of stored program commands and begins executing commands from memory location 0001. The occurrence of an interrupt causes the following:

1) Execution of the current command is completed.
2) The Program Counter is stored in memory location $0100_8$.
3) The Program Counter is set to 0001.
4) Command execution begins starting from memory location 0001.

In general, processing an interrupt requires the execution of a special closed subroutine stored in memory. Location 0001 may be the first executable command of that routine, or the location may contain a jump command to the starting location of the processing subroutine.

Normally, upon completion of interrupt processing, command execution is resumed at the point of interruption. Location $0100_8$ contains the memory address of the next command that would have been executed had the interrupt not occurred. An indirectly addressed jump command would place the contents of $0100_8$ in the Program Counter and, thus, resume command execution from the point of interruption.

## INTERRUPT CHARACTERISTICS

The interrupt cable pin connector for sixteen interrupts is shown in Figure 20. The cable is composed of a twisted pair for each signal.

| SIGNAL NAME | PIN |
|---|---|
| INT1 | A |
| RETURN | B |
| INT 2 | C |
| RETURN | D |
| INT 3 | E |
| RETURN | F |
| INT 4 | H |
| RETURN | J |
| INT 5 | K |
| RETURN | L |
| INT 6 | M |
| RETURN | N |
| INT 7 | P |
| RETURN | R |
| INT 8 | S |
| RETURN | T |
| INT 9 | U |
| RETURN | V |
| INT 10 | W |
| RETURN | X |
| INT 11 | Y |
| RETURN | Z |

INTERRUPT PIN CONNECTOR

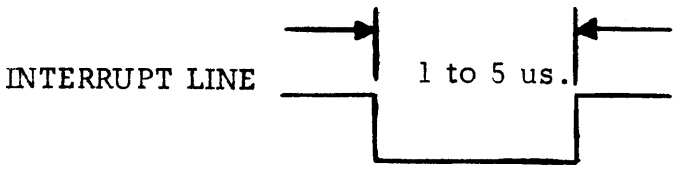
FIGURE 20

| SIGNAL NAME | PIN |
|---|---|
| INT 12 | AA |
| RETURN | AB |
| INT 13 | AC |
| RETURN | AD |
| INT 14 | AE |
| RETURN | AF |
| INT 15 | AH |
| RETURN | AJ |
| INT 16 | AK |
| RETURN | AL |

INTERRUPT PIN CONNECTOR (CONTD)

FIGURE 20

The interrupt waveform is shown in Figure 21. The interrupt line is held high and pulled down for a minimum of one microsecond and a maximum of five microseconds.




INTERRUPT WAVEFORM

FIGURE 21

# OPERATION

The COMP-18 is designed for stand alone, unattended operation in a production area environment. The computer control panel, Figure 22, has been expressly simplified since, in most applications, the primary means of operator interaction is from a special console, typewriter, or teletype.

The control panel provides for displaying registers and various other operations pertinent to program checkout. Since the ASR-33 option is commonly used, its operation is also presented.

The POWER switch is located on the right of the front control panel. It is a two way switch which is lit when power is on.

At the left of the panel is a knob which secures the front panel in a closed position. By unlatching the twist lock knob, the front panel can swing open.

