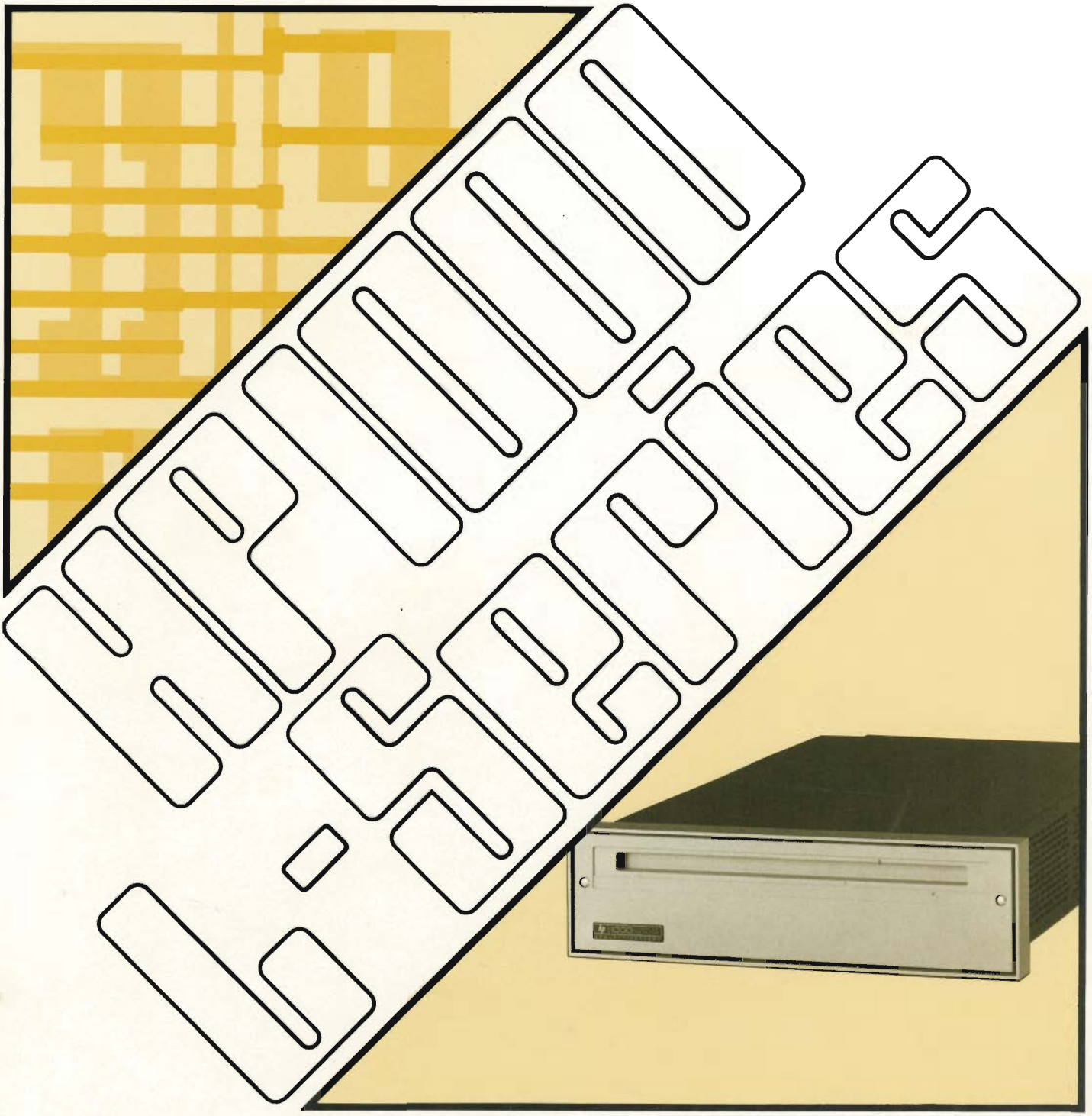
HEWLETT PACKARD

HP 1000 L-Series Computer

Reference Manual



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HP 1000 L-Series Computer

Reference Manual

HEWLETT PACKARD

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

Library Index Number
1000L.320.02103-90007

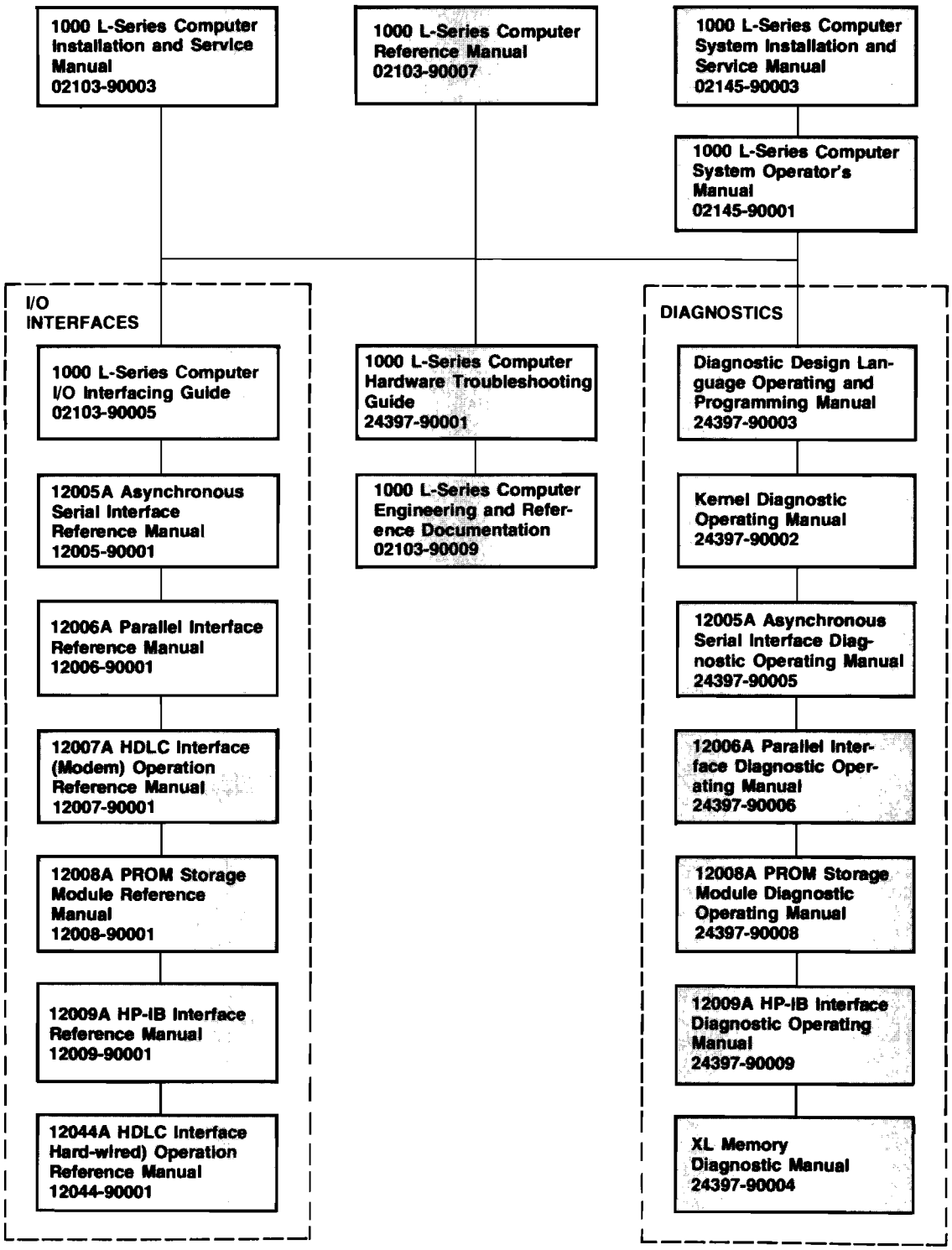
MANUAL PART NO. 02103-90007
Printed in U.S.A. October 1980

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages contain replacement pages or write-in instructions to be merged into the manual by the customer. Manuals will be reprinted as necessary to incorporate all prior updates. A reprinted manual is identical in content (but not in appearance) to the previous edition with all updates incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

