Computer Reference Manual

Lockheed Electronics

MAC 16

Computer Reference Manual

Lockheed Electronics

MAC 16

TM13010009800
January 1970 Third Edition

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

Memoranda

Chapter 1 System Synopsis

INTRODUCTION

This document contains a functional description of the MAC 16 Multi-Application Computer. A general description of the MAC 16 System and available options is followed by details of the MAC 16 processor, instruction repertoire, input-output, and control panel. The MAC 16 System includes a variety of peripheral input-output equipment described in a separate document.

Guide

Accompanying the MAC 16 Computer is a complete software package that includes an assembler Lockheed Electronics Assembly Program, (LEAP), a compiler (FORTRAN IV), Extended Loader (ELOD), and many others. Software documentation is furnished in separate manuals.

Operation of the MAC 16 Computer and details on installation and physical parameters are provided in an Operation and Maintenance Manual that includes text, charts, and logic diagrams.

GENERAL

MAC 16 is a high performance Multi-Application Computer designed to provide fast, efficient, and reliable performance in many application areas. System design consideration has been given towards implementation of data acquisition, instrumentation, communication, process control and automatic test systems. MAC 16 economically performs stored program control functions for systems that ordinarily require extensive special purpose hardware. Because of the MAC 16 speed and multi-level priority interrupt structure, the system is ideally suited for a real-time environment.

System organization of the MAC 16 is shown in figure 1-1. The system consists of a 16-bit parallel processor with a minimum 4,096 word memory, expandable to 65,536 words. High performance is obtained by the one-microsecond-cycle-time magnetic core memories and high speed TTL integrated circuits. The MAC 16 System includes a paper tape reader, punch and a typewriter keyboard. A variety of other peripheral devices are available as options for more complex system requirements.

The MAC 16 Central Processor is packaged on nine printed circuit cards that represent functional units. These cards are located vertically within a card nest that contains 30 printed-circuit card locations. Each 4,096-word memory requires three printed circuit cards. One card occupies two card locations and another, the memory stack board, requires four card locations. By overlapping memory stacks two 4,096 word memories occupy eleven card locations. The remaining ten spaces in the card nest are available for options or special customer applications.

Memory expansion over 8,192 words requires an additional card nest for each 24,576 words of memory.

The MAC 16 card nest is a standard 19-inch unit for rack mounting. The unit is 17-1/2 inches high and 19 inches deep. The MAC 16 power supply is also rack mounted and is 5-1/4 inches high and 19 inches deep. The Control Panel is hinged to the main frame card nest. As an option the Control Panel can be located remotely up to twenty feet from the processor.

Printed circuit cards are two-layer construction. Up to 112 dual-in-line, integrated circuits of 14- or 16-pin configuration can be positioned on one side of the card. The card is 14-1/2 inches by 13 inches.

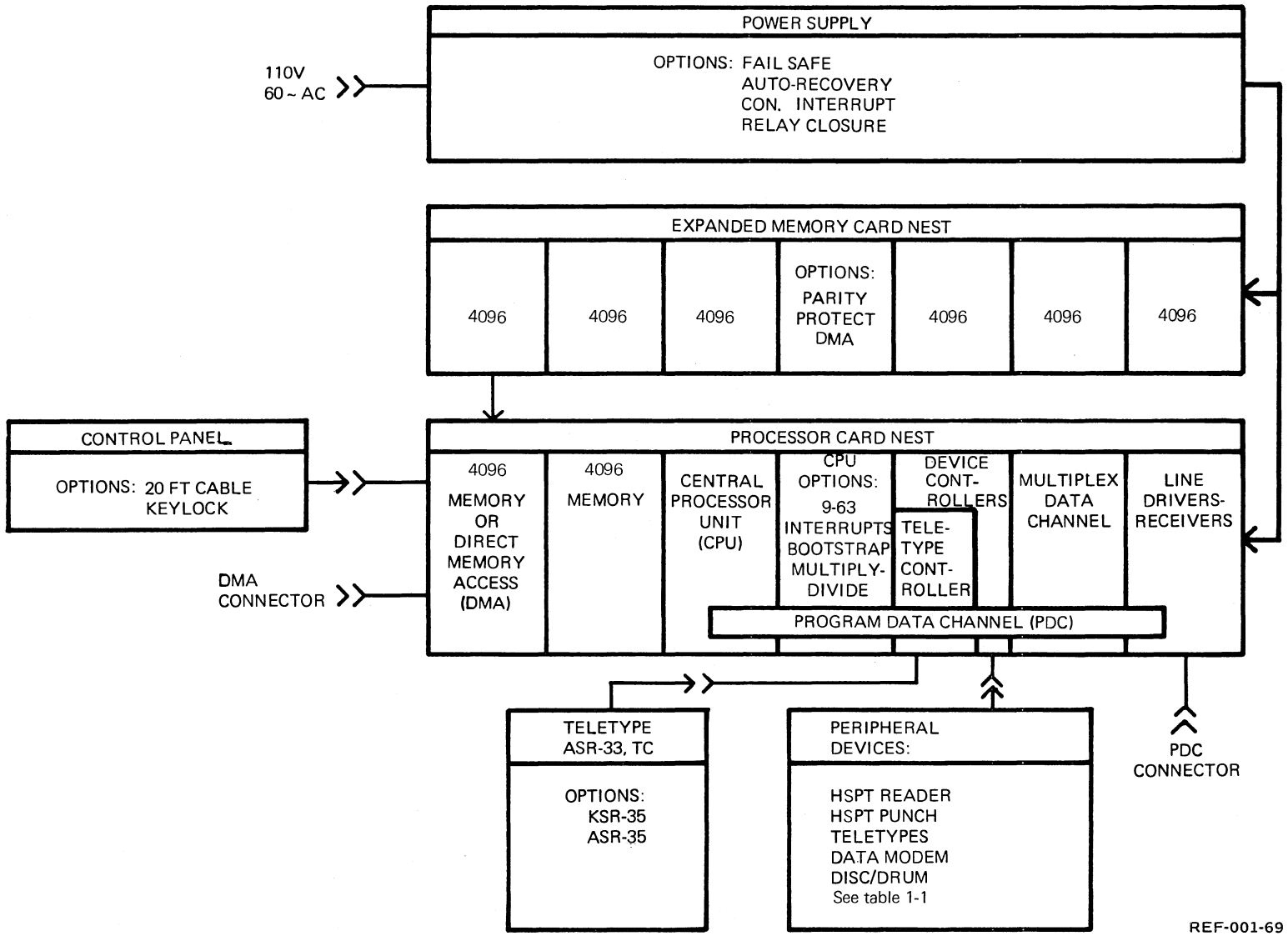


Figure 1-1. MAC 16 Computer System Configuration

CENTRAL PROCESSOR

The MAC 16 Central Processor executes stored instructions at an average rate of 333,000 per second. The longest instruction in the basic MAC 16 is a shift instruction of greater than 12 bit positions requiring five microseconds for execution. (When the hardware multiply-divide option is included, these instructions are executed in ten and thirteen microseconds, respectively.)

The MAC 16 instruction set minimizes storage requirements and provides faster problem solving. Twelve shift instructions allow up to 15 bit positions to be shifted at 250 nanoseconds per bit. Thirteen test instructions permit conditional skip of up to 15 instructions. Byte and word calculation, signed and absolute arithmetic are provided. A carry indicator and two status indicators make double-precision subroutines convenient. Index register load, modification and testing, and memory modification and testing without register alteration are available in the computer instruction repertoire.

Instructions are two types: memory reference and register reference. There are 15 memory reference instructions and 57 register reference instructions providing a total of 72 instructions.

Eight operand addressing modes are provided for each memory reference instruction. These include direct, indirect, and indexed options and combinations of these operations. Particular attention has been accorded the addressing modes to provide effective automatic de-paging with the Extended Loader program. Using this loader, the programmer need not be concerned with paging requirements and can assume that the full 65,536-word memory is available for instruction and data storage.

Register reference instructions provide control of register functions such as shifts, register exchanges and conditional tests. Special register reference instructions allow memory reference indirectly, through an index register. Other register reference instructions allow eight-bit immediate operands.

A unique program interrupt feature is provided in the MAC 16 for handling interrupts and peripheral equipment. The equivalent of this feature is now available only in large-scale, real-time processors. The MAC 16 program interrupt is a multilevel program control that allows up to 64 (optional) programs to be executed in what appears to be simultaneous in real time. These programs are entered in a priority demand basis. Interrupt signals can be set externally or by program control to cause automatic switching of programs. Each program is assigned its own set of Accumulator, Index Register, program status bits and Program Location Register in fixed memory locations. Response to interrupts occurs within one instruction time. Automatic exchange of program registers and status bits occurs within six microseconds.

Processor Options

The basic MAC 16 has four program interrupts. This can be expanded optionally in steps of four up to 64. Sixteen program interrupts may be provided without requiring an additional printed circuit card. Each additional group of 16 interrupts requires an additional printed circuit card.

Processor options also include the addition of hardware multiply and divide. The hardware multiply execution time is ten microseconds and the divide instruction execution time is thirteen microseconds. The hardware multiply and divide instructions are compatible with the software subroutines so that programs are not changed if hardware multiply and divide are included.

A special 64-word diode memory that contains a copy of the bootstrap loader program can be added to the Central Processor. Pressing a Control Panel switch causes this program to be entered automatically into memory and executed. The program reads an object program that is assembled in the bootstrap format into the computer memory. The program may be entered optionally for execution.

MEMORY

The MAC 16 memory is a standard, field-proven Lockheed Electronics Company product consisting of miniature ferrite cores that provide a cycle time of less than one microsecond. Memory expansion up to 65,536 words is possible with add-on units before the reach of the 16-bit address is exceeded. This modular growth occurs in 4,096-word increments.

The processor card nest can contain two 4,096 (8,192) word memories. The expansion option for over 8,192 words, has the memory interface logic added to the processor and a cable to interconnect the additional nests.

Memory Options

Memory options include parity and protect. MAC 16 memory modules with the parity option are 18-bit word

memories. A parity bit is provided for each eight-bit byte. Parity error detection generates a signal that can be used to cause a program interrupt.

The memory protection option is provided with manual switches for each 4,096-word module. Once loaded, these modules can be placed in a read-only mode. Attempts to modify a protected memory module cause generation of an error signal that can be used as a program interrupt signal. The memory options are described in the Programmers Manual.

CONTROL PANEL

The MAC 16 Control Panel affords functions for program test and maintenance operations. Processor registers can be selectively displayed and loaded. Memory words can be modified or displayed simultaneously with word addresses. Instruction single step and address halt control are included. The 16 data entry switches function as sense switches during program execution.

The Control Panel normally is hinged to the processor card nest, allowing access to the card nest wiring for maintenance purposes.

Control Panel Options

A twenty foot cable can be attached to the control panel for remote operations or for a location in the system cabinet other than by hinging to the processor.

A keylock can be added to the control panel. This lock, when on, disables the control panel switch functions. The computer can be locked either on or off.

POWER SUPPLIES

The earlier series of MAC 16 have a type PL power supply but the later series have a type PA. Both types are described in the following paragraphs.

Type PL Power Supply

The type PL power supply receives single-phase 103 to 132 vac input power at line frequency from 45 to 440 Hz. The type PL has one pluggable option card for line failure detection. The line failure detector generates a signal that can be used as high priority interrupt to the processor. This signal causes automatic storage of the content of the current program registers and execution of a special user power failure program.

Type PA Power Supply

The type PA power supply receives either 103 to 127 vac or 206 to 250 vac input power at line frequency from 47 to 63 Hz. All d-c outputs are closely regulated, protected against overvoltage and reverse voltage and overload conditions. D-c output includes +5 volts at 21 amperes, -6 volts at 3 amperes, +18 volts at 4.2 amperes, and a resistance-programmed output of 21 to 32 volts at 10 amperes. This power output is more than adequate to service the MAC 16 with processor options and expanded memory up to 24,576 words.

The type PA has one of two pluggable card options; line failure option identical to the type PL line failure or the power detector option. The power detector option includes line failure, restart, line frequency pulse detector, and relay contact closure. The restart provides automatic run, high priority interrupt and clears the computer. The line frequency pulse detector generates a line frequency signal which is used as a real time interrupt signal. A special program at the selected interrupt level, counts these pulses to record elapsed time. The relay contact closure option provides a signal when the loss of either the -6, +5, or 21 to 32 vdc outputs occur. The loss can be due to an internal power supply malfunction, external over-voltage, or a short. This closure can be used to initiate an alarm.

INPUT-OUTPUT

The basic MAC 16 includes a set of data lines, address lines, and command lines that provide parallel communication between the processor and device controllers. The communication occurs between the accumulator and the data register of the selected device controller. This communication is controlled by the execution of input-output data and command instructions. The set of communications lines is called the Programmed Data Channel, (PDC).

Programmed Data Channel

The PDC has a set of eight address lines used for selecting up to 255 device controllers. The address lines are driven by eight bits of the input-output instruction in

parallel with 16 bits of the accumulator. The accumulator contains pre-loaded commands or data on output or receives status or data on input.

Entry into input-output programs can be made automatic by wiring device controller service requests signals to an interrupt. This system configuration conveniently interleaves programmed I/O operations with normal processing. Response to service requests occurs at the data rate of the device. Each device controller can be provided with a separate interrupt for its service request, or all device service requests can be shared on one interrupt. In the latter case, it is necessary to use special I/O instructions to interrogate the device controller address of the highest priority that is requesting service.

System input-output data rates via the Programmed Data Channel depend upon the amount of processing that occurs in each input-output program. The typical high data rate under program control is 60,000 16-bit words a second. Device controllers can be located externally to the processor card nest, in which instance line driver-receiver printed circuit cards are necessary to extend the Programmed Data Channel. The MAC 16 System offers two options for this purpose. One option is suitable for line lengths up to 50 feet with restrictive loading, the other furnishes line drive for up to 100 feet of cable. Both options have cable connectors on the base of the processor card nest.

Input-Output Options

Higher performance input-output channels are available with MAC-16 systems. These are the Multiplex Data Channel (MDC), the Direct Memory Access (DMA) Channel, the Memory Interleave Channel (MIC), and the Selector Data Channel (SDC).

For more detailed descriptions of the I/O options see the Programmers Manual.

Multiplex Data Channel

The MDC can communicate with up to 16 device controllers independent of programmed instructions and is capable of transferring up to 333,000 words a second to or from the MAC 16 memory. This communication occurs between processor execution of instructions and shares the PDC lines. The MDC uses a set of two control words for each device controller. These control words specify memory address of data, block length and the type of input-output operation. I/O can be in bursts of several words or by single word. A word can be an 8-bit byte or a word of 16 bits. In the case of byte operation, the MDC automatically packs or unpacks a 16-bit word.

The programmer is required to initialize MDC control words. When a block length control word is counted down to zero, the MDC sends an interrupt signal to the processor. An interrupt is required for each device controller under MDC control. The device controllers are connected to the PDC, but their service requests are connected to the MDC. When the MDC detects one of these service requests, it halts the processor and electrically simulates input-output instructions. This simulation offers a capability that means the device controllers can be controlled by either the PDC or MDC without design changes.

The MDC communicates with the system memory by controlling the processor Memory Data and Memory Address Registers.

Direct Memory Access Channel

The Direct Memory Access, (DMA), option is a device that allows a device controller to communicate directly with the system memory over its own set of memory address, data and control lines. The DMA is a two-port switch that allows the special device controller to share memory with the central processor on a "cycle stealing" basis. The DMA provides data communication rates up to 1,000,000 words a second.

Memory Interleave Channel.

The MIC is basic to the CPU and provides the user the internal timing and control signals to implement a unique data controller and channel. The MDC and the hardware Multiply-Divide option are LEC designs that use the MIC capability. The MIC is capable of interleaving its execution between CPU instructions.

Selector Data Channel

The SDC is an optional controller that independently controls data communication between up to four device controllers and memory concurrent with processor operation through the DMA channel. It minimizes interference with the CPU (one memory cycle per data transfer) and accommodates data transfer rates up to 333,000 per second with continuing program execution. The interface of a

device controller to the PDC and SDC busses is identical; thus data transmission with a device controller designed for the PDC can be attached to the SDC.

PERIPHERAL DEVICES

The MAC 16 includes a modified Model ASR-33 Teletype TC and a device controller, which enables the computer to turn the Teletype paper tape reader on or off under program control. The controller is designed for duplex operation with the paper tape reader or keyboard and paper tape punch.

Character codes are the non-parity USASCII set where the 7 least significant bits accommodate the basic 7-bit ASCII code, the most significant bit always being a ONE. The Teletype-generated perforated tape complies with EIA Standard RS-227 for one inch, eight-level tape.

Peripheral Options

Special MAC 16 options make it possible for the Computer to communicate with other peripheral options. Table 1-1 is a list of standard peripheral options.