## DISPLAY CONTROLS

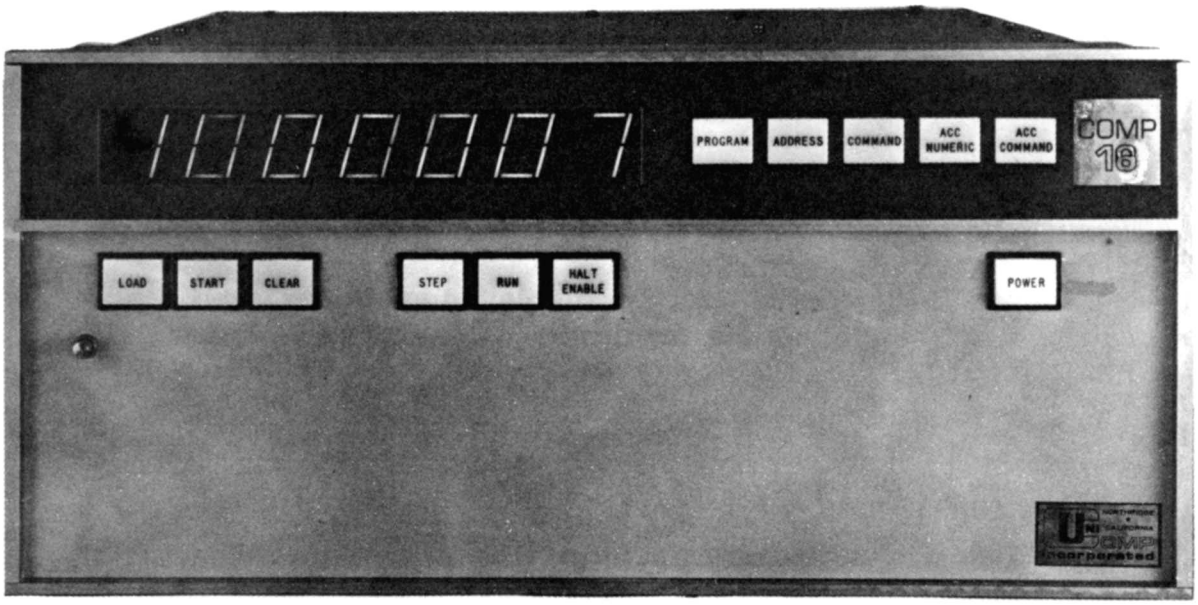
The octal display register can be used to display the Program Counter, L register, accumulator, or command. The five switches next to the display indicators select the time to be displayed. The selection switches are labeled: PROGRAM, ADDRESS, COMMAND, ACC NUMERIC, and ACC COMMAND.

## PROGRAM

The contents of the Program Counter will be displayed in octal format. The Program Counter contains the address of the next command to be executed.

## ADDRESS

The contents of the L register are displayed in octal format. The L register contains the effective address of the last executed command. If the last command were a shift Function, the L register will contain either zero or a negative value (2's complement) one greater than the remaining number of shifts.




Front View of COMP-18

FIGURE 22

COMMAND

The last executed command is displayed in octal command format, as follows:

| $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | octal digits |
|---|---|---|---|---|---|---|---|
| F | | I | L | | | | command codes |

The L code will contain the least significant 10 bits of the effective address.

ACC NUMERIC

The contents of the accumulator are displayed as an unsigned octal number. Negative values are displayed in 2's complement form. The display format is:

| 1 2 3 | 4 5 6 | 7 8 9 | 10 11 12 | 13 14 15 | 16 17 18 | bit positions |
|---|---|---|---|---|---|---|
| $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | octal digits |

The first digit of the display is always zero for numeric format.

ACC COMMAND

The contents of the accumulator are displayed in command word format, as follows:

| 1 2 | 3 4 5 | 6 7 8 | 9 | 10 11 12 | 13 14 15 | 16 17 18 | bit positions |
|---|---|---|---|---|---|---|---|
| $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | octal digits |

PROGRAM SELECTOR CONTROLS

There are three program selector controls: CLEAR, START, and LOAD. Each control causes a specific memory address to be placed in the Program Counter.

LOAD

The LOAD switch causes the Program Counter to be set to an address outside the COMP-18 core memory addresses for execution of the diode stored bootstrap loader.

## CLEAR and START

Execution of the CLEAR and START control will cause the Program Counter to be set with the address 0000 or $02000_8$, respectively. The command held in that memory location can be executed.

In practice the low order $2000_8$ words of memory would be used for indirect addresses, constants, index registers, and housekeeping values. Location $02000_8$ is commonly used as a program origin.

To execute one of the CLEAR or START controls, the RUN switch must be off and the CLEAR, or START, held in while the STEP is pressed. The address will now be in the Program Counter. This can be verified by pressing the PROGRAM display switch and viewing the contents of the display indicators. To begin command execution in the RUN mode, press, in sequence, the STEP and RUN switches.

> The CLEAR and START operations described above cannot be performed if the computer is in the midst of a shift command. To clear up this difficulty, hold the CLEAR switch down while turning the HALT ENABLE and RUN switch on. The computer will complete the shift operation and then halt. The normal CLEAR and START operations may then be performed.

## COMPUTATIONAL CONTROLS

The COMP-18 may be operated in a run mode or idle mode. In the run mode, commands are being continuously executed. In the idle mode, commands are executed singularly in response to a switch action.

There are three computational controls, located on the front panel, which control the mode of operation. They are: RUN, STEP, and HALT ENABLE.