First Edition ..... January 1980	
Second Edition ..... October 1980	**Incorporated update 1 and added expanded memory (XL) information.**

DOCUMENTATION MAP



7700-479-1

CONTENTS

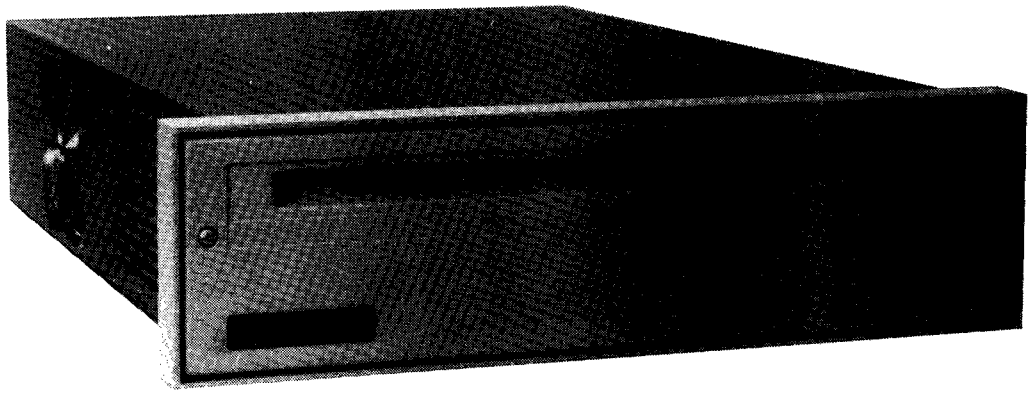
CONTENTS (Continued)

ILLUSTRATIONS

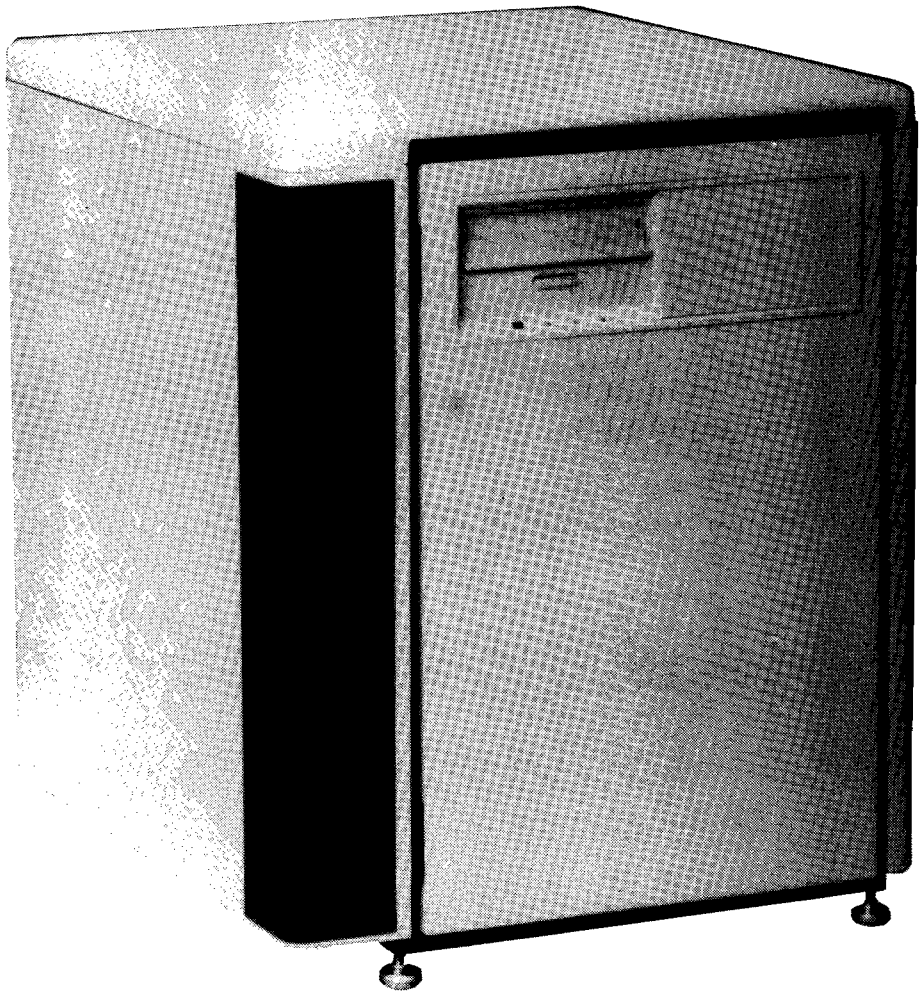
TABLES

ALPHABETICAL INDEX OF INSTRUCTIONS

This Page Is Intentionally Blank.



HP 2103L Computer



L-Series Computer System

7700-483

Figure 1-1. HP 1000 L-Series Computers

SECTION I
GENERAL FEATURES

The HP 1000 L-Series Computer and Computer System (hereafter referred to as L-Series computers) are the low-cost extension to the HP 1000 Computer family. The L-Series computers are designed to deliver full minicomputer power to a variety of cost-critical applications. The L-Series computer hardware is available as printed circuit assemblies (boards), computer units (boxes), and computer systems. (See figure 1-1.)

1-1. ARCHITECTURE

The L-series computer architecture is based on a "distributed intelligence" design utilizing two custom silicon-on-sapphire (SOS) integrated circuit chips. The central processor unit (CPU) chip executes a powerful subset of the HP 1000 instruction set. In conjunction with other logic on the processor card, the CPU chip performs several system level functions, including memory protect, power fail/auto restart, time base generation, parity error interrupt, and extensive self-tests.

All input/output instructions are executed by input/output (I/O) processor chips that reside on the individual I/O interface cards. Since a common backplane links the processor, memory, and I/O cards, the I/O chips monitor the flow of instructions to the central processor but can execute only those that apply to I/O. Because each I/O card is capable of operating independently of the CPU, the L-Series can perform direct memory access (DMA) I/O transfers very efficiently. An I/O card interacts with the CPU only on DMA initiation and completion; beyond that, the entire high-speed transfer is handled by the I/O card, leaving the CPU free to work on other tasks. This achieves significant gains in CPU throughput. Figure 1-2 is a simplified block diagram of the L-Series computer.

1-2. VIRTUAL CONTROL PANEL

The virtual control panel (VCP) program is an interactive program that enables an external device (such as a terminal) to control the CPU in a manner similar to a conventional computer control panel and also provides additional features. That is, it allows the operator to access the various registers (A, B, P, etc.), examine or change memory, and control execution of a program. The VCP program is stored in read-only memory (ROM) on the processor card. In a typical application, the VCP could be an HP 2621 Terminal interfaced by an HP 12005A Asynchronous Serial Interface Card.

1-3. BOOTSTRAP LOADERS

There are several bootstrap loaders stored in ROM on the processor card. The loaders provide program loading from several sources including disc drives, PROM cards, and HP mini-cartridge tapes and a DS/1000 network link. The first three loaders can be selected for auto-boot by switches on the processor card; any of the loaders can be selected by operator commands via the virtual control panel.

1-4. SELF-TEST ROUTINES

Two self-test routines are standard in the L-Series computer. One of these routines is built into the central processor unit (CPU) chip and the other is stored in read-only memory (ROM) on the processor card. These routines are executed whenever computer power is turned on, providing a convenient confidence-check of the processor card, memory cards, and part of the logic on each input/output card. Execution of these routines can also be initiated by a switch on the processor card.

1-5. TIME BASE GENERATOR

The processor card includes a time base generator which may be used to time external events or to create a real-time clock in software. The time base generator can generate an interrupt every 10 milliseconds but is disabled at power-up. The TBG can be enabled and disabled by standard I/O instructions.

1-6. POWER SUPPLY

L-Series computers have a power supply designed to continue normal operation in environments where ac line power may fluctuate widely. Input line voltages and frequencies may vary widely without affecting the operation of the computer. In addition to supplying dc voltages to the plug-in cards, the power supply also outputs two 25-kHz voltages that can be rectified at the load and used to power accessory devices (e.g., the flexible-disc controller in the L-Series computer system). An optional battery backup card plugs into the backplane and sustains memory for up to one hour in the event of a complete power failure, thus providing an automatic restart capability.