SOFTWARE SYSTEM

The software package for the MAC 16 is designed to work as a system. This is accomplished by having all programs interface with peripheral devices, through common control functions, executives, and drivers. Thus, peripheral devices may be changed without changing the system programs. Only the peripheral device driver programs need be changed. System programs are selected by the Elementary Executive Program, ELEX, communicating with the

Table 1-1. Standard Peripheral Options

Peripheral	Description
Teletype ASR33/35	Reader, punch keyboard and printer; 10 characters/sec
Teletype KSR33/35	Keyboard and printer; 10 characters/sec
High Speed Paper Tape Reader	300 characters/sec
High Speed Paper Tape Punch	60 characters/sec
Card Reader	300 cards/min
Line Printer	350 lines/min
External Input Register	32 bits
External Output Register	32 bits
Magnetic Drum	32K words to 2147K words
Magnetic Tape Units	9 channel, IBM compatible, 25 ips, 800 bpi 7 channel, IBM compatible, 25 ips, 556/800 bpi
Asynchronous Data Modem	
Synchronous Data Modem	
Real Time Clock	

peripherals through the Input-Output Executive, IOEX. (See Programmer's Manual.)

Loaders

Initial computer loading is accomplished by the Bootstrap Loader (BLOD), a 64-word program on paper tape that is loaded with a minimum loader, the Hand-Loaded Bootstrap (HLBT). The Bootstrap Loader also is offered on a diode memory as an option, and can be loaded by pressing the BT switch on the control panel. BLOD loads programs that have been assembled in the bootstrap format. This format provides starting memory address for storage and start of the loaded program. The Bootstrap Loader includes a checksum evaluation.

The MAC 16 Extended Loader (ELOD) provides the capability of loading both absolute and relocatable programs generated in the extended mode by LEAP, the assembly program. The Extended Loader automatically depages inter-page addreses, placing the address links in the appropriate base page, allocates common storage space, and links all external references (typically, subroutine calls). At the completion of loading, a memory map can be printed showing linkage space used in the base pages, the address of all programs by name, the common base, and the starting address for execution. A Loader option generates a bootstrap-mode tape of the program instead of loading the program into core memory.

Assemblers

The Lockheed Electronics Assembly Program (LEAP) is a two-pass assembler with many advanced features:

- o predefined MACROS can be used
- o external references to both data and subroutines
- o automatic depaging control
- o selective assembly of program modules
- o pseudo-operations
- o expressions with full range of operations
- o symbol table dump

Compiler

FORTRAN IV, a one pass compiler, accompanies MAC 16 systems with 8,192 word memory or greater, and performs all the functions defined by USASI document number X3.9-1966.

A full package of standard intrinsic and external functions complements the compiler.

User Libraries

A collection of frequently used routines is provided to assist the programmer in development of a program. These include:

Fixed Point Arithmetic, Mathematics:
- single and double precision multiply and divide
- double precision add and subtract
- single precision trigonometic, square root,
- exponential (e, 10, 2), logarithms (e, 10, 2)
- integer to real conversion (single and double)

Floating Point Arithmetic, Mathematics
- single and double word floating point arithmetic
- single and double word floating point FORTRAN IV
- external functions
- real-to-integer conversion (single and double)

Complex Arithmetic, Mathematics
- complex arithmetic
- complex FORTRAN IV external functions

FORTRAN IV Intrinsic Functions

- full complement of standard FORTRAN IV intrinsic functions

Programmer Aids

The Tape Editor (EDIT) program operates under ELEX, assisting the programmer in updating or correcting a symbolic tape. Among the functions available with the editor are
- o deletion of a group of records
- o insertion of one or more records
- o overlay of one record

The Debug Program (DBUG) gives the programmer communication with the computer to selectively control a program being tested. Register and/or memory dumps can be specified, and changes can be made to selected memory locations. Full and partial program trace selection is included.

General Machine Tests

General Machine tests programs are available for test of all logical functions of the computer, memory operation,

and peripheral performance. These programs provide a fast means of isolating operator or program errors from suspected machine malfunctions.

MAC 16 COMPUTER SPECIFICATIONS

WORD SIZE

16 bits

CYCLE TIME

1 microsecond

MEMORY

Basic 4,096 words (expandable to 65,536 words in 4K increments)

ADDRESSING

512 word pages
Fixed and Floating base page
Up to 8 base pages
16-bit index register for each program level
Indirect addressing, 16 bit address
Bi-level indirect indexed addressing
Direct or Indirect addressing

REGISTERS

Six 16-bit registers: Accumulator, Operand, Program Location, Memory Address, Memory Word, Instruction Word
16-bit parallel full adder
Interrupt Level Register, 4 to 64 bits

INTERRUPTS

Multi-level priority interrupts
Automatic register store and restore on interrupt
Four basic interrupt levels
Interrupt response time is 6 microseconds
Complete program control of interrupts
Automatic program change
Expandable to 8, 16, 20, 24, 32, 36, 40, 48, 52, 56 or 64 program interrupt levels (option)

CONTROL PANEL

16 program sense switches
Single instruction execution
Halt on address switch
Register display select
Memory read or load with automatic address increment
Carry, Overflow, Inhibit Interrupt, and Base Page indicators
Two programmable sense indicators
Remote panel (option)
Key protect (option)

INPUT/OUTPUT

16 data bits per channel
Up to 255 devices directly addressable
Programmed Data Channel rate: to 60 kHz words, typical
Multiplexed Data Channel rate: to 333 kHz (words or bytes in burst mode) (option)
Direct Memory Channel rate: to 1 MHz words or bytes (option)

PROCESSOR OPTIONS

Multiply and Divide instructions
Power Fail Safe
Memory Parity
Memory Protect
Automatic Bootstrap Loader
Expanded Interrupts
Expanded Memory

BASIC I/O

ASR 33 Teletype
ASR 35 Teletype (option)
KSR 35 Teletype (keyboard and printer only)
High Speed Paper Tape Reader, 330 cps
High Speed Paper Tape Punch, 60 cps

PHYSICAL

Main frame dimensions: 17-1/2" x 19" x 19"
Weight: less than 64 lbs.

Power Supplies Specifications:

Type PL:

Dimensions: 5-1/4 x 19 x 23 inches
Weight: 75 lbs.
Input Requirements: 105 to 132 vac, single-phase, 45-440 Hz

Type PA:

Dimensions: 5-1/4 x 19 x 19 inches
Weight: 60 lbs
Input Requirements: 103 to 127 or 206 to 250 vac, 47 to 63 Hz.

ENVIRONMENTAL OPERATING CONDITIONS:

Temperature	– Operating: $0^{o}C$ to $50^{o}C$, 100 CFM air Storage: $-55^{o}C$ to $+80^{o}C$
Humidity	– Operating and Storage: To 90% RH without condensation
Shock and Vibration	– Withstands normal commercial shipping shock and vibration when packaged to specification.

Memoranda

Chapter 2

Computer Organization

CENTRAL PROCESSOR ORGANIZATION

The MAC 16 Computer is a general purpose, parallel processor that operates on binary data at a four megacycle rate. The computer organization consists of a central processor unit (CPU) with an expandable core memory, a control panel and input-output lines called the Programmed Data Channel. These units are shown in figure 2-1.

The CPU contains seven registers interconnected with two 16-bit parallel buses and a Function Generator. The content of any register can be transferred to any other register via the Operand Bus, through the Function Generator, to the Function Bus and into the receiving register. Transfer occurs under computer control in 250 nanoseconds. Register reference instructions allow the programmer to process the content of several registers.

Function Generator

The Function Generator, F, can perform several parallel functions as data is transferred from the Operand Bus. The Function Generator can add to the data the content of the Operand Register, shift left, or shift right, and also move the operand byte from right to left. Two special flip-flops and indicators are associated with the Function Generator: the Carry flip-flop (C indicator) and the overflow flip-flop (V indicator). The carry facilitates double length arithmetic and other special programs. It is set on most shift instructions with the last bit shifted out and by a carry out of the most significant bit position on add or subtract instructions. The Carry also can be set, reset, or tested by special instructions.

The Overflow is set on arithmetic overflow of the Accumulator. If an add or subtract instruction causes a result to exceed either a value equal to $(+1.0\text{-}2^{-15})$ or -1.0 the Overflow is set to one. In MAC 16 the largest Accumulator word is bit S=0 bits 1 through 15=1's (0111⟶ 11). The largest negative number is bit S=1, bits 1 through 15=0's (1000⟶ 00). If an arithmetic left shift causes a magnitude bit to be shifted out of the Accumulator the Overflow Indicator is set to one.

Operand Register

The 16-bit Operand Register, W, is used to hold temporarily, one of the operand inputs to the Function Generator.

Accumulator

MAC 16 contains a 16-bit Accumulator called the A Register. The A Register receives the results of many instructions and is used to communicate with devices on the Programmed Data Channel. A copy of the A Register is located in memory for each program level.

Program Location Register

The Program Location Register, P, is a 16-bit register holding the address of the next instruction in sequence. It is incremented during each instruction and is modifiable under program control with jump or skip instructions. The content of the P Register is automatically stored in memory each time a program interrupt signal causes a program level change. The higher priority interrupting program location is fetched from memory and entered into the P Register.

Memory Address Register

The Memory Address Register, M, is a 16-bit register that holds the current address of the memory word when

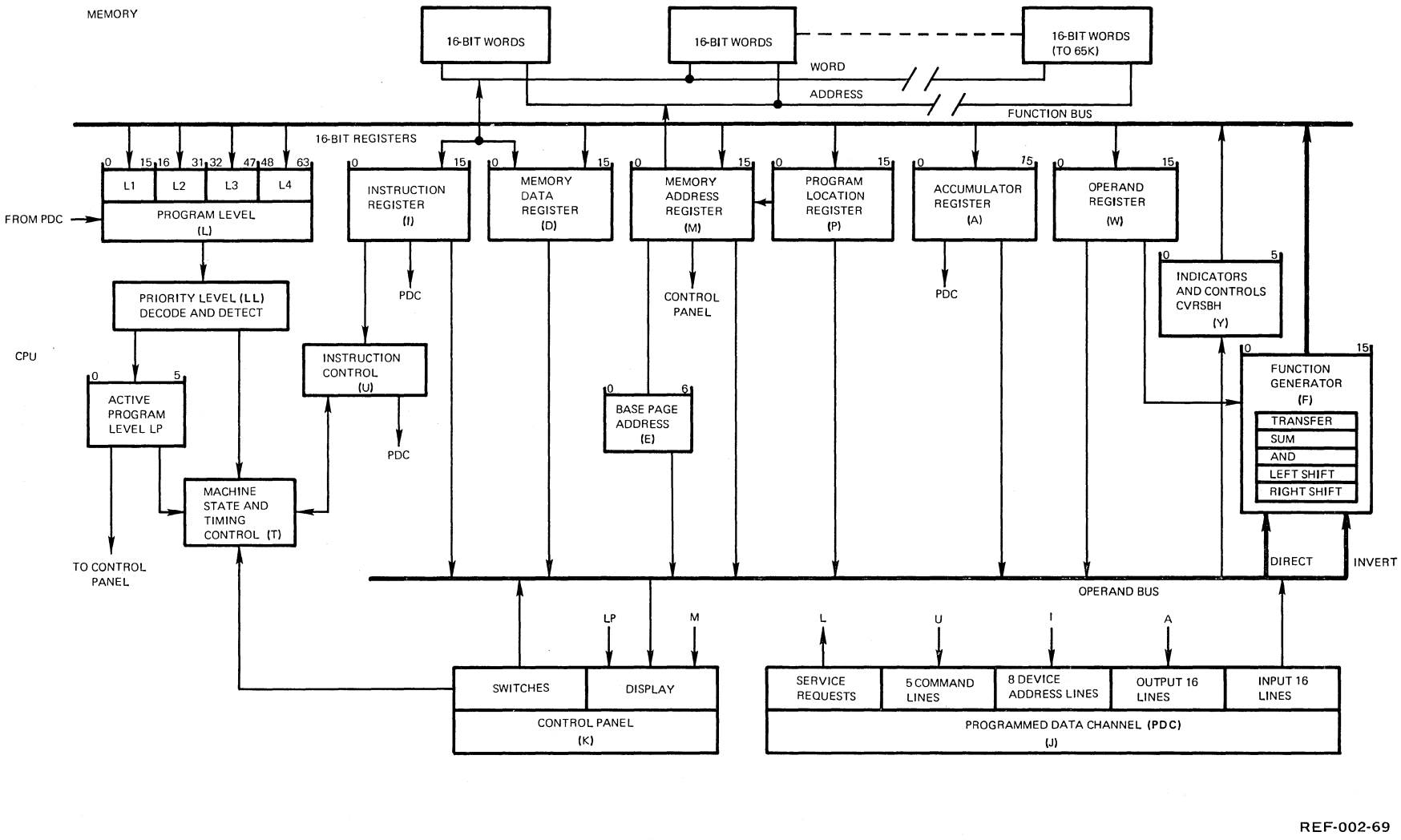


Figure 2-1. MAC 16 Processor Block Diagram

the computer executes a memory cycle. A 16-bit Memory Address Register permits memory expansion up to 65,536 words.

Memory Data Register

The Memory Data Register, D, is a 16-bit register that receives inputs directly from the sense amplifiers on memory read operations, and provides input data for memory write operation. The D Register input and output are available for memory expansion purposes.

Instruction Word Register

The Instruction Word Register, I, is a 16-bit register that holds the current instruction for decode purposes. In conjunction with the Timing Control which generates machine states, the I Register performs micro-operation control of instruction algorithms and controls the machine state transitions.

Program Level Register

The Program Level Register, L, holds priority interrupt signals from peripheral devices. The basic size is 4 bits and expandable to 64 bits. The register can be set under program control or by external devices. The highest priority program level number corresponds to the highest numbered bit position that is set to a logical one. This number is an encoded six-bit binary number called the active Program Level, LP, and is used to address the executive page in memory to select the data stored in the Program Location Register, Accumulator, Index Register and status bits of the active priority program. The programmer has access to the Program Level number with a special instruction.

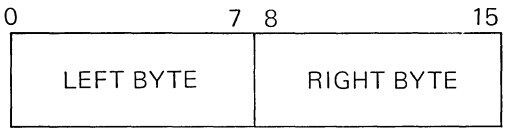
Program interrupt response can be inhibited under program control, a function provided by the Inhibit Interrupt flip-flop, H, that can be controlled by the programmer or a control panel switch.

WORD FORMATS

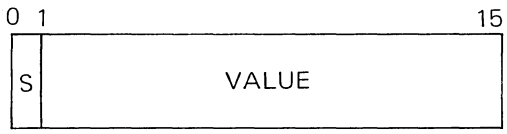
Data Words

MAC 16 can process two data word sizes, an 8-bit byte and a 16-bit word. Bit positions within a word are numbered left to right with the most significant bit position, position 0, representing the algebraic sign, S, of numeric data. A one bit in the S bit position represents a negative number, and a zero bit represents a positive number. Negative numbers are in two's complement form.

BYTE WORD FORMAT

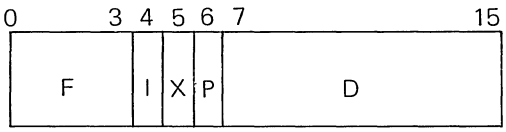


WORD FORMAT



Instruction Words

MAC 16 has two instruction word formats, Memory Reference Instruction Word and the Register Reference Instruction Word.



Five fields comprising a Memory Reference Instruction Word are:

F the instruction operation code, 4 bits.

I indirect address tag. Bit position four specifies indirect addressing if equal to a one and no indirect addressing if equal to zero.

X index tag. If bit 5 is a one, the content of the 16-bit index register for the current program level is added to the address specified by the I and P tags of the instruction.

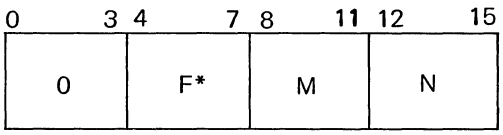
P page tag. If bit 6 is a one, a 16-bit operand address is formed from the most significant seven bits of the current instruction location and the nine bits of the D field. This yields a direct address that is within the local 512 word page. If bit 6 is a zero, a 16-bit indirect operand address is obtained from the base page.

The 16-bit address of the base page word is formed with the most significant three bits forced to zeros, the next four bit positions are bit positions 0, 1, and 2 of the current instruction location increased by one and the remaining nine bits are obtained from the D field of the instruction. This yields an address of a word within a

base page. All addresses within the base page are indirect addresses. The base page can be fixed as page one (addresses 512 to 1023) under program control independent of the instruction location if the page control flip-flop, B, is set to a one.

D address displacement. The least significant nine bits of the instruction are used to address one of 512 words within the page selected by the page tag, P.

2. Register Reference Instruction Word



Four fields comprise a Register Reference Instruction Word:

O Bit positions 0 through 3 must always be zero to specify a register Reference Instruction.

F* the instruction operation code, 4 bits.

M represents the most significant four bits of a device address or immediate operand. Also, in many cases it represents further definition of the instruction command on register reference instructions.

N represents the least significant four bits of a device address or immediate operand. In many register reference instructions it represents an increment, decrement value, or shift value. On instructions that set, reset or test status indicators or control indicators the N field selects the indicators.