## RUN CONTROL

The RUN control is a two way switch. When the COMP-18 is in the run mode, the RUN indicator is on. Pressing the RUN switch, when in the run mode, will cause the COMP-18 to be in the idle mode. In the idle mode, the STEP indicator is on and the RUN indicator is off. Re-entry to the run mode is accomplished by pushing the RUN switch and STEP switch in sequence.

## STEP CONTROL

When in the idle mode, pressing the STEP switch causes the execution of a single command.

Entry to the idle mode from the RUN mode, is accomplished by pressing the RUN switch.

A malfunction of a Direct Memory Access data transmission which stops COMP-18 operation will cause entry to the idle mode.

HALT ENABLE CONTROL

The HALT ENABLE switch can be turned on and off by pressing the HALT ENABLE switch. When the switch is on, the indicator is lit.

With the COMP-18 in the run mode and the HALT ENABLE switch on, execution of the HLT Function will cause the COMP-18 to halt. The STEP indicator will be on. The RUN indicator will also be on. Pressing the STEP switch will re-enter the run mode. Pressing the RUN switch will enter the idle mode.

If the HALT ENABLE switch is off, the HLT Function will be executed as an unconditional jump with no interruption of processing.

ASR-33

The ASR-33 has controls for the paper tape reader, paper tape punch, and the mode of operation.

The power switch is a three position rotating switch located on the right below the keyboard. In the center position, power is off. Counter clockwise rotation turns power on and connects the Teletype to the COMP-18. The switch position is labeled LINE. Clockwise rotation turns power on and the Teletype operates off-line from the COMP-18. The switch position is labeled LOCAL.

On the paper tape reader, located to the immediate left of the read head, is a three way switch which controls the reader operation. In the middle position, STOP, the reader is off. In the up position, START, the reader is operating. In the down position, FREE, the tape can be physically pulled through the reader gate.

Four single action push button switches, located on top of the paper tape punch unit, control the operation of the punch. The RELEASE button allows the tape to be physically pulled from the punch.

The BACK SPACE button will move the tape back one frame each time the button is pushed. The ON button turns power on. The OFF button turns power off.

## BOOTSTRAP LOADER OPERATION

The following switch setting sequence will cause the bootstrap loader to be executed.

1) Depress the LOAD switch, push the STEP switch, and release the LOAD switch.

2) Push the RUN switch.

3) Push the STEP switch.

The loader is now running and will accept characters from either the ASR-33 paper tape reader or the keyboard. If input is to be from the high speed paper tape reader, then depress the CNTRL key on the ASR-33 keyboard and strike the R key. The bootstrap loader will begin reading paper tape, if tape is present in the reader.

The bootstrap loader will interpret the five octal digits following an * as the starting address for loading binary values. The starting address may be input from the keyboard as follows:

*02000 (CR) (LF)

where the (CR) (LF) indicates striking the Carriage Return and Line Feed keys on the keyboard. The starting address in the example would be $2000_8$. The address must be stated as five digits.

After the loading operation is complete, the loader may be given a starting address for command execution. The starting address is set by typing:

@02000(CR)(LF)

where the (CR) (LF) indicates striking the carriage return and line feed keys on the keyboard. The starting address for command execution would be $2000_8$. The address must be stated as five digits. If the HALT ENABLE switch were on, the COMP-18 would have halted with the address loaded in the Program Counter.

## INSTALLATION

COMP-18 basic system is housed in a chassis 8-3/4 by 22-3/4 by 19 inches. The chassis is suitable for a 19 inch standard relay rack or can be mounted in any special cabinetry as dictated by the application.

Sub-chasses are used to contain the memory, central processor, MDSR, interface elements, and other COMP-18 options. A sub-chassis contains 372, 14 pin dual in-line integrated circuit packages.

Figure 23 presents the various COMP-18 options and their sub-chassis requirements. From Figure 23, the chassis requirements for any COMP-18 configuration can be determined.

Figure 24 presents the physical specifications of the COMP-18.

### MODULE CABLING

All of the individual modules, or sub-chassis, are interconnected by means of cables located just behind the front swing-out panel. If a second chassis is used, the cables are fed between chassis just behind the front panels. Modules may be removed from the chassis by first disconnecting cables tied to the modules. The cables may then be reconnected after the module is removed so that the module may be serviced while operating.

Each chassis can be mounted in a RETMA 19 inch wide relay rack. Two holes on either side of the front (with the front panel swung out) are available for attaching the chassis to the mounting flange of the rack. In addition, two mounting brackets are available on the back of the chassis for attaching to the rear mounting flanges of the rack.