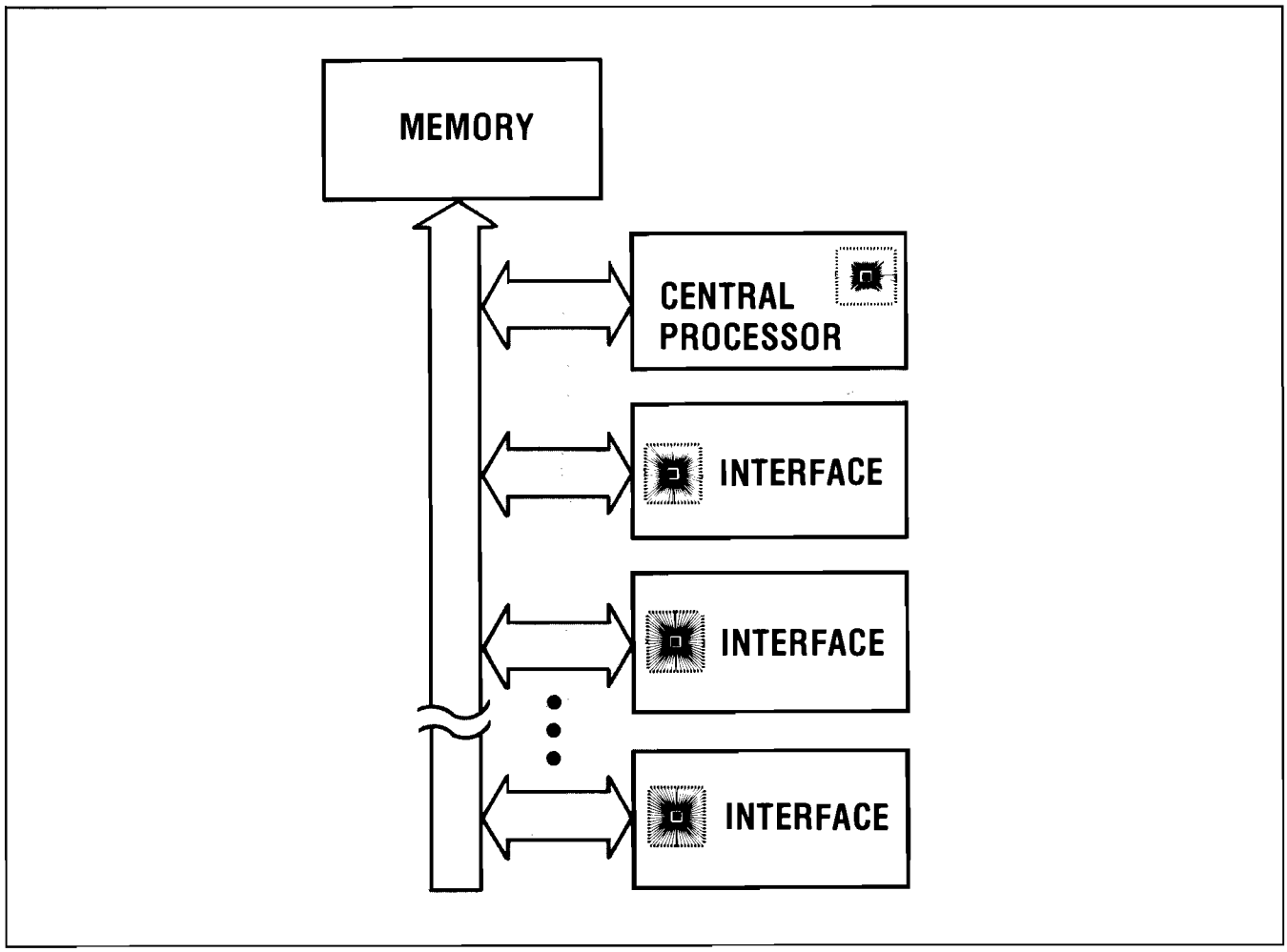


Figure 1-2. L-Series Computer Simplified Block Diagram

1-7. INPUT/OUTPUT

The input/output system for L-Series computers features an SOS chip on each I/O card, enabling each card to process its own I/O instructions and handle direct memory access (DMA) data transfers. The I/O system has a multilevel vectored priority interrupt structure with 60 distinct interrupt levels, each of which has a unique priority assignment. Any I/O device can be selectively enabled or disabled, or all I/O devices can be enabled or disabled under program control.

Data transfer between the computer and I/O devices may take place under DMA control or program control. The DMA capability provides a direct link between memory and I/O devices. The total bandwidth through all DMA channels is 2.7 million bytes (1.35 million words) per second.

The L-Series computer backplane provides the link between the processor, memory, interface cards, and the power supply. The backplane for the L-Series box computer has slots for 10 plug-in cards and the backplane for the computer system has 16 plug-in card slots. One slot must be used for the processor card and another one must be used for the memory card. Depending on the inclusion of the battery backup card, there are seven or eight slots available for I/O cards in the box computer and 11 in the computer system (where the battery, HP-IB interface, and serial interface cards are standard).

An important feature of the L-Series I/O cards is a global register which can be loaded with the select code of a specific I/O card. When the global register is enabled all I/O instructions are executed only by the specified I/O card. This not only facilitates DMA transfers but also makes reconfiguration of an I/O driver a simple matter of changing the global register to the appropriate select code. Also, since the global register can direct I/O instructions to a specific I/O card, the I/O-instruction address bits can be used to access registers on an I/O card. This feature is utilized in the design of the L-Series I/O cards to increase their capabilities.

About one-third of the area on all L-Series I/O cards is occupied by identical logic called the I/O Master, consisting of an I/O processor chip and its associated logic. The I/O Master is also available in breadboard form for users who wish to design their own I/O cards. The I/O Master is described in detail in the *HP 1000 L-Series Computer I/O Interfacing Guide,* part no. 02103-90005.

1-8. MEMORY

The L-Series computer memory is a semiconductor memory using NMOS dynamic RAM chips. Three basic memory configurations are available. In the standard L-Series, the memory controller and 64k bytes of nonexpandable random-access memory (RAM) are provided on a single printed circuit card. L-Series option 011 is an expanded memory option that supplies a single card having a mapped memory controller and 128k bytes of RAM (expandable to up to 512k bytes using 128k-byte memory array cards). Option 012 provides expanded memory by supplying a single card having a controller and 512k bytes of memory. For data integrity, memory parity checking is provided as a standard feature on all cards.

1-9. SOFTWARE

Software for the L-Series computers is based upon RTE-L, and RTE-XL, members of HP's family of Real-Time Executive (RTE) operating systems. RTE-L runs on the standard L-Series computer having 64k bytes of memory. RTE-XL takes advantage of a mapped memory system, supporting user partitions of up to 64k bytes and memory sizes from 128k bytes to 512k bytes. RTE-L and RTE-XL are real-time multiprogramming systems designed to take full advantage of the new L-Series I/O structure to improve overall CPU and I/O throughput. They offer a wide range of configurations, from a small, memory-based, execute-only system to a full disc-based system with on-line program development. The operating systems can supervise execution of multiple, priority-ordered functions; less time-critical operations can share memory through swapping. Program development in FORTRAN IV, BASIC, PASCAL, and HP Assembly Language is supported. Program development for the L-Series can also be performed on a larger HP 1000 system under RTE-IVB.

1-10. HP INTERFACE BUS

Among the I/O interface cards available for the L-Series computer is the HP 12009A HP-IB Interface Card which can interface the L-Series computer to a variety of HP peripherals and other equipment compatible with the Hewlett-Packard Interface Bus (HP-IB). (HP-IB is the Hewlett-Packard implementation of IEEE standard 488-1978, "Digital Interface for Programmable Instrumentation".) A single HP 12009A can control up to 14 HP-IB instruments and several can be used to achieve concurrent operation of multiple HP-IB instrumentation clusters under the RTE-L multiprogramming operating system.