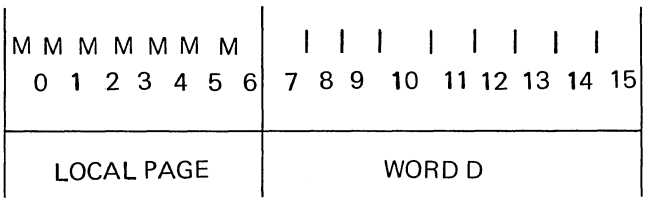
MEMORY ADDRESSING

Bit positions 4, 5 and 6 of the Memory Reference Instruction Word provide the programmer with eight operand address modes. Bit position 4, the indirect tag, I, specifies indirect addressing. When this bit position contains a one, the content of the word specified by the direct address is used as a 16-bit quantity that can be the effective address. A one bit in position 5, the index tag, X, specifies post indexing. That is, the content of the 16-bit index register for the current program level is added to the 16-bit direct or indirect address. Bit position 6, the page tag, P, specifies either the local page of 512 words or a base page of 512 words.

The execution sequence of these address options is P, I, X. (See Table 2-1.)

Local Page Addressing

If the page tag, P, of a memory reference instruction is a one, local page addressing is specified. That is, the desired next address is on this local page of memory. The direct address of the instruction operand or jump address is formed with the seven most significant bits of the Memory Address Register, $M_{0\text{-}6}$, and the nine bit D field of the Instruction Word Register, $I_{7\text{-}15}$. The direct address formed is:

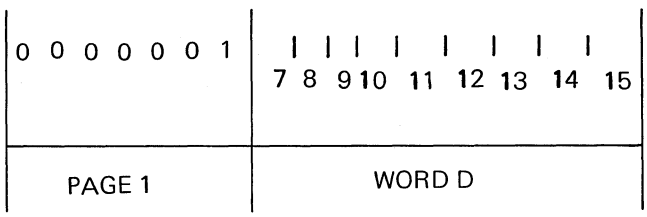


If the indirect tag, I, is a one, this 16-bit direct address is the address of a word within the local page, the content representing an indirect operand address. If the index tag, X, is a one, the 16-bit direct or indirect address is added to the Index Register to form a 16-bit effective address.

Base Page Addressing

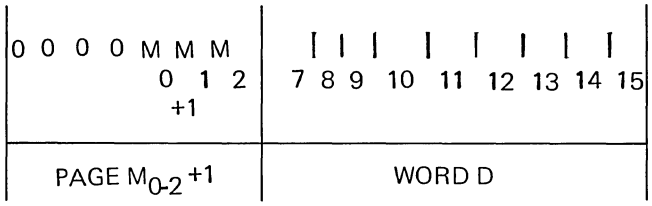
If the page tag, P, of a memory reference instruction is a zero, base page addressing is specified. The base page of 512 words may be a fixed page for all instructions or up to eight base pages may be used. A special Base Page Control flip-flop, B, is provided for each program level.

If the Base Page Control, B, is set to a one, then all memory reference instructions with a page tag, P, set to zero use the D field of the instruction to select a word in the fixed base page, addresses 512_{10} to 1023_{10}. This binary address is formed as follows;



When the Base Page Control is set to zero all memory reference instructions with a page tag, P, set to a zero use bit positions 0, 1 and 2 of the Memory Address, M, and the

D field of the instruction to select a word in the expanding base page. This binary address is formed as follows:



A one is added to bit positions 0 through 2 of the M Register. For instructions with location addresses between 0 and 8191_{10}, the base page is the same as the fixed base page, (page one). For instructions with location addresses between 8192_{10} and $16,383_{10}$, the base page is page two. For instructions with location addresses between $57,344_{10}$ and $65,535_{10}$, the base page is page eight, i.e. 4096_{10} to 4607_{10} is addressed.

Three instructions are provided to control the Base Page Control flip-flop, B, that can be set, reset, and tested.

All references to a base page with the page tag, P, set to a zero, will result in an indirect address. If the indirect tag, I, is a one, two levels of indirect addressing occur.

Index Register Addressing

Six register reference instructions are provided that use the content of the Index Register as a memory address. The N field of the instruction can be used to displace this address. These instructions follow:

LAX Load A from X + N
LIX Load A from (X) + N
SAX Store A in X + N
SIX Store A in (X) + N
JMX Jump Unconditional to X + N
JIX Jump Unconditional to (X) + N

On the LAX, SAX and JMX instructions, the address is formed by adding the four bit N field to the content of the current program level Index Register.

On the LIX, SIX and JIX instructions, the address is formed by adding the four bit N field to the word that is addressed by the content of the current program level Index Register.

Immediate Byte

Three register reference instructions are provided that contain an eight bit operand (byte) within the M and N fields of the instruction:

LDI Load Byte Immediate
ADI Add Byte Immediate
SDI Subtract Byte Immediate

The byte operand is an unsigned quantity that is loaded, added or subtracted into bit positions 8 to 15 of the A Register. The A Register is treated as a 16-bit signed number.

Table 2-1. Memory Reference Address Modes

1 Bits I X P	Effective Address, Y	Mode Name	Description
0 0 0	Y=(B:D)	base page,indirect (adds 1 μs to instruction time)	The D field is linked with the address of the base page to form a 16-bit address. The content of the word at this address is used as the effective operand address.
0 0 1	$Y=M_{0\text{-}6}:D$	local, direct	The D field is linked with the most significant seven bits of the Memory Address Register M, to form a 16-bit effective operand address.
0 1 0	Y=(B:D)+X	base page, indirect post-indexed (adds 2 μs to instruction time)	The D field is linked with the address of the base page to form a 16-bit address. The content of the word at this address is added to the content of the Index Register to form a 16-bit effective operand address.
0 1 1	$Y=M_{0\text{-}6}:D+X$	local, direct, indexed (adds 1 μs to instruction time)	The D field is linked with the most significant seven bits of the Memory Address Register, M, to form a 16-bit quantity that is added to the content of the Index Register to form a 16-bit effective operand address.
1 0 0	Y=((B:D))	base page, double indirect (adds 2 μs to instruction time)	The D field is linked with the address of the base page to form a 16-bit address. The content of the word at this address is used to address a word from memory the content of which represents the 16-bit effective operand address.
1 0 1	$Y=(M_{0\text{-}6}D)$	local, indirect (adds 1 μs to instruction time)	The D field is linked with the most significant seven bits of the Memory Address Register, M, to form the address of a word in memory the content of which is used as the 16-bit effective operand address.
1 1 0	Y=((B:D))+X	base page, double indirect, post-indexed (adds 3 μs to instruction time)	The D field is linked with the address of the base page to form a 16-bit address. The content of the word at this address is used to address a word from memory the content of which is added to the content of the Index Register to form a 16-bit effective operand address.
1 1 1	$Y=(M_{0\text{-}6}:D)+X$	local, indirect, post-indexed (adds 2 μs to instruction time)	The D field is linked with the most significant seven bits of the Memory Address Register, M, to form the address of a word in memory the content of which is added to the content of the Index Register to form a 16-bit effective operand address.

Chapter 3 Instruction Repertoire

BASIC REPERTOIRE

The basic repertoire of the MAC 16 Computer comprises seventy-two instructions. This section contains detailed descriptions of these instructions, grouped according to function plus the optional Multiply and divide instructions. The Appendix contains a summary of this section with timing information, an alphabetical listing of instruction mnemonics, and a numerical listing of hexadecimal operation codes.

Hexadecimal notation is used to represent machine instruction codes. The letters A, B, C, D, E, and F represent the decimal values 10, 11, 12, 13, 14, and 15 respectively.

Descriptive material for each instruction is as follows:
Name of the instruction in English.
Instruction execution time in microseconds.
Mnemonic operation code used in the Lockheed Electronics Assembly Program (LEAP)
Machine operation codes in hexadecimal.
Alogrithm for instruction operation.
Text description of operation.
Computer elements altered by operation.
Notes concerning operation.

The shading in the instruction boxes indicates the following:
Shading only: The field is ignored for that instruction
Shading plus hex number: The range of bits in the field
Shading with field letters: Bits determined by programmer.
No shading and hex number: Bits determine the instruction.

The following notation is used in the instruction descriptions.

Symbol	Definition
A	The Accumulator
$\overline{A}$	One's complement of the Accumulator
B	Base page control flip-flop
C	Carry indicator
DC	Data Controller
F	Memory reference instruction code; a hexadecimal number
F*	Register reference instruction code; a hexadecimal number
H	Master Interrupt Inhibit flip-flop
i	Bit position of the associated register; used as a subscript
I	Instruction Word Register
L	Program Level Register
LP	The priority level number
M	Four-bit field of register reference instructions (8-11)
N	Four-bit field of register reference instructions (12-15)
O	Operand instruction code
P	Program Location Register
Q	Quiescent indicator
R	Status indicator
S	Status indicator
SW	Control panel switches
T	Sense switch
V	Overflow indicator
X	Index Register
Y	The effective address of any instruction
0	A zero bit or all zeroes
1	A logical one or the integer one
→	Replaces
( )	Read as the content of the memory location that is selected by the address within the parentheses.

The following instruction fields take the values indicated unless otherwise specified:

IXP field: Three-bit field with a bit value of 000 to 111 in any combination.

M field: hexadecimal value 0 to F.

N field: hexadecimal value 0 to F.

MN field: combined fields, hexadecimal value 00 to FF.

D field: hexadecimal value 000 to 1FF.

Instruction execution time:

Memory reference instruction execution timing is increased one microsecond for each of the following functions:

I=1 Indirect addressing; add 1 us

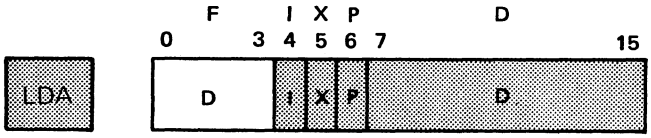
X=1 Indexing; add 1 us

P=0 Base page addressing, add 1 us

Instruction execution time for shift instructions depends upon the number of bit positions shifted as specified in the N field of the instruction.

If N is 0 to 4, shift is 2 microseconds.
If N is 5 to 8, shift is 3 microseconds.
If N is 9 to 12, shift is 4 microseconds.
If N is 13 to 15, shift is 5 microseconds.

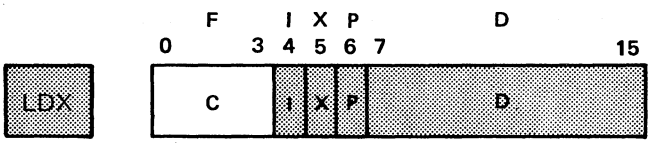
LOAD AND STORE INSTRUCTIONS



LOAD 2 μsec
(Y) ⟶ A

The content of the memory location specified by the effective address replaces the content of the Accumulator.

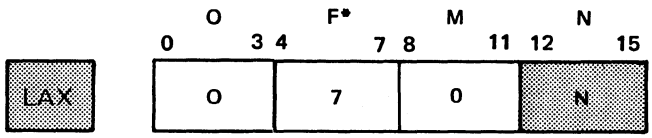
Alters: A



LOAD INDEX 3 μsec
(Y) ⟶ X

The content of the memory location specified by the effective address replaces the content of the Index Register of the current program level.

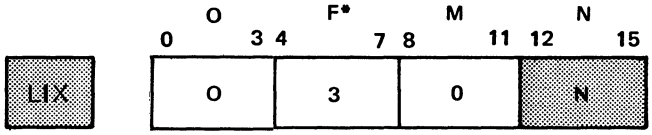
Alters: X



LOAD X+N 3 μsec
(X+N) ⟶ A

An address is formed by the sum of the N field of the instruction and the Index Register. The content of the address replaces the content of the Accumulator.

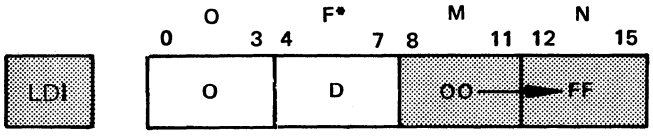
Alters: A



LOAD INDIRECT (X)+N 4 μsec
((X)+N) ⟶ A

An address is formed by the sum of the content of the memory location specified by the Index Register and the N field of the instruction, and the content of that address is placed in the Accumulator.

Alters: A

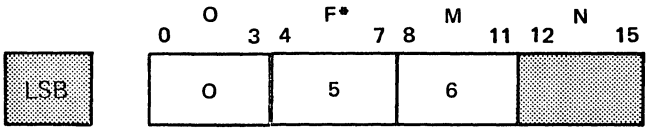


LOAD IMMEDIATE 1 μsec
M, N ⟶ A_{8-15}, 0 ⟶ A_{0-7}

The content of the M and N fields, bits 8 through 15 of the instruction replaces the least significant byte of the Accumulator. The most significant byte of the Accumulator is set to zero.

Alters: A

Note: A program interrupt cannot occur following this instruction.



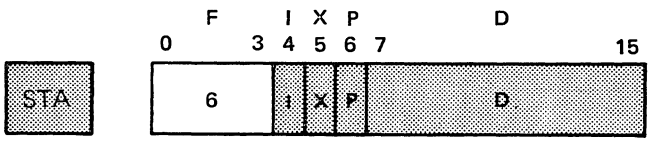
LOAD STATUS BITS 1 μsec

CVRSBH ⟶ A_{0-5}, 0 ⟶ A_{6-15}

The condition of the current program level status indicators is copied into bits 0 through 5 of the Accumulator; bits 6 through 15 of the Accumulator are set to zero.

Alters: A

Note: A program interrupt cannot occur following this instruction.

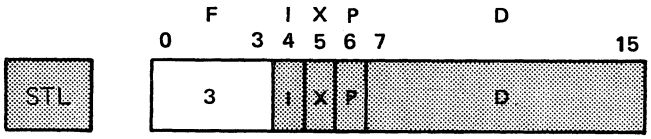


STORE 2 μsec

A ⟶ (Y)

The content of the Accumulator replaces the content of the memory location specified by the effective address.

Alters: (Y)

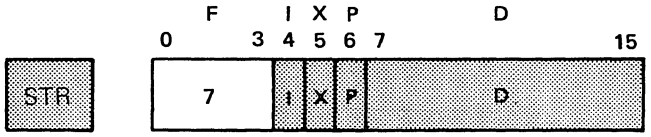


STORE BYTE, LEFT 2 μsec

A_{8-15} ⟶ $(Y)_{0-7}$

The least significant byte of the Accumulator replaces the most significant byte of the memory location specified by the effective address.

Alters: $(Y)_{0-7}$

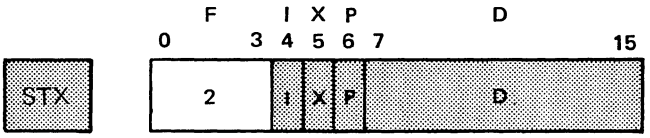


STORE BYTE, RIGHT 2 μsec

A_{8-15} ⟶ $(Y)_{8-15}$

The least significant byte of the Accumulator replaces the least significant byte of the memory location specified by the effective address.

Alters: $(Y)_{8-15}$

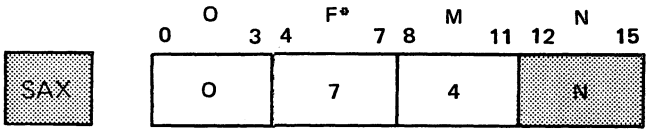


STORE INDEX 3 μsec

X ⟶ (Y)

The content of the current program level Index Register replaces the content of the memory location specified by the effective address.

Alters: (Y)

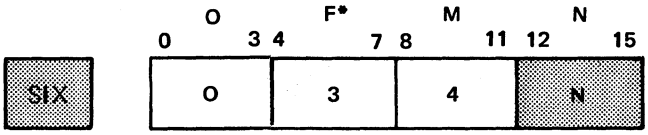


STORE X+N 3 μsec

A ⟶ (X+N)

The content of the Accumulator replaces the content of the memory location specified by the address formed by the sum of the Index Register and the N field of the instruction.

Alters: Y

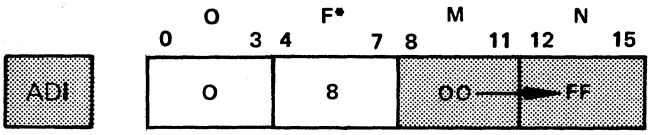


STORE (X)+N 4 μsec

A ⟶ ((X)+N)

A memory address is formed by the sum of the content of the memory location specified by the Index Register with the N field of the instruction. The content of the Accumulator replaces the content of this memory address.

Alters: Y



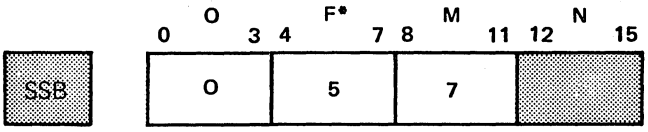
ADD IMMEDIATE 2 μsec

A+M, N ⟶ A

The content of the M and N fields, bits 8 through 15 of the instruction, is treated as an 8-bit unsigned integer and is added to the content of the Accumulator.

The carry out of bit position zero of the Accumulator sets the carry indicator. If the sum of two numbers exceeds $+ 1.0\text{-}2^{-15}$ the Overflow indicator is set.