### PERIPHERAL CABLING

Cabling from the chassis to external devices such as peripherals can be done in either of three ways:

1) Through cable connectors on the back panels of each main chassis.

2) Through cables connected to each sub-chassis, and passing up or down through the top or bottom of the chassis just behind the front panel or just in front of the back panel.

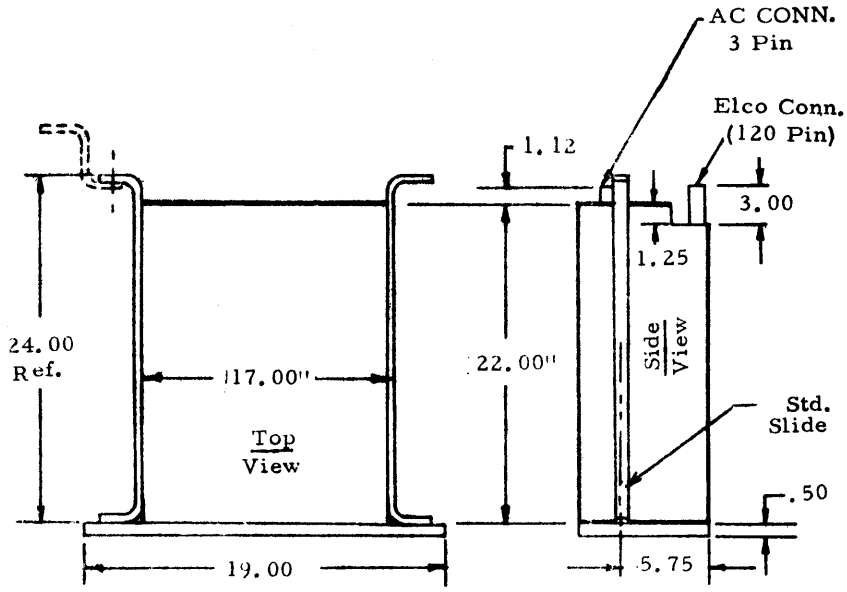
3) Through cables passing through the lower part of the front panel attached to the sub-chassis behind the front panel. This front panel cabling is recommended where access to the back of the relay rack is difficult.

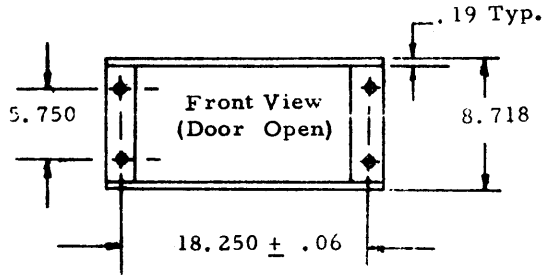
The number, type and size of external cables is dependent upon the external devices connected to the computer; therefore, the external connectors are adaptable to the application.

The interconnection diagram is shown in Figure 25.

Figure 26 shows the physical locations of cable connectors.

| | |
|---|---|
| SIZE | Basic cabinet is 8-3/4 by 22-3/4 by 19 inches and contains, central processor, power supplies, and 4,096 words of memory. |
| WEIGHT | Basic cabinet, not including peripherals, is 48 pounds. |
| LOGIC | Integrated circuit TTL, 5 volt supply. |
| POWER | 115-220 VAC, 50 to 400 cps, 3 wire 250 watts, 5 & -18 VDC. |
| ENVIRONMENT CONDITIONS | Temperature 0 to 140° F, Humidity 10 to 90%. |