1-11. COMPUTER NETWORK

The L-Series computer can be configured into an HP DS/1000 Distributed System by using either an HP 12007A or an HP 12044A HDLC Interface. Both of these interfaces support the high-level data link communications (HDLC) protocol, functioning as a preprocessor to handle low and medium levels of protocol processing. The L-Series computers can be easily mixed with other members of the HP 1000 family in a single distributed system.

1-12. EXPANSION AND ENHANCEMENT

Table 1-1 lists the options and accessories available to expand or enhance the L-Series computers.

Table 1-1. Options and Accessories

DESCRIPTION	PRODUCT NO.	OPTION NO.
Replaces standard 64k byte memory card with 128k byte memory card.	—	011
Replaces standard 64k byte memory card with 512k byte memory card.	—	012
230V Operation.	—	015
128k byte XL Memory Array Card.	12003A	—
Asynchronous Serial Interface.	12005A	—
Parallel Interface.	12006A	—
HDLC Interface (modem operation).	12007A	—
PROM Storage Module.	12008A	—
HP-IB Interface.	12009A	—
Intelligent Breadboard.	12010A	—
Extender Board.	12011A	—
Priority Jumper Card.	12012A	—
Battery Backup Card.	12013A	—
HDLC Interface (hard-wired operation).	12044A	—
Diagnostic Package.	24397A	—
Frontplane Connector for one 12003A.	12028A	—
Frontplane Connector for two 12003A.	12028B	—
Frontplane Connector for three 12003A.	12028C	—

1-13. SPECIFICATIONS

Table 1-2 lists the specifications for the L-Series computers. Both the box computer and the computer system have been designed to meet the safety standards of the Underwriters' Laboratories (UL) and the Canadian Standards Association (CSA). Also, the computer system has been designed to meet RFI standards of the Verband Deutscher Elektrotechniker (VDE). UL, CSA, and VDE standards are approved.

Table 1-2. Specifications

SPECIFICATIONS COMMON TO THE HP 2103L AND THE L-SERIES SYSTEMS

PROCESSOR CARD

Accumulators:	2 (A and B), 16 bits each. Implicitly addressable, also explicitly addressable as memory locations.
Memory Register:	1 (P), 16 bits.
Supplementary Registers:	2 (overflow and extend), one bit each.
Power Fail Provisions:	When primary line power falls below a predetermined level while the computer is running, a power fail warning signal from the computer power supply causes an interrupt to memory location 00004. This location is intended to contain a jump-to-subroutine (JSB) instruction to a user-written power fail subroutine. A minimum of 5 milliseconds is available to execute the power fail subroutine.
Time Base Generator Interrupt:	A time base generator interrupt is provided for maintaining a real time clock. The interrupt request is made when the CPU signals, at 10-millisecond intervals, that its internal clock is ready to roll over. Timing accuracy of the time base generator is ±4.32 seconds per (24 hour) day.

MEMORY

Memory Structure:	32 pages minimum of 2048 bytes, with direct access to current and base (page 0) pages, indirect access to 32 pages. With expanded memory, mapped access to an additional 32-224 pages.
Memory Size:	64k bytes is standard. Option 011 provides 128k bytes, expandable to 512k bytes in 128k-byte increments. Option 012 provides 512k bytes.
Memory Cycle Time:	680 nanoseconds.
Memory Parity Checking:	Parity logic on the memory cards continuously generates correct parity for all words written into memory and monitors the parity of all words read out of memory. Either odd or even parity can be selected. A parity error will generate an interrupt to memory location 00005, which can contain a JSB to a user-written parity error handling subroutine or a halt instruction.

INPUT/OUTPUT

Determination of I/O Address:	I/O address select code is set for each interface by select code switches on the interface and is therefore independent of interface card position in the card cage.

Table 1-2. Specifications (Continued)

SPECIFICATIONS COMMON TO THE HP 2103L AND THE L-SERIES SYSTEMS (Continued)

I/O Device Interrupt Priority: Depends upon I/O interface position in the card cage with respect to the processor card.

Interrupt Masking: The I/O Master logic includes an interrupt mask register which provides for selective inhibition of interrupts from specific interfaces under program control. This capability can be programmed to temporarily cut off undesirable interrupts from any combination of interfaces when they could interfere with crucial transfers.

Interrupt Latency: 0.88 to 33.1 microseconds, 1.6 microseconds typical. (Interrupts cannot be serviced until a DMA cycle or an instruction in progress has completed execution; the worst-case latency of 33.1 microseconds is based upon time to complete an integer divide, the longest instruction.)

Direct Memory Access (DMA): The I/O processor chip supports DMA capability on each I/O interface, which reduces the number of interrupts from one per data item (byte or word) to one per complete DMA block transfer, greatly reducing overhead and increasing throughput.

DMA Latency: Time interval from Service Request by an I/O device through completion of the I/O data transfer to or from the I/O interface is 0.908 microsecond for input, 1.135 microseconds for output for the interface with highest hardware priority.

Data Packing Under DMA: When byte mode is specified in control word instructions, the I/O processor chip automatically manages byte packing or unpacking.

Maximum Achievable DMA Rate: 1.35 million words (2.7 megabytes) per second.

POWER SUPPLY

Output: DC voltages, tolerances, and Periodic and Random Deviation, no load-to full load:

+ 5V	±2%	50mV, nom., 300mV, max.
+12V	±3%	100mV, max.
−12V	±6%	100mV, max.

AC Voltages and Tolerances: 39V rms ±8% split phase from three pins on backplane-mating rear connector and 27V rms ±8% single phase from two pins of front connector.

Maximum DC Output Only Current Ratings:*

+5V	+12V	−12V
25A	4.0A	2.0A

Table 1-2. Specifications (Continued)

SPECIFICATIONS COMMON TO THE HP 2103L AND THE L-SERIES SYSTEMS (Continued)

Maximum AC Power and DC Current Ratings:*†

25 kHz AC Power	DC Power Supply Current +5V	+12V	−12V
70 Watts	25A	4.0A	2.0A
140 Watts	25A	3.0A	1.5A
180 Watts	21A	2.0A	0.5A

** When operated in ambient temperature to 55°C (131°F) and at altitudes to 4.6 km (15,000 ft).*

†NOTE: Alternate ac power and dc current output combinations are possible within the 250W to 319W maximum total power output, provided that no more than the highest power or current listed above is drawn from any output. However, because of complex thermal interactions within the power supply you cannot rely upon directly trading all of the power not used in one or more dc outputs for additional ac power.

Short Circuit Protection: All dc and ac power outputs are fault protected for short circuits. The power supply will shut down, lighting the fault indicator, if any of the outputs are short circuited.

+5V Output Overvoltage Protection: The +5V output is sensed for overvoltage and the +5V supply will shut down if its output voltage exceeds 6.5V. The ac power switch must be cycled to reset the +5V output.

DC REQUIRED:

MODEL	+5V dc	+5M dc	+12V dc	−12V dc
Processor	2.962A	0.25A	0.036A	
(standby)*		0.25A		
12002A 128kb Memory	2.43A	0.63A		
(standby)*		0.42A		
12002B 512kb Memory	2.43A	1.00A		
(standby)*		0.62A		
12003A 128kb Memory Array	0.91A	0.46A		
(standby)*		0.26A		
12004A 64kb Memory	1.30A	0.757A	0.273A	0.02A
(standby)*		0.757A	0.048A	0.02A
12005A Async. Serial Interface	1.6A		0.14A	0.11A
12006A Parallel Interface				
With +5V logic	1.94A		0.18A	
With +12V logic	1.61A		0.18A	
12007A HDLC Interface (modem)	2.55A		0.35A	0.174A
12008A PROM Storage Module	2.0A		0.05A	
12009A HP-IB Interface	2.1A		0.084A	
12010A Breadboard Interface**	0.75A		0.03A	
12013A Battery Backup Card			0.17A†	0.01A
12044A HDLC Interface (direct)	2.433A		0.308A	0.035A

* Standby indicates the voltage supplied from the battery backup card during power failure.

** 12010A current requirement listed here is for the I/O Master circuitry only; logic added by the user will require additional current.

† Systems equipped with power status lights required an additional 0.4A.