Alters: A, C, V

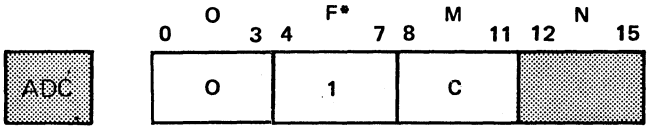


STORE STATUS BITS 1 μsec

$A_{0\text{-}5}$ ⟶ CVRSBH

Bits 0 through 5 of the Accumulator are copied into the current program level status indicators.

Alters: CVRSBH



ADD CARRY 2 μsec

A+C ⟶ A

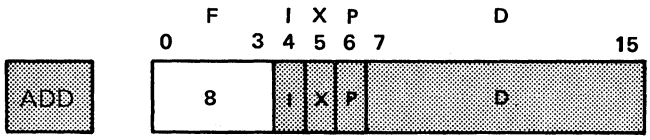
The content of the Carry indicator is added to the least significant bit position of the Accumulator.

The carry out of bit position zero of the Accumulator sets the carry indicator. If the sum of two numbers exceeds $+ 1.0\text{-}2^{-15}$ the Overflow indicator is set.

Alters: A, C, V

Note: The N field of this instruction is ignored.

ARITHMETIC INSTRUCTIONS



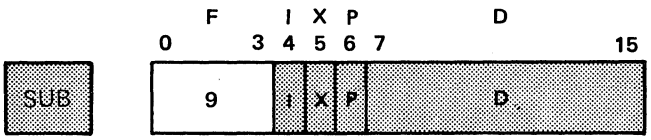
ADD 2 μsec

A+(Y) ⟶ A

The content of the memory location specified by the effective address is added to the content of the Accumulator.

A carry out of bit position zero of the Accumulator sets the carry indicator. If the sum of two numbers exceeds $+ 1.0\text{-}2^{-15}$ or -1.0 the Overflow indicator is set.

Alters: A, C, V



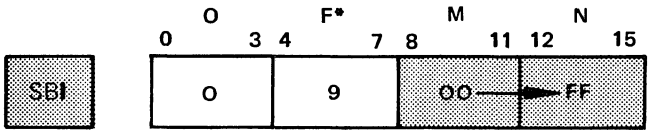
SUBTRACT 2 μsec

A - (Y) ⟶ A

The content of the memory location specified by the effective address is subtracted from the content of the Accumulator.

A Carry out of bit position zero of the Accumulator sets the Carry indicator. If the difference of the two numbers exceeds $+ 1.0 - 2^{-15}$ or -1.0 the Overflow indicator is set.

Alters: A, C, V



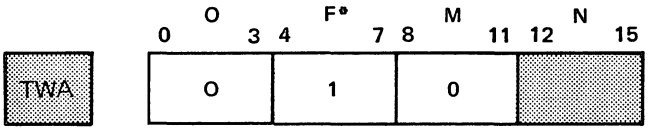
SUBTRACT IMMEDIATE 2 μsec

A - M, N ⟶ A

The content of the M and N fields, bits 8 through 15 of the instruction, is treated as an 8-bit unsigned integer and is subtracted from the content of the Accumulator.

A Carry out of bit position zero of the Accumulator sets the Carry indicator. If the difference of the two numbers exceeds -1.0 the Overflow indicator is set.

Alters: A, C, V



TWO'S COMPLEMENT 2 μsec

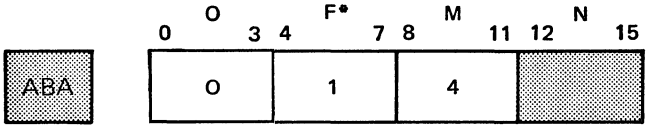
2^{16} - A ⟶ A

The content of the Accumulator is arithmetically subtracted from all zeros to form the two's complement.

The two's complement of minus one sets the V indicator; the content of the Accumulator is not altered. The two's complement of zero sets the C indicator, the content of the Accumulator is not altered.

Alters: A, C, V

Note: The N field of this instruction is ignored.



ABSOLUTE VALUE 2 μsec

A ⟶ A

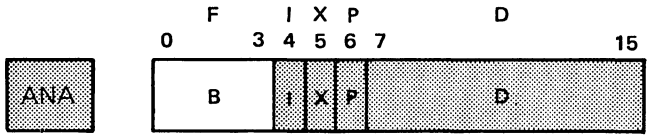
The content of the Accumulator, if negative, is complemented to form the equivalent positive value. No operation occurs if the content of the Accumulator is positive.

The absolute value of minus one sets the V indicator; the content of the Accumulator is not altered.

Alters: A, V

Note: The N field of this instruction is ignored.

LOGIC INSTRUCTIONS



AND 2 μsec

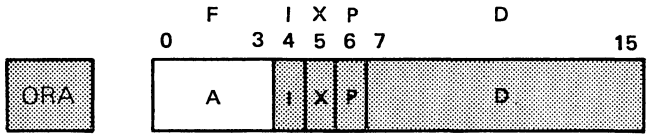
A ∩ (Y) ⟶ A

The logical product of the Accumulator and the content of the memory location specified by the effective address replaces the content of the Accumulator.

Alters: A

Note: Truth table for AND:

0 AND 0 = 0 1 AND 0 = 0
0 AND 1 = 0 1 AND 1 = 1



OR 2 μsec

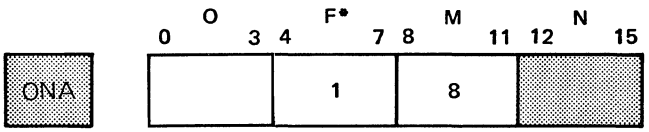
A ∪ (Y) ⟶ A

The logical sum of the Accumulator and the content of the memory location specified by the effective address replaces the content of the Accumulator.

Alters: A

Note: Truth table for OR:

0 OR 0 = 0 1 OR 0 = 1
0 OR 1 = 1 1 OR 1 = 1



ONE'S COMPLEMENT 2 μsec

$\overline{A}$ ⟶ A

The one's complement (logical inversion) of the Accumulator replaces the content of the Accumulator

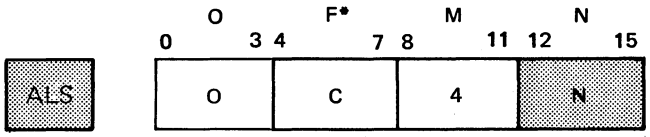
Alters: A

Note: Truth table:

$\overline{0}$ = 1
$\overline{1}$ = 0

The N field of this instruction is ignored.

SHIFT INSTRUCTIONS



ARITHMETIC LEFT, SAVE CARRY 2-5 μsec

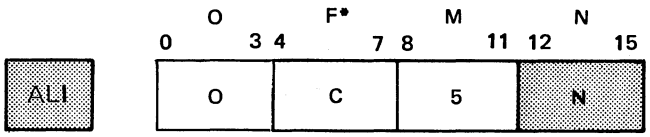
$A_{i+1} \longrightarrow A_i$, $0 \longrightarrow A_{15}$

i = 1-14, $A_1 \longrightarrow$ C, Set V if $A_1 \neq A_0$

Bits 1 through 15 of the Accumulator are shifted N bit positions left, the sign bit is unchanged. Bits shifted out of position one pass through the Carry indicator. Zeros are shifted into bit position 15 of the Accumulator.

The Overflow indicator is set if a bit unlike the sign bit is shifted into the Carry indicator.

Alters: A, C, V



ARITHMETIC LEFT, INSERT CARRY 2-5 μsec

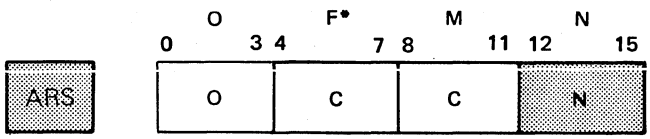
$A_{i+1} \longrightarrow A_i$ $C \longrightarrow A_{15}$

i = 1-14, $A_1 \longrightarrow$ C, Set V if $A_1 \neq A_0$

Bits 1 through 15 of the Accumulator are shifted N bit positions left, the sign bit is unchanged. Bits shifted out of position one pass through the Carry indicator and into position 15 of the Accumulator. The original state of the Carry indicator is shifted into the Accumulator on the first shift.

The Overflow indicator is set if a bit unlike the sign bit is shifted into the Carry indicator.

Alters: A, C, V



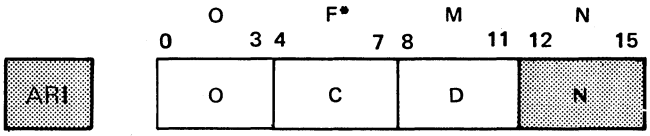
ARITHMETIC RIGHT, SAVE CARRY 2-5 μsec

$A_i \longrightarrow A_{i+1}$, $A_0 \longrightarrow A_1$

i = 1-14, $A_{15} \longrightarrow$ C

Bits 1 through 15 of the Accumulator are shifted N bit positions right, the sign bit is propagated right into position one. Bits shifted out of position fifteen pass through the Carry indicator.

Alters: A, C



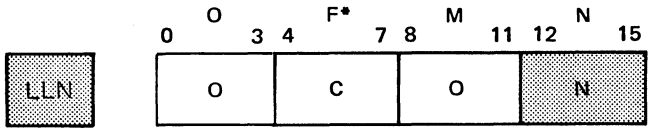
ARITHMETIC RIGHT, INSERT CARRY 2-5 μsec

$A_i \longrightarrow A_{i+1}$ $C \longrightarrow A_1$

i = 1-14, $A_{15} \longrightarrow$ C

Bits 1 through 15 of the Accumulator are shifted N bit positions right. Bits shifted out of position fifteen pass through the Carry indicator and into position one of the Accumulator. The original station of the Carry indicator is shifted into position one on the first shift. The sign bit is unchanged.

Alters: A, C



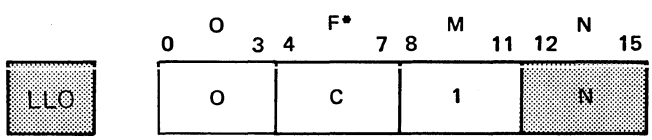
LOGICAL LEFT NO CARRY 2-5 μsec

$A_{i+1} \longrightarrow A_i$, $0 \longrightarrow A_{15}$

i = 0-14

The content of the Accumulator is shifted N bit positions left. Bits shifted out of position zero are lost; zeros are shifted into bit position fifteen.

Alters: A



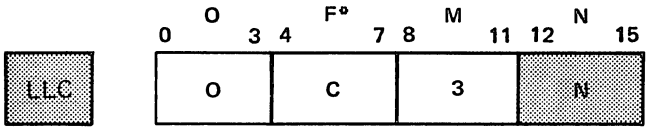
LOGICAL LEFT OPEN 2-5 μsec

$A_{i+1} \longrightarrow A_i$, $0 \longrightarrow A_{15}$

i = 0-14, $A_0 \longrightarrow$ C

The content of the Accumulator is shifted N bit positions left. Bits shifted out of position zero pass through the Carry indicator. Zeros are shifted into bit position 15 of the Accumulator.

Alters: A, C



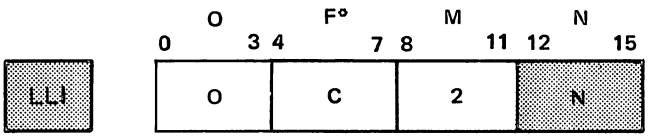
LOGICAL LEFT CLOSED 2-5 μsec

$A_{i+1} \longrightarrow A_i$, $A_0 \longrightarrow A_{15}$

i = 0-14, $A_0 \longrightarrow C$

The content of the Accumulator is shifted N bit positions left. Bits shifted out of position zero pass into the Carry indicator and bit position 15 of the Accumulator.

Alters: A, C



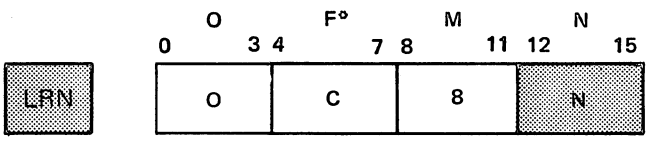
LOGICAL LEFT, INSERT CARRY 2-5 μsec

$A_{i+1} \longrightarrow A_i$, $C \longrightarrow A_{15}$

i = 0-14, $A_0 \longrightarrow C$

The content of the Accumulator is shifted N bit positions left. Bits shifted out of position zero pass into the Carry indicator and then into position 15 of the Accumulator. The original state of the Carry indicator is shifted into the Accumulator on the first shift.

Alters: A, C



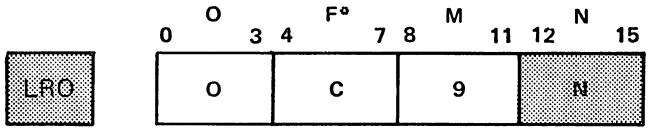
LOGICAL RIGHT, NO CARRY 2-5 μsec

$A_i \longrightarrow A_{i+1}$, $0 \longrightarrow A_0$

i = 0-14

The content of the Accumulator is shifted N bit positions right. Bits shifted out of position 15 are lost; zeros are shifted into bit position zero.

Alters: A



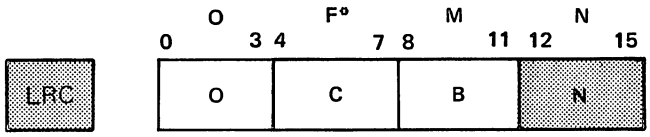
LOGICAL RIGHT OPEN 2-5 μsec

$A_i \longrightarrow A_{i+1}$, $0 \longrightarrow A_0$

i = 0-14 $A_{15} \longrightarrow C$

The content of the Accumulator is shifted N bit positions right. Bits shifted out of position 15 pass through the Carry indicator. Zeros are shifted into bit position zero of the Accumulator.

Alters: A, C



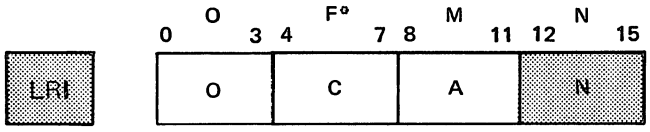
LOGICAL RIGHT CLOSED 2-5 μsec

$A_i \longrightarrow A_{i+1}$ $A_{15} \longrightarrow A_0$

i = 0-14, $A_{15} \longrightarrow C$

The content of the Accumulator is shifted N bit positions right. Bits shifted out of position 15 pass into the Carry indicator and bit position zero of the Accumulator.

Alters: A, C



LOGICAL RIGHT, INSERT CARRY 2-5 μsec

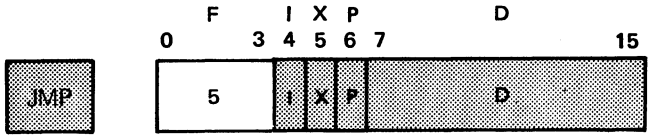
$A_i \longrightarrow A_{i+1}$, $C \longrightarrow A_0$

i = 0-14, $A_{15} \longrightarrow C$

The content of the Accumulator is shifted N bit positions right. Bits shifted out of position 15 pass into the Carry indicator and then into position zero of the Accumulator. The original state of the Carry indicator is shifted into the Accumulator on the first shift.

Alters: A, C

JUMP INSTRUCTIONS

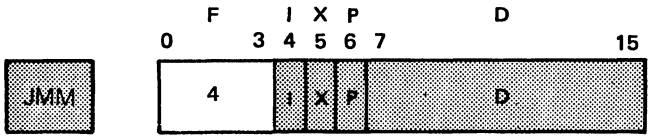


JUMP UNCONDITIONAL 2 μsec

Y ⟶ P,

The effective address specified by the instruction replaces the content of the Program Location Register. The next instruction executed is taken from the effective address.

Alters: P

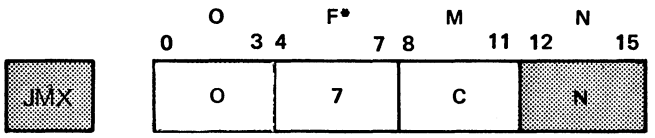


JUMP, MARK 3 μsec

P + 1 ⟶ X, Y ⟶ P

The address of the next instruction in sequence, P+1, is stored in the memory location used to store the contents of the Index Register for the current program level. The effective address specified by the current instruction replaces the content of the Program Location Register. The next instruction executed is that address taken from the effective address (specified by P.)

Alters: P, X

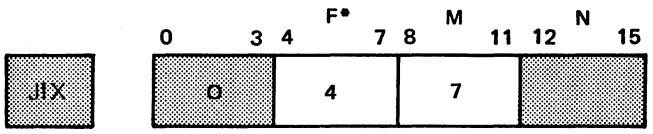


JUMP TO X+N 3 μsec

X + N ⟶ P

The address specified by the sum of the Index Register and the N field of the Instruction Word Register replaces the content of the Program Location Register. The next instruction executed is taken from this address.