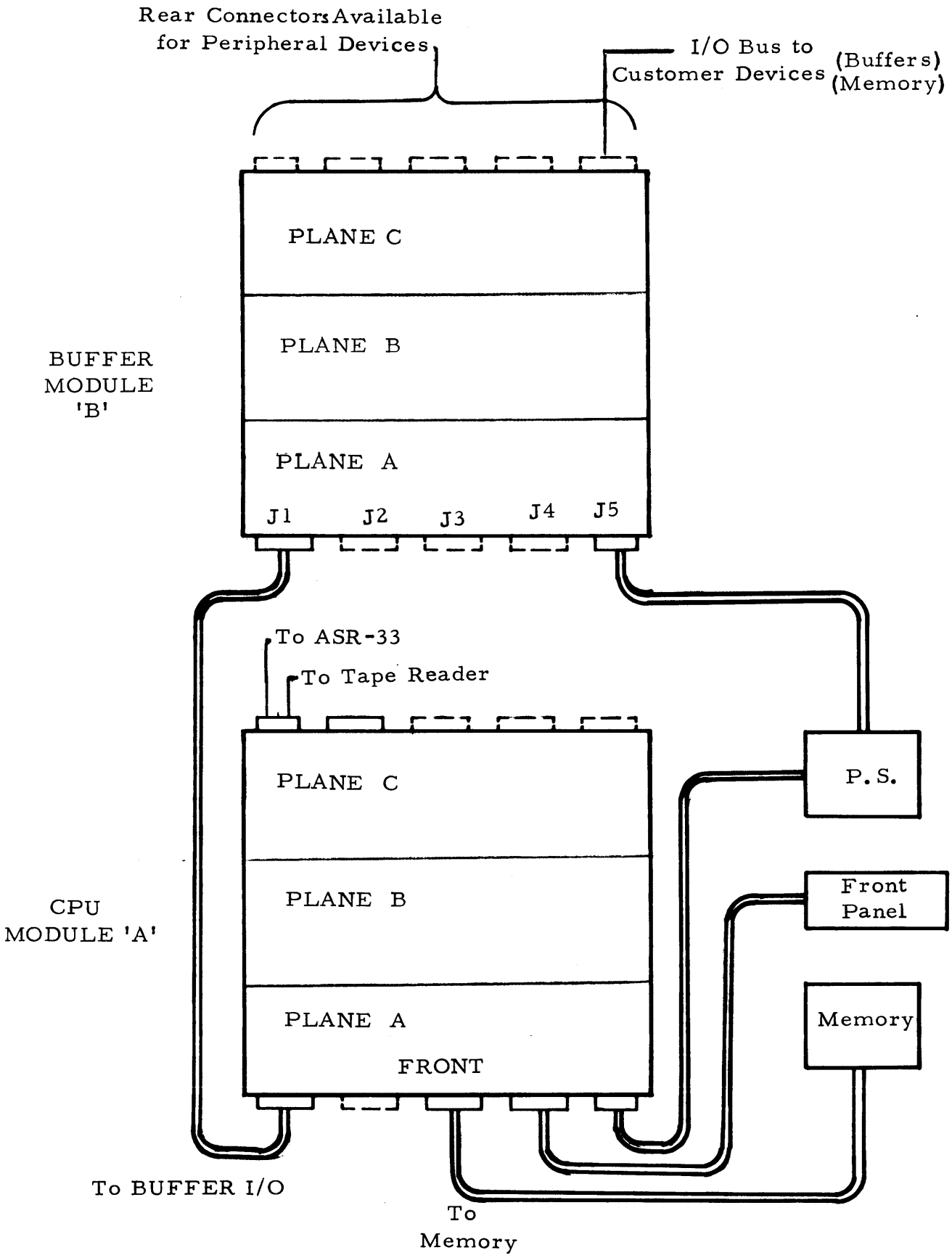




INSTALLATION DRAWING FIGURE 24

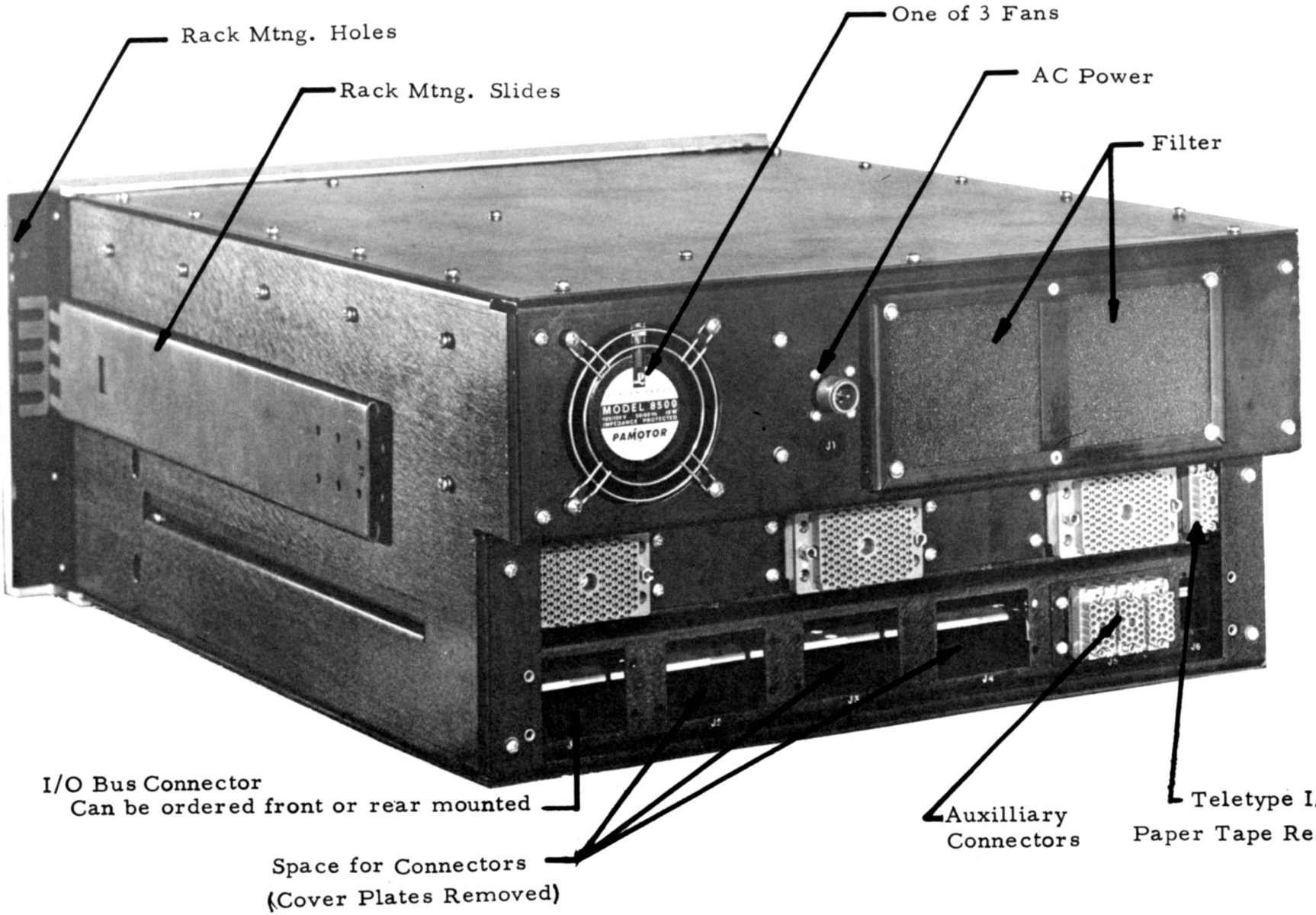
| UNICOMP MODEL NUMBER | UNIT | 1/24 | 1/18 | 1/12 | 1/6 | 1/3 | 1/2 | 1 |
|---|---|---|---|---|---|---|---|---|
| 100 | Memory Module | | | | | | | x |
| 001 | Central Processor (Includes paper tape reader & console Teletype interface | | | | | | | x |
| 020 | Interrupt Module (16 interrupts) | | | x | | | | |
| 002 | MDSR | | | | | | x | |
| 011 | Real Time Clock | | x | | | | | |
| 004 | Power Fail Restart | | x | | | | | |
| 300 or 301 | Magnetic Tape Controller (including DMA) | | | | | x | | |
| 107 | Memory Protect | | | | x | | | |
| 329 | Disc Controller (including DMA) | | | | | x | | |
| 251 | Card Reader Interface | x | | | | | | |
| 261 | Card Punch Interface | x | | | | | | |
| 271 | Line Printer Interface | x | | | | | | |
| 519 | CRT Interface | | | x | | | | |
| 413 | Programmed Teletype I/O Interfaces (Group of 4) | | | | | x | | |
| 431 | Discrete Input Module | x | | | | | | |
| 432 | Discrete Output Module | | x | | | | | |
| 410 | UniMux Controller | | | | | x | | |
| 412 or 413 | UniMux Buffer (1 Channel) | | | x | | | | |
| 430 | Programmed I/O Channel | | x | | | | | |
| 433 | Universal DMA I/O Channel | | | | x | | | |
| 219 | Paper Tape Punch | x | | | | | | |

COMP-18 CHASSIS REQUIREMENTS

FIGURE 23




INTERCONNECTION DIAGRAM
FIGURE 25




Comp-18 Rear View
FIGURE 26