Table 1-2. Specifications (Continued)

SPECIFICATIONS COMMON TO THE HP 2103L AND THE L-SERIES SYSTEMS (Continued)

ELECTRICAL SPECIFICATIONS

AC Power Required

Line Voltage:	86-127V (115V −25%/+10%) or 195-253V (230V −15%/+10%). Input line voltage is easily changed in the field by removing the HP 2103L front panel and using a screwdriver to reset the line selector switch in the power supply.
Line Frequency:	47-66 Hz.
Maximum Power Required:	500W, fused at 6A on 86-127V range.
Power Factor:	Approximately 0.65 at full load.

PHYSICAL CHARACTERISTICS

Dimensions

Height:	13.3 cm (5.25 in).
Width:	48.3 cm (19 in) panel.
Depth:	59.7 cm (23.5 in) overall; 56.4 cm (22.5 in) behind panel.

SPECIFICATIONS APPLICABLE ONLY TO THE HP 2103L

Weight:	14.1 kg (31 lb).
Ventilation:	Air intake is on the left side, exhaust is on the right.
Maximum Heat Dissipation:	258 Kilogram-Calories/hr (1024 BTU/hr).
Air Flow (Power Supply and Card Cage Fans)	
Velocity:	61 metres/min (200 ft/min) at exit with atmospheric pressure at 1016 millibars (30 inches of mercury).
Volume:	1 cubic metre/min (35.3 cfm).

ENVIRONMENTAL SPECIFICATIONS

Temperature

Operating:	0° to 55°C (32° to 131°F).
Storage:	−40° to 75°C (−40° to 167°F).
Relative Humidity:	5% to 95% at 40°C (104°F), without condensation.

Altitude

Operating:	To 4.6 km (15,000 ft).
Non-operating:	To 15.3 km (50,000 ft).

Table 1-2. Specifications (Continued)

SPECIFICATIONS APPLICABLE ONLY TO THE HP 2103L (Continued)

Vibration and Shock: HP 1000 L-Series products are type tested for normal shipping and handling shock and vibration (contact factory for review of any application that requires operation under continuous vibration).

Vibration: 0.38 mm (0.15 in) p-p deflection over 5-55 Hz frequency range, 3 axis.

Shock, 1/2 Sine, 3 Axis:

Component Weight kg (and lb)	Magnitude (g's)	Duration (millisec)
<0.454 (1)	500	1
0.454-4.54 (1-10)	100	4
>4.54 (10)	30	11

SPECIFICATIONS APPLICABLE ONLY TO THE L-SERIES SYSTEMS

ELECTRICAL SPECIFICATIONS

Standard Line Voltage and Line Frequency

Line Voltage: 90-105V (100V +5%/−10%), 108-126V (120V +5%/−10%). (Not configurable by the customer.)

Line Frequency: 57.9-62.1 Hz (60 Hz ±3.5%).

Option 015 Line Voltage and Line Frequency

Line Voltage: 198-231V (220V +5%/−10%) or 216-252V (240V +5%/−10%). (Not configurable by the customer.)

Line Frequency: 48.25-51.75 Hz (50 Hz ±3.5%).

Power Requirements: 2145B/2146B: Require at least a 15 Ampere, grounded wall-mounted power receptacle for 100/120VAC operation or at least a 7.5 Ampere, grounded wall receptacle for 220/240VAC operation (option 015). An additional power receptacle is required for the system console.

2145A/2146A: Require at least a 20 Ampere, grounded wall-mounted power receptacle for 100/120VAC operation or at least a 10 Ampere, grounded wall receptacle for 220/240VAC operation (option 015). An additional power receptacle is required for the system console.

Maximum Power Required:
Model 9 (2145A): 1500W.
Model 9 (2145B): 1200W.
Model 10 (2146A with 7910HR): 1500W.
Model 10 (2146B with 7910HR): 1200W.

Maximum Available Power for Add-on Equipment:
2145A: 500W.
2146A with 7910HR: 300W.

Table 1-2. Specifications (Continued)

SPECIFICATIONS APPLICABLE ONLY TO THE L-SERIES SYSTEMS (Continued)

PHYSICAL CHARACTERISTICS

Dimensions, Centimeters (Inches)	2145A/2146A	2145B/2146B
Height:	157.9 (62.2)	92.7 (36.5)
Width:	73.4 (29)	73.4 (29)
Depth:	81.3 (32)	81.3 (32)

Weight, Kilogram (Pound):

Model 9 (2145A): 133.6 (294).
Model 9 (2145B): 124.4 (274).
Model 10 (2146A with 7910HR): 159.5 (351).
Model 10 (2146B with 7910HR): 150.3 (331).

Racking Limitations:

The additional space in the top half of the 2145A/2146A cabinet is meant for instrumentation installed on rails and not on slides. The maximum weight for this add-on equipment is limited to 125 Kg (275 lbs) due to anti-tip safety requirements.

Maximum Heat Dissipation, Kg-Calories/hr (BTU/hr):

Model 9 (2145A/B): 567 (2252).

Model 10 (2146A/B with 7910HR): 738 (2928).

Note: The heat dissipation figures given here apply to the lower half of the 2145A/2146A cabinet; the upper half is not cooled.

ENVIRONMENTAL SPECIFICATIONS

Temperature

Operating: 10° to 40°C (50° to 104°F).

Storage: −40° to 60°C (−40° to 140°F).

Relative Humidity: 10% to 80% with maximum wet bulb temperature not to exceed 25.5°C (77.9°F), excluding all conditions which cause condensation.

Altitude

Operating: To 4.6 km (15,000 ft).

Non-operating: To 15.3 km (50,000 ft).

Vibration and Shock: HP 1000 L-Series products are type tested for normal shipping and handling shock and vibration (contact factory for review of any application that requires operation under continuous vibration).

OPERATING FEATURES

SECTION II

This section describes the bootstrap loaders, the virtual control panel (VCP) program, and the processor card registers accessible to the programmer.

2-1. HARDWARE REGISTERS

The processor card has several working registers which can be selected for display and modification via the virtual control panel program. (Interface card registers are described in section V of this manual and in the interface card reference manuals.) The functions of these processor card registers are described in the following paragraphs.

2-2. A-REGISTER

The A-register is a 16-bit accumulator that holds the results of arithmetic and logical operations performed by programmed instructions. This register can be addressed directly by any memory reference instruction as location 000000 (octal), thus permitting interrelated operations with the B-register (e.g., "add B to A," "compare B with A," etc.) using a single-word instruction.

2-3. B-REGISTER

The B-register is a second 16-bit accumulator which can hold the results of arithmetic and logical operations completely independent of the A-register. The B-register can be addressed directly by any memory reference instruction as location 000001 (octal) for interrelated operations with the A-register.

2-4. P-REGISTER

The P-register holds the address of the next instruction to be fetched from memory.

2-5. EXTEND (E) REGISTER

The one-bit extend (E) register is used by rotate instructions to link the A- and B-registers or to indicate a carry from the most-significant bit (bit 15) of the A- or B-register by an add instruction or an increment instruction. This is of significance primarily for multiple-precision arithmetic operations. If already set (logic 1), the extend bit cannot be cleared by a carry. However, the extend bit can be selectively set, cleared, complemented, or tested by programmed instructions.

2-6. OVERFLOW (O) REGISTER

The one-bit overflow (O) register is used to indicate that an add instruction, divide instruction, or an increment instruction referencing the A- or B-register has caused (or will cause) the accumulators to exceed the maximum positive or negative number that can be contained in these registers. The overflow bit can be selectively set, cleared, or tested by programmed instructions.

2-7. CENTRAL INTERRUPT REGISTER

The central interrupt register is a six-bit register that holds the select code of the last interface card whose interrupt request was serviced.

2-8. VIOLATION REGISTER

The violation register is a 16-bit register that records the address of any fetched instruction that violates memory protection rules.

2-9. INTERRUPT SYSTEM REGISTER

The interrupt system register is a one-bit register that indicates the status of the interrupt system. When set (logic 1), the interrupt system is enabled; when cleared (0), the interrupt system is disabled.