Alters: P

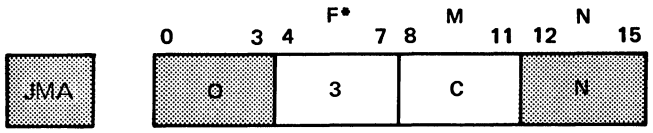


JUMP INDIRECT TO (X)+N 4 μsec

(X) + N ⟶ P

The sum of the content of the memory location specified by the Index Register and the N field of the Instruction Word replaces the content of the Program Location Register. The next instruction executed is taken from this address.

Alters: P



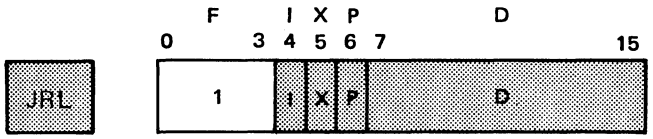
JUMP TO ACCUMULATOR 2 μsec

A ⟶ P, P + 1 ⟶ A

The content of the Accumulator is exchanged with the content of the Program Location Register. The next instruction executed is taken from the address previously contained in the Accumulator.

Alters: A, P

Note: The N field of this instruction is ignored.



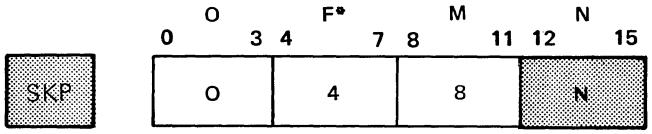
JUMP RESET LEVEL 2 μsec

Y ⟶ P, 0 ⟶ L_i

The effective address specified by the instruction replaces the content of the Program Location Register. The interrupt flip-flop corresponding to the current program level is reset and the next highest program level becomes active. If no interrupt is set in the L Register the computer becomes quiescent. When the current program level is reactivated, execution begins at the effective address of this instruction.

Alters: P, L_i

TEST AND SKIP INSTRUCTIONS



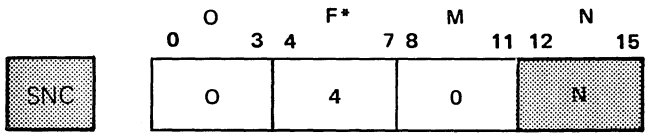
SKIP UNCONDITIONAL 2 μsec

P + 1 + N ⟶ P

The next N instructions in sequence are unconditionally skipped.

Alters: P

Note: If N = 0, see NO OPERATION.

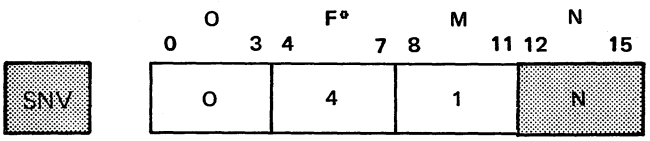


SKIP NO CARRY 2 μsec

P + 1 + N ⟶ P if C=0

The next N instructions in sequence are skipped if the C indicator is zero. The next instruction in sequence is executed if the C indicator is one.

Alters: P if skip occurs.

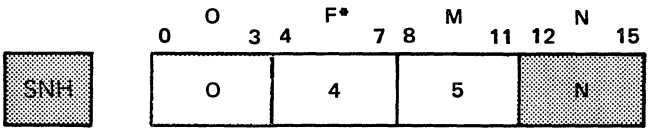


SKIP NO OVERFLOW 2 μsec

P + 1 + N ⟶ P if V=0

The next N instructions in sequence are skipped if the Overflow indicator is zero. The next instruction in sequence is executed if the V indicator is one.

Alters: P if skip occurs.

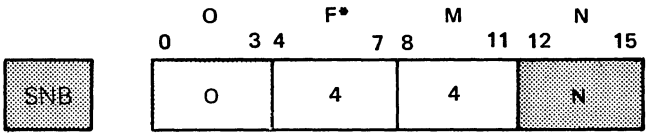


SKIP NO INTERRUPT INHIBIT 2 μsec

P + 1 + N ⟶ P if H=0

The next N instructions in sequence are skipped if the Interrupt Inhibit indicator is zero. The next instruction in sequence is executed if the H indicator is one.

Alters: P if skip occurs.

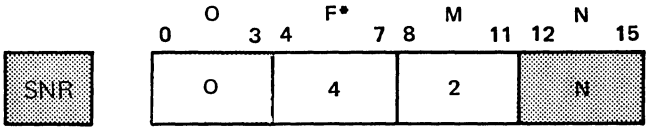


SKIP NO BASE PAGE CONTROL 2 μsec

P + 1 + N ⟶ P if B=0

The next N instructions in sequence are skipped if the B indicator is zero. The next instruction in sequence is executed if the B indicator is one.

Alters: P if skip occurs.

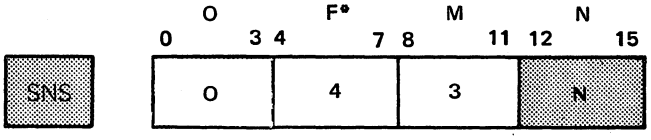


SKIP IF R ZERO 2 μsec

P + 1 + N ⟶ P if R=0

The next N instructions in sequence are skipped if the R indicator is zero. The next instruction in sequence is executed if the R indicator is one.

Alters: P if skip occurs.

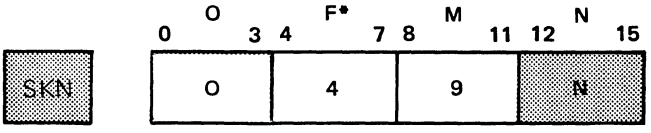


SKIP IF S ZERO 2 μsec

P + 1 + N ⟶ P if S=0

The next N instructions in sequence are skipped if the S indicator is zero. The next instruction in sequence is executed if the S indicator is one.

Alters: P if skip occurs.



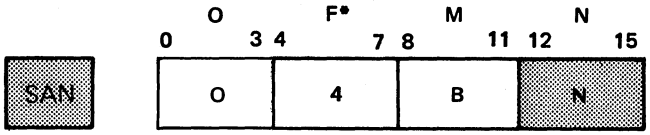
SKIP IF NORMALIZED 2 μsec

P + 1 + N ⟶ P if $A_0=A_1=A_2=A_3=A_4$

The next N instructions in sequence are skipped if the five most significant bits of the Accumulator are identical (all ones or all zeros). The next instruction in sequence is executed if the five most significant bits of the Accumulator are different (do not match).

Alters: P if skip occurs.

Note: It is assumed that a floating point base of 16 is used and therefore provides a hex normalize test.

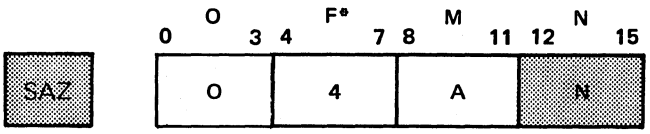


SKIP IF NEGATIVE 2 μsec

P + 1 + N ⟶ P if A_0=1

The next N instructions in sequence are skipped if the most significant bit position of the Accumulator is one. The next instruction in sequence is executed if the most significant bit of the Accumulator is zero.

Alters: P if skip occurs.

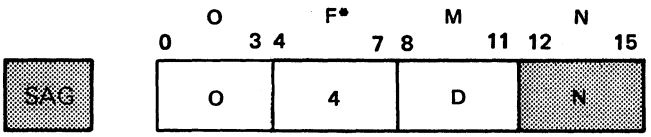


SKIP IF ZERO 2 μsec

P + 1 + N ⟶ P if A = 0

The next N instructions in sequence are skipped if the content of the Accumulator is zero. The next instruction in sequence is executed if the content of the Accumulator is not zero.

Alters: P if skip occurs.

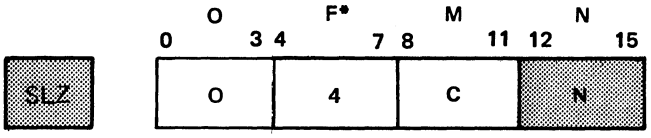


SKIP IF GREATER THAN ZERO 2 μsec

P + 1 + N ⟶ P if A > 0

The next N instructions in sequence are skipped if the content of the Accumulator is a positive value. The next instruction in sequence is executed if the content of the Accumulator is zero or a negative value.

Alters: P if skip occurs.

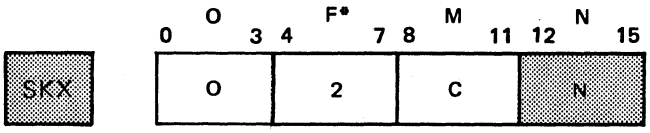


SKIP IF LSB ZERO 2 μsec

P + 1 + N ⟶ P if A_{15}=0

The next N instructions in sequence are skipped if the least significant bit of the Accumulator is zero. The next instruction in sequence is executed if the least significant bit of the Accumulator is one.

Alters: P if skip occurs.



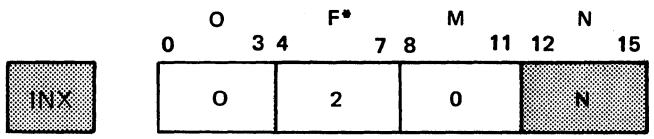
SKIP IF INDEX ZERO 3 μsec

P + 1 + N ⟶ P if X=0

The next N instructions in sequence are skipped if the content of the Index Register is zero. The next instruction in sequence is executed if any bit position of the Index Register contains a one.

Alters: P if skip occurs.

INDEX INSTRUCTIONS



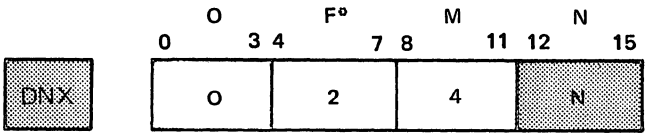
INCREMENT INDEX AND SKIP 3 μsec

X + N ⟶ X

The content of the N field of the instruction is treated as a positive integer and is added to the content of the Index Register. The next instruction in sequence is executed if the resultant 16-bit sum does not become zero or pass through zero. The next instruction in sequence is skipped if the sum equals zero or passes through zero.

Alters: X; P if skip occurs.

Note: If X and N are initially zero, no skip occurs.



DECREMENT INDEX AND SKIP 3 μsec

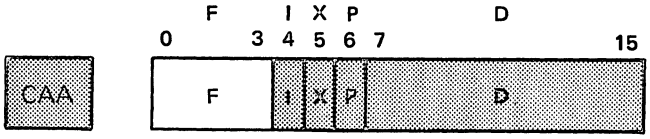
X - N ⟶ X

The content of the N field of the instruction is treated as a positive integer and is subtracted from the content of the Index Register. The next instruction in sequence is executed if the resultant 16-bit difference does not become zero or pass through zero. The next instruction in sequence is skipped if the difference equals zero or passes through zero.

Alters: X; P if skip occurs.

Note: If X and N are initially zero, a skip occurs.

COMPARE INSTRUCTION



COMPARE ARITHMETIC AND SKIP 3 μsec

P + 1 ⟶ P if A>(Y)

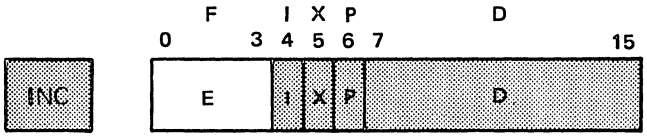
P + 2 ⟶ P if A<(Y)

P + 3 ⟶ P if A=(Y)

The content of the memory location specified by the effective address is compared arithmetically with the content of the Accumulator. If the content of the Accumulator is algebraically greater than the content of the memory location word, the next instruction in sequence is executed. If the content of the Accumulator is algebraically less than the memory word, the next instruction in sequence is skipped. If the content of the Accumulator is equal to the memory word, the next two instructions in sequence are skipped.

Alters: P if skip occurs.

INCREMENT MEMORY INSTRUCTION



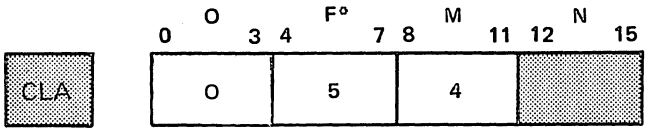
INCREMENT MEMORY AND SKIP 3 μsec

(Y) + 1 ⟶ (Y)

The content of the memory location specified by the effective address is incremented by one. The next instruction in sequence is executed if the result of the incremented location is not zero. The next instruction in sequence is skipped if the result of the incremented location is zero.

Alters: Y, P if skip occurs.

REGISTER INSTRUCTIONS



CLEAR ACCUMULATOR 1 μsec

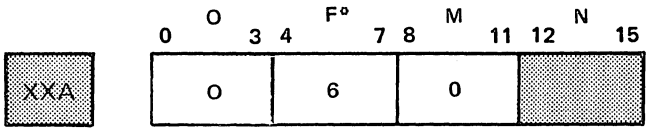
0 ⟶ A

The content of the Accumulator is reset to zero.

Alters: A

Note: A program interrupt cannot occur following this instruction.

The N field of this instruction is ignored.



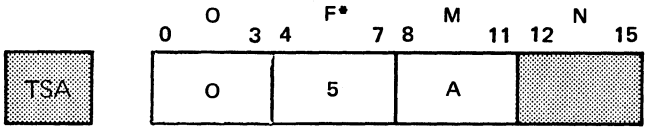
EXCHANGE INDEX WITH ACCUMULATOR 3 μsec

X ⟶ A, A ⟶ X

The content of the Accumulator and Index Register of the current program level are exchanged.

Alters: A, X

Note: The N field of this instruction is ignored.



TRANSFER DATA SWITCHES 1 μsec

SW ⟶ A if SEN=1

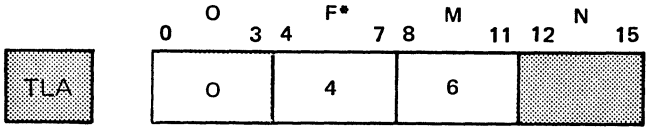
The content of the Accumulator is replaced by the setting of the control panel switches if the SEN indicator is on. If the SEN indicator is off the Accumulator is cleared to zero.

The SEN indicator is set by momentarily pressing the Control Panel Sense Switch. The SEN indicator is reset by execution of the Transfer Data Switches instruction.

Alters: A

Note: A program level interrupt cannot occur following this instruction.

The N field of this instruction is ignored.



TRANSFER PROGRAM LEVEL 2 μsec

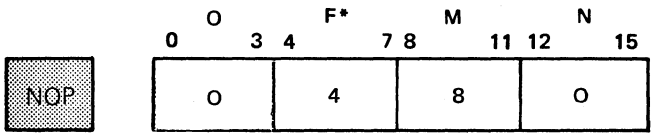
LP ⟶ $A_{8\text{-}13}$

The content of the Accumulator is reset to zero and the six-bit encoded binary number of the current Program Level is transferred to bit positions 8 through 13 of the Accumulator.

Alters: A

Note: The N field of this instruction is ignored.

CONTROL INSTRUCTIONS



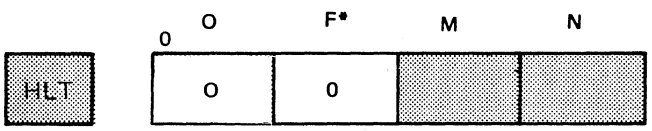
NO OPERATION 2 μsec

P + 1 ⟶ P

No instruction operation is performed. The content of the Program Location Register is incremented by one.

Alters: P

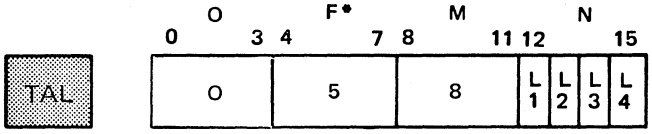
Note: The N field of this instruction must be zero. See Skip Unconditional.



HALT 1 μsec

The Halt instruction stops program execution. Restart at the next instruction in sequence is accomplished by pressing the Run switch on the Control Panel.

The M and N fields of this instruction are ignored.



TRANSFER TO PROGRAM LEVEL REGISTER 1 μsec

A∪L ⟶ $L_{1, 2, 3, \text{ or } 4}$

The content of the Accumulator is logically OR'd with the content of the Program Level Register (L). The N field of the Instruction Register specifies a 16-bit field within the Program Level Register.

Alters: L

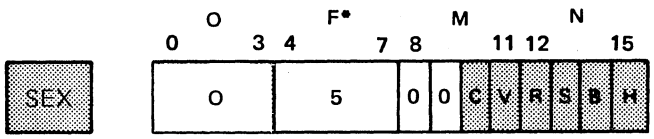
Note:

N=8 bits 0 through 15 of L N=2 bits 32 through 47 of L

N=4 bits 16 through 31 of L N=1 bits 48 through 63 of L

The N fields can be combined.

A program interrupt cannot occur following this instruction.



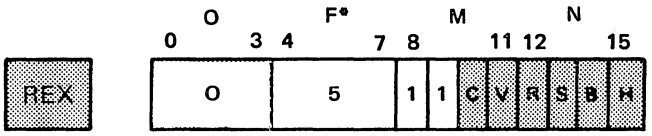
SET INDICATORS 1 μsec

M, $N_{10\text{-}15}$ ⟶ CVRSBH

The corresponding CVRSBH status indicator is set if any bit in the respective instruction positions 10 through 15 is set. Non-selected indicators (zero bit) are not altered.