2-10. VIRTUAL REGISTERS

There are four virtual registers, M, T, U, and K that are created by the virtual control panel program and which can be accessed, via the VCP, to examine or change a program in memory or to manually create a program in memory.

2-11. M-REGISTER

The M-register holds the address of the memory cell currently being read from or written into by the virtual control panel.

2-12. T-REGISTER

The T-register indicates the contents of the memory location currently pointed to by the M-register.

2-13. U-REGISTER

The U-register indicates the contents of the mapped memory location pointed to by the M-register.

2-14. K-REGISTER

This is a one-bit register which indicates maps on or off (0 = off).

2-15. OPERATOR CONTROLS AND INDICATORS

Operator controls and indicators for the L-Series computer system are described in the system operator's manual, part no. 02145-90001.

On the HP 2103L L-Series Computer there is only one operator control: a line-power switch on the rear panel. This two-position switch controls the application of ac line power to the computer power supply and ventilating fans. If the battery backup card is installed and enabled and line power fails or is removed, the card will emit a one-second beep sound every ten seconds, indicating that the card is sustaining memory. If the card does not sustain memory for the duration of a power removal, it emits a single two-second beep when power is restored. Also, during initial installation of the card, some types of improper installation will cause the card to emit a continual beep. (A switch on the battery card allows the user to disable the card.)

2-16. SELF-TEST

The self-test consists of two test programs (Test 1 and Test 2) that automatically execute each time the computer is powered up and which provide a quick, convenient check of basic computer operation. (Also, the self-test can be executed by pressing the Test switch on the processor card.) If either self-test program fails, the computer will not operate. Successful completion of the self-test is followed immediately by execution of either a bootstrap loader, the virtual control panel program, or a program sustained in memory by a battery card, as preselected by the user.

Test 1 is built into the CPU chip on the processor card and executes immediately upon power up. It checks logic on the processor card and checks 15 lines of the address and data buses; the sixteenth data line is checked by Test 2. On successful completion all processor card LED's except the left one are lit and Test 2 is started. If Test 1 detects a failure, it stops executing and all LED's remain lit. Test 1 execution time is negligible.

Test 2 is stored in read-only memory (ROM) on the processor card and executes upon successful completion of Test 1. Test 2 checks the computer's basic instruction set, several internal registers, and the memory. If memory was sustained by the optional battery backup card, it is checked in a non-destructive manner; otherwise, Test 2 clears memory. Test 2 checks memory by reading a memory location and complementing, storing, and comparing the data; it repeats this process for each memory location. Test 2 also checks the I/O Master logic on each interface card to ensure that data transfer and flag functions are processed correctly, in addition the interrupt system and normal DMA operation of each I/O chip is checked. If Test 2 detects a failure, it stops executing and the processor card LED's indicate a failure code. The LED indication on successful completion of Test 2 depends on the computer action selected by the Start-Up switches on the processor board. Test 2 has a maximum execution time of 18 seconds for 512 kilobytes.

2-17. BOOTSTRAP LOADERS

Bootstrap loading of a program for the L-Series computer is provided for by several loaders contained in ROMs on the processor card. These are the same ROMs that contain the self-test 2 and the virtual control panel (VCP) program. The loading devices include a disc drive (via HP-IB), PROM card, DS/1000 network Link and HP 264x mini-cartridge tape. There are two ways to invoke a loader: auto-boot when power comes up; and by VCP command. Auto-boot can only invoke three of the loaders: disc, PROM card, and DS/1000; the VCP can invoke any of the loaders by a command from the operator. The VCP load commands are discussed later in this section.

2-18. LOADER SELECTION FOR AUTO-BOOT

The selection of an auto-boot is by means of switches located on the processor card. These switches, the Start-Up switches, are set during installation and also provide options other than auto-boot selection. When a loader has been selected for auto-boot and the self-test completes, the boot loader executes if memory was lost; or the program in memory executes if memory was sustained. Refer to table 2-1 for Start-Up switch settings. (Also, see the start-up flowchart in the appendix.)

2-19. PROGRAM STARTS

When an auto-boot completes without error, the loaded program starts execution at memory location 02, except on power fail the auto-restart program starts execution at location 04. The loader sets the contents of the A- and B-registers as follows:

a. Cold start (memory not sustained):
   1. A = loader command parameters.
   2. B = zero.

b. Auto-restart (memory sustained):
   1. A = zero.
   2. B = zero.

c. %E command from VCP:

1. A = −1.
2. B = zero.

d. %B command from VCP:

1. A = loader command parameters.
2. B = pointer to ASCII string. If the first word is 0, no is entered.

Table 2-1. Start-Up Switch Settings

SWITCHES* U1- S1	S2	S3	S4	COMPUTER ACTION
C	C	C	C	Loop on self-test regardless of error.
C	C	C	O	Do not use. (Reserved.)
C	C	O	C	Loop on self-test and stop on error.
C	C	O	O	Do not use. (Reserved.)
C	O	C	C	Do not use. (Reserved.)
C	O	O	C	Run VCP** routine on completion of Test-1 (CPU) section of self-test. (Skip Test 2.)
O	C	C	C	If memory lost (not sustained), run VCP routine; otherwise, restart program (JMP 4B). (Note 2)
O	C	O	C	If memory lost, load from PROM card; otherwise, restart program (JMP 4B). (Note 2) (In order to auto-boot from PROM, the card must have select code 22.)
O	O	C	C	If memory lost, load via HDLC card; otherwise, restart program (JMP 4B). (Note 2) (In order to auto-boot via HDLC, the card must have select code 24.)
O	O	O	C	If memory lost, load from first file of disc (via HP-IB); otherwise, restart program (JMP 4B). (Note 2) (In order to auto-boot via HP-IB, the HP-IB interface card must have select code 27 and the disc drive must have HP-IB address 2.)

*O = open (up); C = closed (down).
**Virtual control panel.

Notes: 1. When a loader finishes an auto-boot, it starts execution of the loaded program at location 02.

2. If the auto-restart feature is disabled (switch U1S8 closed), the program cannot restart and the boot loader (or VCP routine) will execute.

3. Do not use any switch combination that is not shown above.

2-20. ROM RE-ENTRY AND SEQUENTIAL EXECUTION

The processor card ROM can be re-entered from a program either to execute the self-test or to boot load and execute the next sequential program from the loading device. There are four ways to re-enter the ROM code, as follows:

a. Restart the self-test and assume memory sustained, which will result in restarting the program in memory if the Start-Up switches are set for it. (Refer to the programming sequence following the next item.)

b. Restart the self-test at the memory section, clear or retain memory, and continue to the loader section. The programming sequence is as follows:

```
CLA          Self-test 1
OTA 1
CLC 2        Enable ROM
JMP 2/3      Self-test 1 or Memory (Self-test 2)
```

c. Re-enter to boot load the next sequential program using the loaders. The programming sequence is as follows:

```
CCA          Re-enter loader
OTA 1
CLC 2        Enable ROM
JSB 1        Go to loader
```

Note that without VCP only the loader initially used for auto-boot can be re-entered for sequential program loading. With VCP you can return to the loader that started the sequence, with specified parameters.

d. With the disc loader, re-enter to boot load the specific program described by the "ABS" code in the following call back programming sequence.

```
LDA 400B      Set correct entry page
OTA 1           for continuation load
CLA,CLE,INA   Indicate disc call back - do not
                suspend
CLC 2         Enable ROM
JSB 1         Go to driver & save parameter
                address
ABS...        HP-IB bus address
ABS...        Device unit number (head for 7906)
ABS...        Absolute starting sector
ABS...        Cylinder offset
```

This sequence assumes that the global register is set prior to entry to the loader and the absolute starting sector is the combined cylinder/head/sector for that drive. When the load is completed, the loader will start execution in the standard JMP 2 manner. If a suspend after load was specified by the E-register being set when called, then the program will build a halt (102077) followed by a JMP 2 in its temporary storage area and then jump to the halt. In the case of the halt the operator can enter either a %E or a %R to continue. Any error will return to the VCP, if present, or start the original load over.

The 7906 will be accessed in the surface mode only, all other discs will be accessed in the cylinder mode.

2-21. DEVICE PARAMETERS AND MEDIA FORMATS

There is a specific data format for each combination of loader, interface card, loading device, and media. The data formats are described in figure 2-1.

MINI-CARTRIDGE TAPE

Device: HP 264x Terminal

Interface: HP 12005A Asynchronous Serial Interface

Default Parameters*: 000020

Format: Reads absolute binary file, writes 4k absolute binary block.

Loader: Transmits special escape sequence to invoke a read of a record and does checksum of the data. When writing to tape, a block number is used to specify which 4k-word memory area is to be dumped to tape (0 = 0 to 4k).

If a file number is specified then the program will issue a find file command; if not, the tape is read from where it stands. When writing to the tape, the program will not write a file mark; this allows sequential blocks to be written in a series. There are only two units (0 and 1) on the terminal; if a larger unit number is specified, the result will be unpredictable.

PROM MODULE

Device: PROM (2k x 8 bits)

Interface: HP 12008A PROM Storage Module

Default Parameters*: 000022

Format: Count-Address-Data
Count = number of data words (see Table 2-4 for file format).
Address = starting location in memory where data is to be stored.
Data = 16-bit words, one word per location until count is satisfied.

Loader: Uses STC-LIA process to transfer data. The PROM cannot be written to nor does it use the block number or unit number.

*See Figure 2-2 for loader command formats.

DISC DRIVE

Device: HP 7902, 7906, or 7910 Disc Drive

Interface: HP 12009A HP-IB Interface

Default Parameters*: 002027

Format: Transfers memory image of 32k (or 16k) locations (see Table 2-4 for file formats).

Loader: Uses HP-IB protocol to communicate with the disc. The load sequence is:

1. Device clear
2. Status check
3. Read/write 32k (or 16k) words via DMA
4. Status check

Figure 2-1. Loading Device Parameters and Media Formats (Part 1 of 2)

Note: For L each file starts on a track boundary.
For XL each file is 256 blocks.

COMPUTER NETWORK

Device:	HP 1000 Computer.
Interface:	HP 12007A/12044A HDLC Interface.
Default Parameters*:	000024
Format:	Reads absolute binary or memory image files, writes a 32k memory image file.
Loader:	Standard handshake using HP distributed system protocol. Block number and unit number are not used.

*See Figure 2-2 for loader command formats.

Figure 2-1. Loading Device Parameters and Media Formats (Part 2 of 2)

2-22. VIRTUAL CONTROL PANEL

The virtual control panel (VCP) program is an interactive program that enables an external device (such as a terminal) to control the CPU in a manner similar to a conventional computer control panel. That is, it allows the operator to load programs using the loaders, access the various registers (A, B, P, etc. plus I/O card registers), examine or change memory, and control execution of a program. The VCP program is stored in ROM on the processor card and is for use with an HP 12005A Interface Card having select code 20 or DS/1000 with a select code 24. Only one interface card in the computer can serve as a VCP interface; the card selection is established when the system is installed. When the L-Series computer is operating as a node in a computer network, the VCP can be an adjacent computer in the network.

2-23. VCP PROGRAM OPERATION

The VCP program is executed from ROM as a software program and uses the various machine registers (A, B, etc.) during its execution. Therefore, these registers are automatically saved upon entry to the VCP code. Thus a response to an inquiry is the data that was saved at the time of entry to the program. The exceptions to this are indicated by an asterisk in table 2-2. When the operator enters the Run (%R) or Single Cycle (%S) command, the VCP program restores the machine with the current data in the save area and starts execution as specified by the program execution address.

The VCP program can be entered in three ways as follows:

a. After a power-up, ROM execution is directed to the VCP program instead of a boot load routine;

b. When the VCP interface card requests a slave cycle to enable the VCP program (e.g., BREAK key pressed on VCP) ; or

c. When a HLT (halt) instruction is executed.

After a power-up the program address is set to the last available memory address calculated during the self-test and for the L the A-register is set to the number I/O chips that were tested, for the XL the A-register is set to the highest page of memory tested during the self-test plus the number of I/O cards. This enables the operator to verify that all installed memory and I/O cards were tested. (Also, the B-register contains the revision code of the processor card ROM.) When entered, the VCP displays the basic set of registers (P, A, B, M, and T) and waits for an operator response. The operator can enter any of the characters or commands listed in tables 2-2 and 2-3 and the VCP program will respond as indicated in the tables. Multiple inquiries can be made on a line; a carriage return is entered when a new line is necessary.

After a response to an inquiry the operator can change the data contained in that register or memory location by entering new data; for example:

Table 2-2. VCP Characters and Associated Registers

CHARACTER ENTERED	RESPONSE**	MEANING
A	xxxxxx	A-register contents
B	xxxxxx	B-register contents
C*	0000xx	Central interrupt register contents
E	x	E-register contents
G	x000xx	Global register (GR) contents and status (Bit 15 = 1 = GR enabled; = 0 = GR disabled)
I	x	Interrupt system status (0 = off, 1 = on)
O	x	O-register contents
M	0xxxxx	Memory address (pointer for T)
P	0xxxxx	Program execution address
T	xxxxxx	Memory contents of location pointed to by M
U	xxxxxx	Memory contents of mapped location pointed to by M; XL only
V*	0xxxxx	Violation register contents
K	x	Indicate maps on or off (0 =off); XL only
The following characters apply only to the interface card whose select code is in the global register; if the GR is 0, the response is "!?". Note that reading an interface card register is equivalent to executing an LIA/B instruction addressed to that register; e.g., RS is equivalent to LIA/B 32.		
R0-R3*	xxxxxx	Interface card registers 20 through 23 (DMA)
RD*	xxxxxx	Interface card register 30 (data)
RC*	xxxxxx	Interface card register 31 (control)
RS*	xxxxxx	Interface card register 32 (status)
RM*	xxxxxx	Interrupt mask register contents
RI	xxxxxx	Displays the ID obtained by diagnose mode 1
RX	xxxxx	Displays the internal flag data obtained by diagnose mode 2
RF	xxxxx	Displays the flag status
*Registers not contained in the save area. **x = octal data.		

A 001234 4321<cr>
operator inputs
<cr> is carriage return

A 004321

NOTE: When operating in DS the response will be DSVCP:A, etc.

Table 2-3. VCP Commands

COMMAND	MEANING
%E	Execute. Start execution of program at location 02 (A-register equals −1 (all ones) and B-register equals 0).
%R	Run. Set all registers to the appropriate values in the save area and start execution at address specified by the P-register.
%S	Single cycle. Set all registers to the appropriate values in the save area and execute one instruction pointed at by P.
%P	Preset. Generate a control reset (CRS) signal to reset all interface cards.
%C	Clear memory. Set all memory to zero and perform preset.
N	Next. Increment memory pointer and display the contents of the M- or T-register, whichever was selected (either M or T must be selected by the operator).
D	Decrement. Same as N except decrement the pointer.
%L	Load. Execute a specified loader routine. See figure 2-2 for format.
%B	Load and go (boot). Same as %L except start program execution at completion of load. See figure 2-2 for format.
%W	Write. Write to the selected device. See figure 2-4 for format. For XL, set P- and B-registers before executing this command.
%Mnn	Set maps sequentially starting with nn; XL only.
%T	Execute the pretest and return to VCP (memory is sustained but the I/O system is reset)

Data input is terminated by the operator entering a carriage return. If during an input the program cannot interpret a character, the program will output the characters "!?" and then start a new line. During any data input the operator can abort the input by entering a rub-out (DEL). The loader commands, %B, %L, and %W can also be aborted by a rub-out. When entering data into a register, leading zeros may be omitted.

2-24. LOADER COMMANDS

The loader commands can be entered via the VCP in either of two ways:

a. Allow the parameter default values (given in figure 2-1) to be used; or

b. Specify all necessary parameters.

The loader command format is shown in figure 2-2.

LOADER COMMAND FORMAT

%B/L/W *dv ffbusc text*

where:

dv = device type as follows:

DC = disc (cartridge or flexible) via HP-IB
CT = cartridge tape (HP 264x)
RM = PROM card
DS = DS computer network Link

ff = file number (octal 0 to 17 only)

b = 4k-word memory block number when writing to cartridge tape; HP-IB bus address of disc drive; otherwise, use 0.

u = unit number (0 to 7) only if used on device. For the HP 7906 Disc Drive, the unit number is the head number.

sc = select code of interface card to be used

text = file name, or ASCII string to be passed to the program after it is loaded. This is only available with the %B command.