Alters: C, V, R, S, B or H

Note: A program interrupt cannot occur following this instruction.



RESET INDICATORS 1 μsec

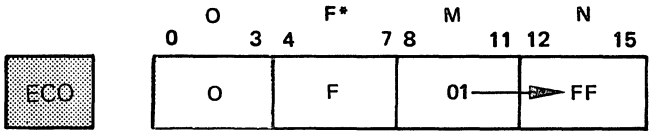
M, N_{10-15} ⟶ CVRSBH

The corresponding CVRSBH status indicator is reset if any bit in the respective instruction positions 10 through 15 is set. Non-selected indicators (zero bit) are not altered.

Alters: C, V, R, S, B, or H

Note: A program interrupt cannot occur following this instruction.

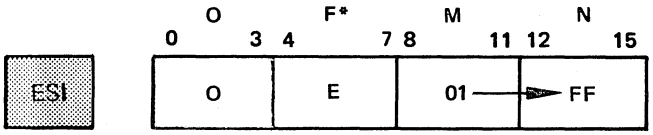
DEVICE CONTROL INSTRUCTIONS



EXTERNAL COMMAND OUT 3 μsec

A ⟶ DC

The content of the Accumulator is transmitted via the Programmed Data Channel to the device controller addressed by the M and N fields of the Instruction Register. The next instruction in sequence is skipped if the device controller acknowledges receipt of the transmission. The next instruction in sequence is executed if the device controller does not acknowledge receipt of the transmission.

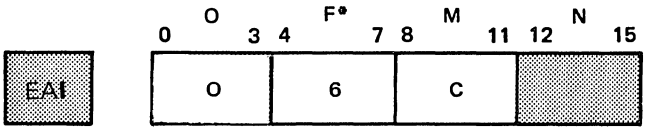


EXTERNAL STATUS IN 3 μsec

Device Status ⟶ A

The device controller addressed by the M and N fields of the Instruction Register transmits up to 16 bits of status information to the Accumulator.

Alters: A



EXTERNAL ADDRESS IN 3 μsec

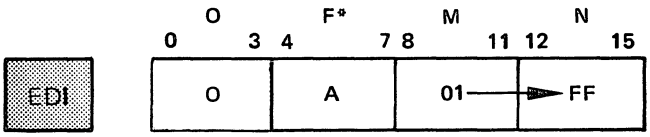
Device Address ⟶ A_{8-15}, Zero ⟶ A_{0-7}

The address of the highest priority device controller requesting service on a shared interrupt line of the Programmed Data Channel is transmitted to the eight least significant bits of the Accumulator. The eight most significant bits of the Accumulator are set to zero.

Alters: A

Note: The N field of this instruction is ignored.

INPUT/OUTPUT INSTRUCTIONS

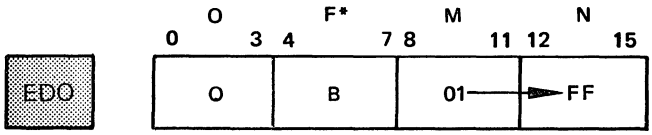


EXTERNAL DATA IN 3 μsec

Device Controller ⟶ A

The device controller addressed by the M and N fields of the Instruction transmits up to 16 bits of data to the Accumulator via the Programmed Data Channel. The next instruction in sequence is skipped if the device controller acknowledges transmission of the data. The next instruction in sequence is executed and the Accumulator is normally set to zero if the device controller does not acknowledge transmission of the data.

Alters: A



EXTERNAL DATA OUT 3 μsec

A ⟶ Device Controller

The content of the Accumulator is transmitted via the Programmed Data Channel to the device controller addressed by the M and N field of the Instruction.

Up to 16 bits of data can be transmitted. The next instruction in sequence is skipped if the device controller acknowledges receipt of the data. The next instruction in sequence is executed if the device controller does not acknowledge receipt of the data.

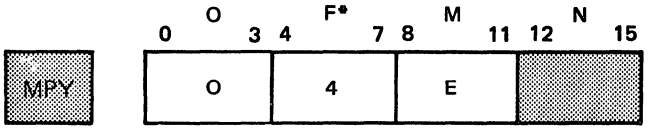
OPTIONAL INSTRUCTIONS

Multiply and Divide

Multiply and Divide instructions may be added to the MAC-16 instruction list. These instructions are represented by special register reference instruction codes.

Use of the optional Multiply and Divide instructions does not require changes in MAC-16 software. This is because the calling sequence for the multiply or divide subroutine is similar to the entry for the hardware multiply or divide instruction. These entries or calling sequences are outlined for the respective instruction below.

For both multiply and divide, the double-word fixed-point data format is:



FIXED POINT MULTIPLY 10 μsec

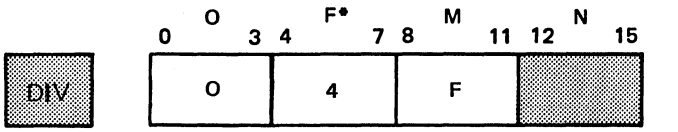
Calling sequence or entry:

P MPY
P+1 JMM MPYX
P+2 PTR multiplier
P+3 PTR least sig. half of product

The contents of the A register (multiplicand) are multiplied by the multiplier (pointed to by P+2). The most significant half of the product (15 bits plus sign) replaces the contents of the A register. The least significant half of the product (15 bits plus sign) replace the contents of the word pointed to by P+3.

Symbolically, the Multiply instruction performs the following:
(A)* ((P+2))→(A) and ((P+3))

The sign of the product is the sign of the A register and also that of ((P+3)).



FIXED POINT DIVIDE 14 μsec

Calling sequence or entry:

P DIV
P+1 JMM DIVX
P+2 PTR least sig. half of dividend (also R)
P+3 PTR divisor

The contents of the A register and the contents of the word pointed to by P+2 represent a double-word dividend. Together, they are divided by the word pointed to by P+3 (divisor). The quotient replaces the contents of the A register, and the remainder replaces the least significant half of the dividend (pointed to by P+2).

Symbolically, the Divide instruction performs the following:
((A) and ((P+2)))/((P+3))→(A),Remainder→((P+2))

The sign of the remainder is the same as the sign of the original dividend (unless the remainder is zero). Negative quotients are represented in two's complement form in the A register.

Chapter 4

Multi-level Control Program

MULTI-LEVEL PROGRAM CONTROL

The MAC 16 computer has a multi-level program interrupt feature that is unique to computers in its class. For each interrupt signal provided there exists a memory storage area for all data from the processor registers. If an interrupt signal occurs of a higher priority than the currently active program, MAC 16 automatically stores the content of the current processor registers in an Executive Page in memory and restores the data for the hardware registers for the higher priority program. This register and memory exchange occurs within six microseconds after the interrupt signal recognition.

This multi-level program control provides an auto-monitor capability that minimizes requirements for a real-time monitor program, and simplifies control of communication with input/output devices on the Programmed Data Channel. Multi-level program control occurs when device requests for service are wired as interrupt signals.

Each interrupt signal initiates a response to a program level. A level corresponds to the number of the interrupt signals (0 to 63) with the highest number receiving the highest priority. The interrupt signals can be expanded in increments of 4, to a maximum of 64 levels. Only one program, the highest level, can be active at a time. Each program level is terminated by a special instruction that re-initializes the Program Location Register and the next highest program level is automatically entered. Higher priority programs are entered as the result of external interrupt signals or by programmed operation.

Instructions are provided for the programmer to control the multi-level program feature of the MAC 16. These include the ability to inhibit response to all interrupts, to reset interrupt signals individually, and to examine the current level number. The Executive Page, which contains a copy of all currently inactive program registers, is accessible for modification by the program.

Executive Page

The low numbered memory addresses of page zero (memory addresses 000 to 255_{10}) are reserved for word-storage of each Program Level Register. The number of word locations reserved is a function of the number of interrupt levels provided in the system. Four word locations are reserved for each program level. The basic MAC 16 provides four interrupt program levels. Therefore, addresses 000 to 015_{10} are reserved in basic systems; if 32 interrupt levels are provided, addresses 000 to 127_{10} are reserved for the Executive Page. Unused portions of the Executive Page are available for general use.

Three of the four words of each program level are for register storage and the fourth word contains program status bits and control flip-flops. The four words are as follows:

Word Number	Word
0	Program Location Register, P.1
1	Index Register, X.
2	A Register.
3	Status Bits: Carry C, Overflow V, Base Page Control B, Interrupt Control H, R and S Status indicators.

Program Level #		Executive Page Hexidecimal Address and Contents			
Decimal	Hex.				
0 0	00	0000 P	0001 X	0002 A	0003 Status
0 1	01	0004 P	0005 X	0006 A	0007 Status
0 2	02	0008 P	0009 X	000A A	000B Status
0 3	03	000C P	000D X	000E A	000F Status
0 4	04	0010 P	0011 X	0012 A	0013 Status
6 2	3E	00F8 P	00F9 X	00FA A	00FB Status
6 3	3F	00FC P	00FD X	00FE A	00FF Status

REGISTER STORAGE IN EXECUTIVE PAGE

STATUS						WORD	
0	1	2	3	4	5	6	15
C	V	R	S	B	H		

The address of these words is formed by placing the program level number, LP, in bit positions 8 through 13 of an address word and the word number desired in bit positions 14 and 15. Bit positions 0 through 7 of the address are set to zero.

Words 0 and 2 for each inactive program level are copies of the computer hardware registers P and A. These two words do not represent the current status of these registers for the active program.

Word 1, the Index Register, does not have a corresponding hardware register and is therefore, always current in memory for each program level.

Word 3 contains a copy of the Carry Indicator, C; Overflow Indicator, V; Status Indicator, R; Status Indicator, S; Base Page Control flip-flop, B; and the Interrupt Inhibit flip-flop, H for each program level. The status indicators occupy bit positions 0, 1, 2, 3, 4 and 5 of word 3, respectively. These bit positions represent the indicator status of inactive programs when the program was last terminated or interrupted. These bit positions do not represent the state of flip-flops for the currently active program level. The test and modify instructions for these indicators affect the hardware flip-flops of the active program.

Program Level Register, L.

The Program Level Register, L consists of at least four (basic) and up to 64 bits. These bits are numbered 0 through 63_{10}. The highest numbered bit corresponds to the highest priority interrupt. These bits can be set to a one by an external signal or by execution of a TAL instruction (Transfer A to Interrupt Priority Level). These bits are reset by the CPU reset switch. Only the bit of the L Register belonging to the active priority level is reset by the JRL instruction (Jump and Reset Level).

The priority detect and encode logic continually monitors the L Register to determine if an interrupt level flip-flop with a bit position, or number, that is higher than the current program level number has been set, or if the interrupt level flip-flop associated with the active program has been reset by a JRL instruction. If either of these conditions are detected, an automatic program level change occurs at the end of the current instruction.

An Inhibit Interrupt Control Indicator, H, is provided so that level change can be postponed under program control. The SEX, H instruction (Set Interrupt Inhibit), is used to set this flip-flop, and the REX, H instruction (Reset Interrupt Inhibit) is used to reset this flip-flop. Setting of priority level bits in the L Register is not inhibited by this control, so that any interrupt signals received while program level change was temporarily inhibited are processed as soon as H is reset.

The programmer can control the external setting of interrupt level bits in the L Register by commands to device controllers that have the ability to send interrupt signals. Each device controller can be commanded to not interrupt with an ECO (External Command Output) instruction.

If no interrupt level bits in the L Register are set to one, the computer enters a quiescent state and remains in this state until a bit in the L Register becomes a one.

Chapter 5 Input/Output System

INPUT/OUTPUT CHANNELS

The MAC 16 Computer system has provisions for three types of input/output (I/O) channels. Each I/O channel permits a progressively higher system performance by providing higher degrees of independent and automatic control of data communication. These channels are the Programmed Data Channel, PDC, and as options, the Multiplexed Data Channel, MDC, and Direct Memory Access Channel DMA. The Programmed Data Channel, figure 5-1, is basic to all MAC 16 systems, and provides data communication under program execution.

The MDC channel provides communication with up to 16 device controllers independent of program execution through the use of channel control words. The DMA channel allows device controllers to transfer data directly from or to memory.

The I/O channels take advantage of the priority interrupt capability. Whenever a device controller desires computer programmed service, the device transmits an interrupt signal to the computer. If the interrupt signal is of higher priority than any other current interrupt signal received, the device causes automatic service of the interrupt request. For more detailed information on the optional channels see Programmer's Manual.

Programmed Data Channel

The PDC is a set of data input/output lines, address lines, and command lines that can be shared with several device controllers. A device controller is a special unit designed to provide buffered communication and control for an I/O device. The device controller is designed to meet the PDC interface requirements. This is generally a simple unit containing a byte or word buffer register, a command register, and a status register. The PDC provides the device controller with timing signals.

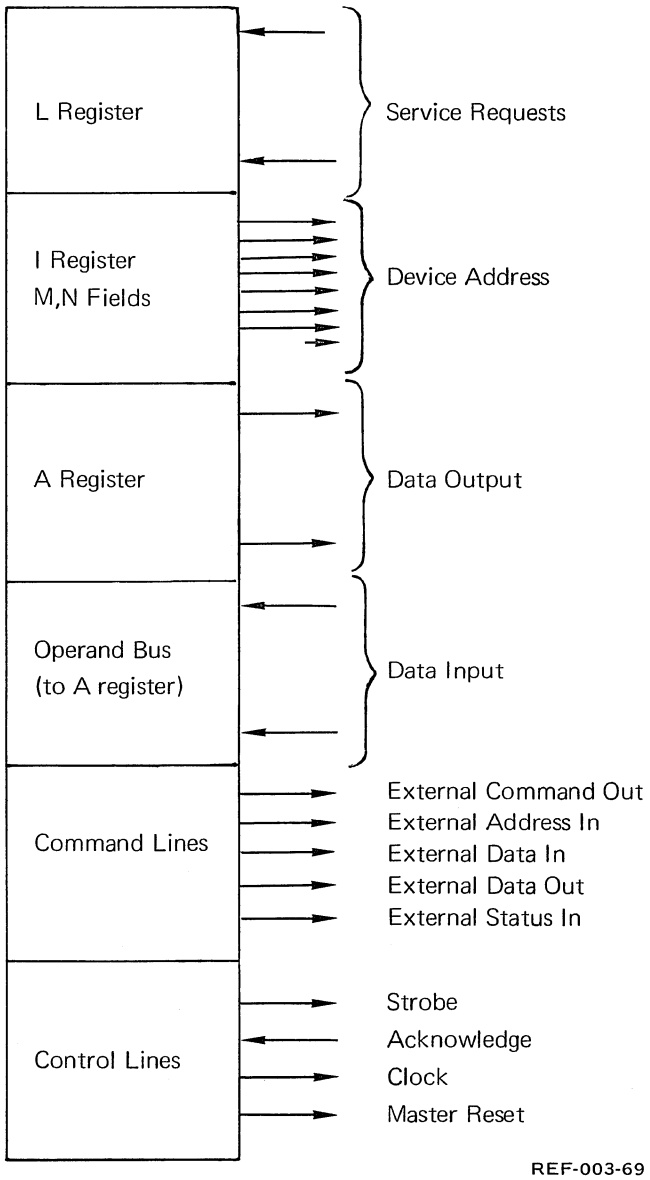


Figure 5-1. Programmed Data Channel

Control and operation of the PDC is by execution of five input/output instructions:

1. External Command Out and Skip, ECO.
2. External Data Out and Skip, EDO.
3. External Data In and Skip, EDI.
4. External Status In, ESI.
5. External Address In, EAI.

External Command Output and Skip Instruction

Execution of the ECO instruction causes the M and N fields of the instruction to be transmitted over the PDC device address lines. The address is accompanied by a "Command Out" and a "Strobe" signal. A device controller recognizes its own address and prepares to receive the content of the 16 "Data Out" lines into its command register. Eight lines or less are sufficient for commands and originate from the least significant, or right byte of the Accumulator. The programmer has previously loaded the Accumulator with the command data that is transmitted on the "Data Out" lines.

When the device controller is not in a state capable of accepting the command, or signals that an abnormal or error condition exists, an "Acknowledge" signal is not returned to the computer during execution of the ECO instruction. This causes the next instruction of the current program level (address stored in P register) to be executed. This instruction could be a jump to a special sequence of instructions. If the "Acknowledge" signal is received during execution of the ECO instruction, the next instruction in sequence is skipped by incrementing the P register and the following instruction is then executed.

The ECO instruction is used to initiate a device controller to receive or transmit a block of data at its own transmission rate. After a device controller is "Initiated to Communicate," and each time the device controller has a word or byte to send or is ready to receive a word or a byte, it signals the computer with a service request. This automatically causes the computer to enter a program level corresponding to the interrupt signal and to execute an input/output service program. Characteristics of the I/O service program are at the programmers option. Typical shared service routines require less than twenty microseconds for each word or byte transmitted.

External Data Output and Skip Instruction

Execution of the EDO instruction causes the M and N fields of the instruction to be transmitted over the PDC "Address Output" lines. The address is accompanied by a "Data Out" and a "Strobe" signal. The content of the previously loaded Accumulator is transmitted on the "Data Out" lines. The device controller can receive up to 16 bits of data. If the controller is for a byte-oriented device, the left byte of the data is ignored.

If the device controller is not ready, has sufficient data, or signals to terminate the data communication for some reason, it does not return an "Acknowledge" signal to the computer during the execution of the EDO instruction. This causes the next instruction of the current program level to be in sequence. The next instruction could be a jump to a special sequence of instructions. If the "Acknowledge" signal is received during execution of the EDO instruction, the next instruction in sequence is skipped.

External Data Input and Skip Instruction

Execution of the EDI instruction causes the M and N fields of the instruction to be transmitted over the PDC address output lines. This address is accompanied by a "Data In" and a "Strobe" signal. The addressed device controller then transmits to the computer up to 16-bits of data on the PDC "Data In" lines which is copied into the Accumulator.

If the device controller wishes to terminate the data communication, or is not ready it does not return an "Acknowledge" signal to the computer during execution of the EDI instruction. This causes the next instruction on this program level to be in sequence. The instruction could be a jump to a special sequence of instructions. If the "Acknowledge" signal is received, the next instruction in sequence is skipped.

External Status Input Instruction

Execution of an ESI instruction is similar to execution of an EDI instruction except that a "Status In" signal is transmitted with the device address to the selected controller. The controller responds by placing status information on the PDC "Data In" lines. This information is subsequently copied into the Accumulator.

Eight Bits or less are generally sufficient for a device to indicate its status.

External Address In Instruction

During execution of the EAI instruction the address output lines of the PDC are ignored. The EAI instruction is used when more than one device controller shares a common service request line to inform the I/O service program which device controller is requesting service. The sharing of interrupt request lines is standard. A device controller must have its own interrupt line if a quicker response time is required.

If an interrupt request line is shared, only one device controller at a time can request service. Each device controller is assigned a precedence, determined by the controllers interconnection on the common interrupt request line. A device controller can detect when no higher precedence controller is using the common interrupt line. Therefore, when an "Address In Signal" is transmitted by the PDC during execution of an EAI, the highest precedence device controller which requests service causes its 8-bit device address to be placed on the "Data In" lines of the PDC. This address is subsequently copied into the right byte position of the Computer Accumulator. The programmer can use the device address to enter the proper I/O service program for the requesting controller.

If a unique interrupt request line is used, an EAI instruction must not be used.

Memoranda

Chapter 6 Control Panel

CONTROL PANEL

The MAC 16 Computer control panel, figure 6-1, includes a full set of control switches and indicators for program checkout and maintenance functions. Indicators permit complete state-of-the-machine display, including hardware registers, program level, and status. Memory read/write, halt-on-address, interrupt control and single step instruction switches facilitate program testing.

Indicators

Memory Address, 16 lamps. Indicates the content of the Memory Address (M) Register. The M register stores the address of the current memory cycle.

DATA, 16 lamps. These indicators are located just below the MEMORY ADDRESS indicators. The data display selectively indicates the content of the D, M, P, A, I and W registers and the data manually entered by the Data Switches.

Q (quiescent), 1 lamp. Signifies that programs are not being executed because all priority requests have been serviced. The computer clock is running but computer is idle waiting for instructions.

PRIORITY LEVEL, 6 lamps. These indicators display the binary number of the current, active program level.

C (carry), 1 lamp. Displays the state of the Carry flip-flop, lighting if there is a carry.

V (overflow), 1 lamp. This indicator displays the state of the Overflow flip-flop, lighting if there is an overflow.

R (program status indicator), 1 lamp. Displays the state of the R flip-flop that is set under program control, lighting when R is set.

S (program status indicator), 1 lamp. Displays the state of the S flip-flop that is set under program control, lighting if S is set.

B (base page control) 1 lamp. Displays the state of the Base Page Control flip-flop that is set under program control, lighting indicates program confined to one base page.

H (interrupt inhibit), 1 lamp. Displays the state of the Interrupt Inhibit flip-flop H, lighting if there is an inhibit interrupt condition.

W (branch), 1 lamp. Displays the condition that a Skip or Jump instruction or an instruction which has a skip condition has been executed. The Program Location (P) Register contains the next instruction address. The Operand (W) Register contains the address of the next instruction if the branch has not occurred.

SEN (sense), 1 lamp. This indicator is located above the SEN switch and displays the set condition of that switch. The indicator is reset when a TSA (Transfer Switches to A) instruction is executed.

HLT (Halt), 1 lamp. This indicator is located above the HLT switch and displays the indication that the computer clock is off.

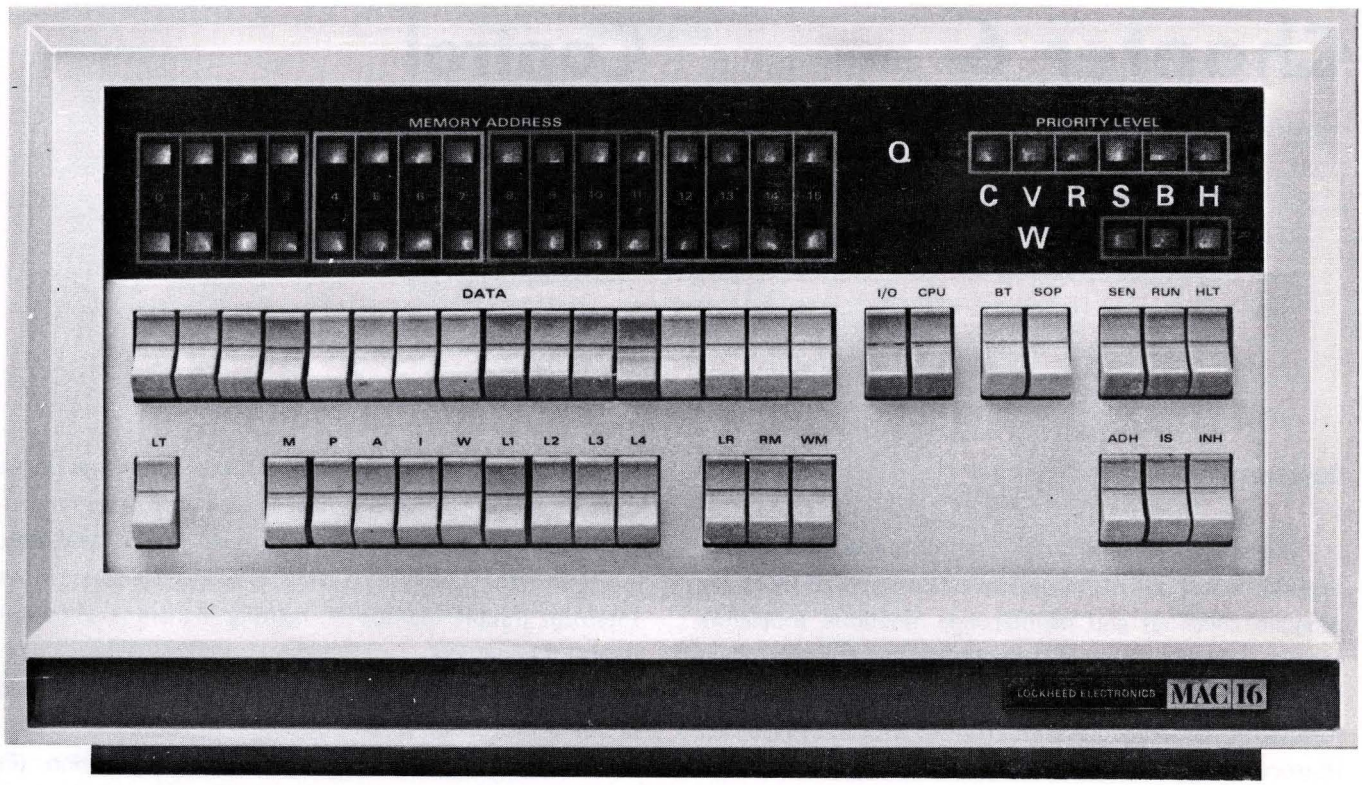
RUN, 1 lamp. This indicator is on when the computer clock is on. The computer may be executing programs or be in the quiescent state.

Switches

DATA, 16 alternate action switches. These switches are used to load selected registers manually with binary data; as 16 sense switches for operator input; or to specify an address when the ADH (address halt) switch is on.

I/O (reset), 1 momentary action switch. When depressed this switch causes a reset signal to be transmitted to all device controllers wired to receive the signal. The switch is inactive in the run mode.

CPU (reset), 1 momentary action switch. When depressed this switch causes the Program Level (L) Register,



LEGEND:

Upper row:

Memory Address indicators (16)
Bit position numbers in hex groupings
Data display indicators (16)

Q: quiescent indicator
Priority Level: six indicators

C,V,R,S,B,H: six status indicators
W: branch indicator
SEN, RUN, HLT indicators

Center row:

Data Switches (16)

I/O: Input/output device controller reset switch
CPU: Processor reset switch

BT: Automatic bootstrap entry switch (optional feature)
SOP: Set zero priority switch

SEN: Sense switch for DATA switches
RUN: Run switch
HLT: Halt switch

Lower row:

LT: Lamp test switch

Register select switches:

M: Memory Address Register
P: Program Location Register
A: Accumulator
I: Instruction Word Register
W: Operand Register
L1: Priority Level, Register 1
L2: Priority Level, Register 2
L3: Priority Level, Register 3
L4: Priority Level, Register 4

LR: LOAD REGISTER
RM: READ MEMORY
WM: WRITE MEMORY
ADH: ADDRESS HALT
IS: INSTRUCTION STEP
INH: INHIBIT INTERRUPT

REF-004-69

Figure 6-1. MAC 16 Control Panel

the Interrupt Inhibit flip-flop H, and internal clock states to be reset. The CPU switch is inactive when the computer is in the run mode.

BT (bootstrap), 1 momentary action switch. When pressed this switch causes a 64-word program to be loaded automatically into memory and executed. The program is a binary data input routine for input of a program to the computer. The switch is inactive when the computer is the run mode.

SOP (set program level 0), 1 momentary action switch. When pressed this switch causes bit position zero of the Program Level Register (L) to be set to a one. If the computer is in the quiescent state it enters program level zero.

SEN (sense), 1 momentary action switch. When pressed, this switch causes the SEN indicator to be on. When the computer executes a TSA instruction the status of the DATA switches are input to the Accumulator and the SEN indicator is reset.

RUN, 1 momentary action switch. This switch is effective only when the Computer clock is halted. It starts the Computer clock and processing starts from current state of the Computer. The Computer can be halted by executing a Halt instruction, pressing the HLT switch, or pressing either the ADH or IS switches on. If the CPU switch is not pressed before the RUN switch, the next instruction in sequence is processed. If the CPU is pressed before RUN and the INH switch is off, the Quiescent state is entered and the Computer waits for a program level interrupt signal. If the INH switch is on, the next instruction addressed by the Program Location Register, P, is executed.

HLT, 1 momentary action switch. Pressing this switch causes the computer to complete the current instruction and stop the computer clock.

LT (lamp test), 1 momentary action switch. When pressed this switch causes all indicators on the control panel to be lighted for test purposes.

Register select, 6 (3 optional) alternate action switches. These switches select the register whose content is to be displayed in the DATA display indicators. The switches are also used to select the register that will be loaded from the 16 DATA switches when the LR (load register) switch is pressed. If two or more switches are on, only the left-most switch is active. If no switches are pressed, the content of the Memory Data Register (D) is displayed. The register select switches are as follows:

(M) Memory Address Register – display or load

(P) Program Location Register – display or load

(A) Accumulator – display or load

(I) Instruction Word Register – display only

(W) Operand Register – display or load

(L1) Priority Level Register 1 (bits 0 through 15 – load only

(L2) Priority Level Register 2 (bits 16 through 31) (optional) – load only

(L3) Priority Level Register 3 (bits 32 through 47) (optional) – load only

(L4) Priority Level Register 4 (bits 48 through 63) (optional) – load only

LR (load register), 1 momentary action switch. When pressed and released this switch causes the content of the DATA entry switches to be copied into the Memory Data Register (D) and to the register specified by the register select switches. The switch is inactive when the computer is in the run mode.

RM (read memory), 1 momentary action switch. When pressed and released this switch causes the content of the memory location specified by the Memory Address Register (M) to be copied into the Memory Data Register (D). The memory location specified by the M Register is then automatically incremented by one. The switch is inactive when the computer is in the run mode.

WM (write memory), 1 momentary action switch. When pressed and released this switch causes the content of the memory location specified by the Memory Address Register (M) to be replaced with the data specified by the DATA switches. The memory location specified by the M Register is also changed to contain a copy of the DATA switches. The switch is inactive when the computer is in the run mode.

Note

The LR/RM/WM switches are such that the left-most switch takes priority. For example if LR and WM were pressed in that order, only LR would activate.

ADH (address halt), 1 alternate action switch. This switch causes the computer to halt if the instruction address or effective address is equal to the setting of the DATA switches.

IS (instruction step), 1 alternate action switch. This switch causes the computer to halt after executing each instruction. Pressing this switch while the computer is in the run mode causes the computer to halt.

INH (inhibit interrupt), 1 alternate action switch. When on, this switch prevents a program level change independent of the state of the Interrupt Inhibit control (H) and causes the H indicator to light.

Appendix A Functional Listing of MAC 16 Instructions

Mnemonic	Hex Op Code	Instruction Description	Time in μs	Operation	Page No.
	LOAD AND STORE INSTRUCTIONS				
LDA	D000	Load	2	(Y) $\longrightarrow$ A	3-2
LDX	C000	Load Index	3	(Y) $\longrightarrow$ X	3-2
LAX	070N	Load X + N	3	(X+N) $\longrightarrow$ A	3-2
LIX	030N	Load (X) + N	4	((X)+N) $\longrightarrow$ A	3-2
LDI	0DMN	Load Immediate	1	M,N $\longrightarrow$ A_{8-15},0 $\longrightarrow$ A_{0-7}	3-2
LSB	056X	Load Status Bits	1	CVRSBH $\longrightarrow$ A_{0-5},0 $\longrightarrow$ A_{6-15}	3-3
STA	6000	Store	2	A $\longrightarrow$ (Y)	3-3
STL	3000	Store Byte, Left	2	A_{8-15} $\longrightarrow$ $(Y)_{0-7}$	3-3
STR	7000	Store Byte, Right	2	A_{8-15} $\longrightarrow$ $(Y)_{8-15}$	3-3
STX	2000	Store Index	3	X $\longrightarrow$ (Y)	3-3
SAX	074N	Store X + N	3	A $\longrightarrow$ (X+N)	3-3
SIX	034N	Store (X) + N	4	A $\longrightarrow$ ((X) + N)	3-4
SSB	057X	Store Status Bits	1	A_{0-5} $\longrightarrow$ CVRSBH	3-4
	ARITHMETIC INSTRUCTIONS				
ADD	8000	Add	2	A + (Y) $\longrightarrow$ A	3-4
ADI	08MN	Add Immediate	2	A + M,N $\longrightarrow$ A	3-4
ADC	01CX	Add Carry	2	A + C $\longrightarrow$ A	3-4
SUB	9000	Subtract	2	A – (Y) $\longrightarrow$ A	3-4
SBI	09MN	Subtract Immediate	2	A – M,N $\longrightarrow$ A	3-5

Mnemonic	Hex Op Code	Instruction Description	Time in μs	Operation	Page No.
	ARITHMETIC INSTRUCTIONS (CONT)				
TWA	010X	Two's Complement	2	$2^{16}-A \longrightarrow A$	3-5
ABA	014X	Absolute Value	2	$\|A\| \longrightarrow A$	3-5
	LOGIC INSTRUCTIONS				
ANA	B000	AND	2	$A \cap (Y) \longrightarrow A$	3-5
ORA	A000	OR	2	$A \cup (Y) \longrightarrow A$	3-5
ONA	018X	One's Complement	2	$A \longrightarrow \overline{A}$	3-5
	SHIFT INSTRUCTIONS				
ALS	0C4N	Arithmetic Left, Save Carry	2-5	$A_{i+1} \longrightarrow A_i, 0 \longrightarrow A_{15}$ $i=1\text{-}14, A_1 \longrightarrow C$, Set V if $A_1 \neq A_0$	3-6
ALI	0C5N	Arithmetic Left, Insert Carry	2-5	$A_{i+1} \longrightarrow A_i, C \longrightarrow A_{15}$ $i=1\text{-}14, A_1 \longrightarrow C$, Set V if $A_1 \neq A_0$	3-6
ARS	0CCN	Arithmetic Right, Save Carry	2-5	$A_i \longrightarrow A_{i+1}, A_0 \longrightarrow A_1$ $i=1\text{-}14, A_{15} \longrightarrow C$	3-6
ARI	0CDN	Arithmetic Right, Insert Carry	2-5	$A_i \longrightarrow A_{i+1}, C \longrightarrow A_1$ $i=1\text{-}14, A_{15} \longrightarrow C$	3-6
LLN	0CON	Logical Left, No Carry	2-5	$A_{i+1} \longrightarrow A_i, 0 \longrightarrow A_{15}$ $i=0\text{-}14$	3-6
LLO	0CIN	Logical Left Open	2-5	$A_{i+1} \longrightarrow A_i, 0 \longrightarrow A_{15}$ $i=0\text{-}14, A_0 \longrightarrow C$	3-6
LLC	0C3N	Logical Left Closed	2-5	$A_{i+1} \longrightarrow A_i, A_0 \longrightarrow A_{15}$ $i=0\text{-}14, A_0 \longrightarrow C$	3-7
LLI	0C2N	Logical Left, Insert Carry	2-5	$A_{i+1} \longrightarrow A_i, C \longrightarrow A_{15}$ $i=0\text{-}14, A_0 \longrightarrow C$	3-7
LRN	0C8N	Logical Right, No Carry	2-5	$A_i \longrightarrow A_{i+1}, 0 \longrightarrow A_0$ $i=0\text{-}14$	3-7
LRO	0C9N	Logical Right Open	2-5	$A_i \longrightarrow A_{i+1}, 0 \longrightarrow A_0$ $i=0\text{-}14, A_{15} \longrightarrow C$	3-7

Mnemonic	Hex Op Code	Instruction Description	Time in μs	Operation	Page No.
	SHIFT INSTRUCTIONS (CONT)				
LRC	0CBN	Logical Right Closed	2-5	$A_i \longrightarrow A_{i+1}, A_{15} \longrightarrow A_0$ $i=0\text{-}14, A_{15} \longrightarrow C$	3-7
LRI	0CAN	Logical Right, Insert Carry	2-5	$A_i \longrightarrow A_{i+1}, C \longrightarrow A_0$ $i=0\text{-}14, A_{15} \longrightarrow C$	3-7
	JUMP INSTRUCTIONS				
JMP	5000	Jump Unconditional	2	$Y \longrightarrow P$	3-8
JMM	4000	Jump, Mark	3	$P + 1 \longrightarrow X, Y \longrightarrow P$	3-8
JMX	07CN	Jump to X + N	3	$X + N \longrightarrow P$	3-8
JIX	03CN	Jump Indirect To (X) + N	4	$(X) + N \longrightarrow P$	3-8
JMA	047X	Jump to Accumulator	2	$A \longrightarrow P, P+1 \longrightarrow A$	3-8
JRL	1000	Jump, Reset Level	2	$Y \longrightarrow P, O \longrightarrow L;$	3-8
	TEST AND SKIP INSTRUCTIONS				
SKP	048N	Skip Unconditional	2	$P+1+N \longrightarrow P$	3-9
SNC	040N	Skip, No Carry	2	$P+1+N \longrightarrow P$ if C=0	3-9
SNV	041N	Skip, No Overflow	2	$P+1+N \longrightarrow P$ if V=0	3-9
SNH	045N	Skip, No Interrupt Inhibit	2	$P+1+N \longrightarrow P$ if H=0	3-9
SNB	044N	Skip, No Base Page Control	2	$P+1+N \longrightarrow P$ if B=0	3-9
SNR	042N	Skip If R Zero	2	$P+1+N \longrightarrow P$ if R=0	3-9
SNS	043N	Skip If S Zero	2	$P+1+N \longrightarrow P$ if S=0	3-9
SKN	049N	Skip If Normalized	2	$P+1+N \longrightarrow P$ if $A_0=A_1=A_2=A_3=A_4$	3-9
SAN	04BN	Skip If Negative	2	$P+1+N \longrightarrow P$ if $A_0=1$	3-10
SAZ	04AN	Skip If Zero	2	$P+1+N \longrightarrow P$ if A=0	3-10
SAG	04DN	Skip If Greater Than Zero	2	$P+1+N \longrightarrow P$ if A>0	3-10
SLZ	04CN	Skip If LSB is Zero	2	$P+1+N \longrightarrow P$ if $A_{15}=0$	3-10
SKX	02CN	Skip If Index Zero	3	$P+1+N \longrightarrow P$ if X=0	3-10

Mnemonic	Hex Op Code	Instruction Description	Time in μs	Operation	Page No.
	INDEX INSTRUCTIONS				
INX	020N	Increment Index And Skip	3	X+N ⟶ X	3-10
DNX	024N	Decrement Index And Skip	3	X−N ⟶ X	3-11
	COMPARE INSTRUCTION				
CAA	F000	Compare Arithmetic And Skip	3	P+1 ⟶ P if A>(Y) P+2 ⟶ P if A<(Y) P+3 ⟶ P if A=(Y)	3-11
	INCREMENT MEMORY INSTRUCTION				
INC	E000	Increment Memory And Skip	3	(Y) + 1 ⟶ (Y)	3-11
	REGISTER INSTRUCTIONS				
CLA	054X	Clear Accumulator	1	0 ⟶ A	3-11
XXA	060X	Exchange Index With Accumulator	3	X ⟶ A, A ⟶ X	3-11
TSA	05AX	Transfer Data Switches	1	SW ⟶ A if SEN = 1	3-12
TLA	046X	Transfer Program Level	2	LP ⟶ A_{8-13}	3-12
	CONTROL INSTRUCTIONS				
NOP	0480	No Operation	2	P+1 ⟶ P	3-12
HLT	00XX	Halt	1		3-12
TAL	058N	Transfer To Program Level Register	1	AUL ⟶ $L_{1,2,3, \text{ or } 4}$	3-12
SEX	050N thru 053N	Set Indicators	1	M,N_{10-15} ⟶ CVRSBH	3-12
REX	05CN thru 05FN	Reset Indicators	1	M,N_{10-15} ⟶ CVRSBH	3-13
	DEVICE CONTROL INSTRUCTIONS				
ECO	0FMN	External Command Out	3	A ⟶ DC	3-13
ESI	OEMN	External Status In	3	Device Status ⟶ A	3-13
EAI	06CX	External Address In	3	Device Address ⟶ A_{8-15}, Zero ⟶ A_{0-7}	3-13

Mnemonic	Hex Op Code	Instruction Description	Time in μs	Operation	Page No.
	INPUT/OUTPUT INSTRUCTIONS				
EDI	OAMN	External Data In	3	Device Controller ⟶ A	3-13
EDO	OBMN	External Data Out	3	A ⟶ Device Controller	3-13

Memory reference instruction execution timing is increased one microsecond for each of the following functions:

I=1	Indirect addressing; add 1 μs
X=1	Indexing; add 1 μs
P=0	Base page addressing; add 1 μs

Instruction execution time for shift instructions depends upon the number of bit positions shifted as specified in the N field of the instruction.

If N is 0 to 4, shift is 2 microseconds.
5 to 8, shift is 3 microseconds.
9 to 12, shift is 4 microseconds.
13 to 15, shift is 5 microseconds.

Memoranda

Appendix B Mnemonic Listing of MAC 16 Instructions

Mnemonic	Hex Op Code	Instruction Description	Time μsec	Page No.
ABA	014X	Absolute Value	2	3-5
ADC	01CX	Add Carry	2	3-4
ADD	8000	Add	2	3-4
ADI	08MN	Add Immediate	2	3-4
ALI	0C5N	Arithmetic Left, Insert Carry	2-5	3-6
ALS	0C4N	Arithmetic Left, Save Carry	2-5	3-6
ANA	B000	AND	2	3-5
ARI	0CDN	Arithmetic Right, Insert Carry	2-5	3-6
ARS	0CCN	Arithmetic Right, Save Carry	2-5	3-6
CAA	F000	Compare Arithmetic and Skip	3	3-11
CLA	054X	Clear Accumulator	1	3-11
DNX	024N	Decrement Index And Skip	3	3-11
EAI	06CX	External Address In	3	3-13
ECO	0FMN	External Command Out	3	3-13
EDI	0AMN	External Data In	3	3-13
EDO	0BMN	External Data Out	3	3-13
ESI	0EMN	External Status In	3	3-13
HLT	00XX	Halt	1	3-12
INC	E000	Increment Memory And Skip	3	3-11
INX	020N	Increment Index And Skip	3	3-10
JIX	03CN	Jump Indirect To (X) + N	4	3-8

Mnemonic	Hex Op Code	Instruction Description	Time μsec	Page No.
JMA	047X	Jump to Accumulator	2	3-8
JMM	4000	Jump, Mark	3	3-8
JMP	5000	Jump Unconditional	2	3-8
JMX	07CN	Jump To X + N	3	3-8
JRL	1000	Jump, Reset Level	2	3-8
LAX	070N	Load X + N	3	3-2
LDA	D000	Load	2	3-2
LDI	0DMN	Load Immediate	1	3-2
LDX	C000	Load Index	3	3-2
LIX	030N	Load (X) + N	4	3-2
LLC	0C3N	Logical Left Closed	2-5	3-7
LLI	0C2N	Logical Left, Insert Carry	2-5	3-7
LLN	0C0N	Logical Left, No Carry	2-5	3-6
LLO	0CIN	Logical Left Open	2-5	3-6
LRC	0CBN	Logical Right Closed	2-5	3-7
LRI	0CAN	Logical Right, Insert Carry	2-5	3-7
LRN	0C8N	Logical Right, No Carry	2-5	3-7
LRO	0C9N	Logical Right Open	2-5	3-7
LSB	056X	Load Status Bits	1	3-3
NOP	0480	No Operation	2	3-12
ONA	018X	One's Complement	2	3-5
ORA	A000	OR	2	3-5
REX	05CN thru 05FN	Reset Indicators	1	3-13
SAG	04DN	Skip If Greater Than Zero	2	3-10
SAN	04BN	Skip If Negative	2	3-10
SAX	074N	Store X + N	3	3-3
SAZ	04AN	Skip If Zero	2	3-10

Mnemonic	Hex Op Code	Instruction Description	Time μsec	Page No.
SBI	09MN	Subtract Immediate	2	3-5
SEX	050N thru 053N	Set Indicators	1	3-12
SIX	034N	Store (X) + N	4	3-4
SKN	049N	Skip If Normalized	2	3-9
SKP	048N	Skip Unconditional	2	3-9
SKX	02CN	Skip If Index Zero	3	3-10
SLZ	04CN	Skip If LSB Is Zero	2	3-10
SNB	044N	Skip, No Base Page Control	2	3-9
SNC	040N	Skip, No Carry	2	3-9
SNH	045N	Skip, No Interrupt Inhibit	2	3-9
SNR	042N	Skip If R Zero	2	3-9
SNS	043N	Skip If S Zero	2	3-9
SNV	041N	Skip, No Overflow	2	3-9
SSB	057X	Store Status Bits	1	3-4
STA	6000	Store	2	3-3
STL	3000	Store Byte, Left	2	3-3
STR	7000	Store Byte, Right	2	3-3
STX	2000	Store Index	3	3-3
SUB	9000	Subtract	2	3-4
TAL	058N	Transfer To Program Level Register	1	3-12
TLA	046X	Transfer Program Level	2	3-12
TSA	05AX	Transfer Data Switches	1	3-12
TWA	010X	Two's Complement	2	3-5
XXA	060X	Exchange Index With Accumulator	3	3-11

Memoranda

Appendix C Hexadecimal Listing of MAC 16 Instructions

Hex Op Code	Mnemonic Op Code	Instruction Description	Time μsec	Page No.
1000	JRL	Jump, Reset Level	2	3-8
2000	STX	Store Index	3	3-3
3000	STL	Store Byte, Left	2	3-3
4000	JMM	Jump, Mark	3	3-8
5000	JMP	Jump Unconditional	2	3-8
6000	STA	Store	2	3-3
7000	STR	Store Byte, Right	2	3-3
8000	ADD	Add	2	3-4
9000	SUB	Subtract	2	3-4
A000	ORA	OR	2	3-5
B000	ANA	AND	2	3-5
C000	LDX	Load Index	3	3-2
D000	LDA	Load	2	3-2
E000	INC	Increment Memory and Skip	3	3-11
F000	CAA	Compare Arithmetic and Skip	3	3-11
00XX	HLT	Halt	1	3-12
010X	TWA	Two's Complement	2	3-5
014X	ABA	Absolute Value	2	3-5
018X	ONA	One's Complement	2	3-5
01CX	ADC	Add Carry	2	3-4

Hex Op Code	Mnemonic Op Code	Instruction Description	Time μsec	Page No.
020N	INX	Increment Index and Skip	3	3-10
024N	DNX	Decrement Index and Skip	3	3-11
02CN	SKX	Skip If Index Zero	3	3-10
030N	LIX	Load (X) + N	4	3-2
034N	SIX	Store (X) + N	4	3-3
03CN	JIX	Jump Indirect to (X) + N	4	3-8
040N	SNC	Skip, No Carry	2	3-9
041N	SNV	Skip, No Overflow	2	3-9
042N	SNR	Skip If R Zero	2	3-9
043N	SNS	Skip If S Zero	2	3-9
044N	SNB	Skip, No Base Page Control	2	3-9
045N	SNH	Skip, No Interrupt Inhibit	2	3-9
046X	TLA	Transfer Program Level	2	3-12
047X	JMA	Jump to Accumulator	2	3-8
0480	NOP	No Operation	2	3-12
048N	SKP	Skip Unconditional	2	3-9
049N	SKN	Skip If Normalized	2	3-9
04AN	SAZ	Skip If Zero	2	3-10
04BN	SAN	Skip If Negative	2	3-10
04CN	SLZ	Skip If LSB Is Zero	2	3-10
04DN	SAG	Skip If Greater Than Zero	2	3-10
050N thru 053N	SEX	Set Indicators	1	3-12
054X	CLA	Clear Accumulator	1	3-11
056X	LSB	Load Status Bits	1	3-3
057X	SSB	Store Status Bits	1	3-4
058N	TAL	Transfer To Program Level Register	1	3-12
05AX	TSA	Transfer Data Switches	1	3-11

Hex Op Code	Mnemonic Op Code	Instruction Description	Time μsec	Page No.
05CN thru 05FN	REX	Reset Indicators	1	3-14
060X	XXA	Exchange Index With Accumulator	3	3-11
06CX	EAI	External Address In	3	3-13
070N	LAX	Load X + N	3	3-2
074N	SAX	Store X + N	3	3-3
07CN	JMX	Jump To X + N	3	3-8
08MN	ADI	Add Immediate	2	3-4
09MN	SBI	Subtract Immediate	2	3-5
0AMN	EDI	External Data In	3	3-13
0BMN	EDO	External Data Out	3	3-13
0C0N	LLN	Logical Left No Carry	2-5	3-6
0C1N	LLO	Logical Left Open	2-5	3-6
0C2N	LLI	Logical Left, Insert Carry	2-5	3-7
0C3N	LLC	Logical Left Closed	2-5	3-7
0C4N	ALS	Arithmetic Left, Save Carry	2-5	3-6
0C5N	ALI	Arithmetic Left, Insert Carry	2-5	3-6
0C8N	LRN	Logical Right, No Carry	2-5	3-7
0C9N	LRO	Logical Right Open	2-5	3-7
0CAN	LRI	Logical Right, Insert Carry	2-5	3-7
0CBN	LRC	Logical Right Closed	2-5	3-7
0CCN	ARS	Arithmetic Right, Save Carry	2-5	3-6
0CDN	ARI	Arithmetic Right, Insert Carry	2-5	3-6
0DMN	LDI	Load Immediate	1	3-2
0EMN	ESI	External Status In	3	3-13
0FMN	ECO	External Command Out	3	3-13

Memoranda

Appendix D USAC11 Character Set and Hexadecimal Codes

HEX	CHARACTER
A0	space
A1	!
A2	"
A3	
A4	$
A5	%
A6	&
A7	' (apostrophe)
A8	(
A9	)
AA	*
AB	+
AC	, (comma)
AD	-
AE	. (period)
AF	/
B0	0
B1	1
B2	2
B3	3
B4	4
B5	5
B6	6
B7	7
B8	8
B9	9
BA	:
BB	;
BC	less than
BD	=
BE	greater than
BF	?
C0	@

HEX	CHARACTER
C1	A
C2	B
C3	C
C4	D
C5	E
C6	F
C7	G
C8	H
C9	I
CA	J
CB	K
CC	L
CD	M
CE	N
CF	O
D0	P
D1	Q
D2	R
D3	S
D4	T
D5	U
D6	V
D7	W
D8	X
D9	Y
DA	Z
DB	left bracket
DC	back slash
DD	right bracket
DE	up arrow
DF	left arrow
87	bell
8A	line feed
8D	carriage return

Memoranda

Appendix E Arithmetic References

TABLE OF POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 45
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5

Memoranda

Appendix F Table of Powers of Sixteen

16^n							n
						1	0
						16	1
						256	2
					4	096	3
					65	536	4
				1	048	576	5
				16	777	216	6
				268	435	456	7
			4	294	967	296	8
			68	719	476	736	9
		1	099	511	627	776	10
		17	592	186	044	416	11
		281	474	976	710	656	12
	4	503	599	627	370	496	13
	72	057	594	037	927	936	14
1	152	921	504	606	846	976	15

Decimal Values

Memoranda

Computer Reference Manual

Lockheed Electronics Company
Data Products Division
6201 East Randolph Street
Los Angeles, California, U.S.A. 90022
(213) 722-6810 TWX 910-580-3623
A division of Lockheed Aircraft Corporation