Note: See figure 2-1 for default parameters for each loading device.

Note that spaces cannot be used in the command entry. The following formats are all acceptable:

%B*dvtext*	Device parameters are defaulted; text cannot start with a number.
%B*dvffbusc*	No text passed.
%B*dvffbusctext*	Text passed.

EXAMPLES:

%BDC	Load and start execution of the default program on disc. (Disc parameters defaulted to 002027; see figure 2-1).
%BDC2	Load and start execution of the third program on the disc and default other parameters.
%LDC27025	Load (but don't execute) and override parameter default values: file number 2 HP-IB bus address 7 unit 0 select code 25
%WDC27025	Same as above except write to file 2.

Figure 2-2. Loader Command Format

The VCP loader command error codes and their meanings are listed in table 2-4. Error codes 000010 and up are loader device related errors. To use the table for error codes ten or greater, first find the loader you are using (e.g., disc) then look up the desired error code and its meaning.

Table 2-4. VCP Loader Command Errors

ERROR CODES	MEANING
000001	Device type incorrect.
000002	Select code less than 20, or no card with the select code you specified with the loader command.
	DISC RELATED ERRORS
000010	Time out initializing bus (HP-IB; check bus connection and device address).
000011	Time out reading disc type (HP-IB; check bus connection).
000012	Time out UDC (Universal Device Code) or reading status (HP-IB; check bus connection).
000013	Status error (HP-IB; check bus connection).
000014	Time out during file mask command (HP-IB; check bus connection).
000015	Time out during seek command (HP-IB; check bus connection).
000016	Time out during read command (HP-IB; check bus connection).
000017	Time out during read data (HP-IB; check bus connection).
000020	Parity error during write (HP-IB; check bus connection).
000021	Time out during flush of FIFO (HP-IB; check interface).
000022	Time out during DSJ (Device Specified Jump) command (HP-IB; check bus connection).
000023	Bad DSJ return (check disc status, e.g., door open, etc).
000040	Loading RTE-L software on L-Series having expanded memory and battery backup. This is not allowed.
000060	Disc not identifiable.

Table 2-4. VCP Loader Command Errors (Continued)

ERROR CODES	MEANING
	MINI-CARTRIDGE TAPE RELATED ERRORS
000010	File forward error (mini-cartridge error).
000011	Checksum error (mini-cartridge error).
000012	End-of-File mark only (empty file; no data before EOF).
000020	Write error to mini-cartridge (%W command).
	PROM CARD RELATED ERRORS
000010	Beyond address range or no more cards (file is larger than the number of PROM locations available).
000012	EOD on present card (End-of-Data, indicates that no more PROM locations are available); or Select code rollover (select code rolled over 77 octal on load, which is illegal); or Load of more than 32K (block = 1 to 8) not on card boundary.
000013	DMA timed out on a transfer (data transfer hangup).
000040	Loading RTE-L software on L-Series having expanded memory and battery backup. This is not allowed.
	DS/1000 (HDLC) RELATED ERRORS
000010	Time out after CLC 0 (timed out during transfer; check DS link).
000011	Checksum error (check PROGL in central).
000012	Time out after down load request (check DS link).
000013	Time out after file number (check DS link).
000014	Bad transfer (central generated; check DS link or PROGL in central).
000015	Time out after buffer request (check DS link).
000016	Time out after count echo (check DS link).
000017	Time out waiting for data (check DS link).
000020	VCP mode request time out (%W command).
000021	Central will not accept data (%W command).

2-25. PROM AND DISC LOADER FILE FORMATS

The file formats for the prom and disc loaders are shown in table 2-5. The table shows that the first two words of a file are used to identify the type of program being loaded, and also where the save area for the VCP code will reside. The battery backup columns indicate whether battery backup may be installed or not during the boot-up procedure for disc or PROM. NO indicate that the battery backup should not be installed and will result in error message (ER 000040) if it is installed. OK indicates that does not matter, the boot-up will proceed with or without battery backup installed.

Table 2-5. File Formats

A-REG* WORD 1	B-REG WORD 2	MEMORY TYPE	DISC† BATTERY BACKUP	DISC† VCP** AREA	PROM †† BATTERY BACKUP	PROM †† VCP†† AREA
-word count	address <4	XL L	NO OK	last page last page	NO OK	last page last page
-word count	address >3	XL L	NO OK	last page last page	OK OK	last page last page
+word count>8	address <4	XL L	NO OK	last page last page	NO OK	last page last page
+word count>8	address >3	XL L	NO OK	last page last page	OK OK	base page last page
block count	partial	XL L	OK OK	base page last page	OK OK	base page last page

* Block represents a 64k byte area, it requires 8 full blocks to fill a 512k byte system. The partial represents a portion of a block. When using blocks and partial starting location zero is assumed so that block and partial resolve into the A- and B-registers. The word count represents the data area only and does not include the A- and B-registers (location 0 and 1). The cassette (CT) and the distributed system (DS) network loaders save and restore locations 0 and 1 as the A- and B-registers. These loaders do not act on the content, as do the disc and PROM loaders, the VCP save area is the same as described in the following note.

** The VCP save area will be the last 64 locations on the last page on a standard (L) memory system. The VCP save area for an expanded (XL) memory system will be the last 64 locations of the base page. When loading on an XL the first two words are used to determine if the save area should be moved.

† Disc always transfers a full 64k bytes, even when using a block and partial the partial results in a full transfer. When using a standard memory (L) the A- and B-registers have no meaning and may be random when written to disc, and will remain unchanged when reading. The transfer always starts at location zero in physical memory. If a block count greater than zero is loaded on a standard L memory system the load will be successful but the execution of the ! program will fail.

†† When loading a standard (L) memory from PROM and block count and partial area used the block count must be zero, otherwise an ER 000040 will occur. When loading an expanded (XL) memory and block is not zero then the program must start with PROM address zero.

2-26. VCP USER CONSIDERATIONS

When using the VCP to debug a program the user should be aware of the following conditions:

a. The VCP program uses an interface card and modifies the characteristics of that card. When the program exits it sets Register 24 on the interface card to all ones to allow software detection of a VCP interaction and, thus, reinitialization of an operation. (This also causes an interrupt if the interrupt system is enabled.)

b. The status of the interrupt system (STC 4 (on) or CLC 4 (off)) is not indicated and will remain unchanged unless %P is executed to preset the computer.

c. Memory protect is the same as b, above.

d. Single Cycle Operation. If an indirect execution is greater than three levels, the instruction will not execute and the P-register will remain unchanged.

2-27. VCP SLAVE FUNCTIONS

The slave feature of an I/O processor chip is used in conjunction with the VCP program. The slave feature enable is read into the I/O chip of the VCP interface card on power-up and cannot be altered until the next power-up condition. After power-up a change in the state of the slave signal causes the I/O chip to generate a slave request on the next instruction fetch. When the request is granted, the I/O chip requests the CPU's current P-register contents and saves these contents in a register in the I/O chip. The I/O chip then stores the starting address of the VCP program into the CPU's P-register, instructs the CPU to enable the boot ROM, and allows execution to start. The VCP program can be started in several other ways, as follows:

a. On power-up and after the self-test the VCP program starts execution if it is selected in lieu of a boot loader. This selection may often be used because the loaders can be invoked individually from the VCP.

b. When a HLT* (halt) instruction is executed the I/O processor chip interprets it in the same manner as a change in the slave enable signal. This allows a program to have breakpoints for debugging purposes.

c. After a CLC 3 instruction is executed the I/O chip will allow three instruction executions before generating a slave request. This instruction is used by the VCP program to facilitate single instruction executions, allowing the user to step through a program.

During execution of the VCP program, access to the P-save register in the I/O chip is accomplished with LIA/B 3 and OTA/B 3* (without the instruction's Flag bit set). It should also be noted that the I/O chip will not execute a slave request until an STC 2 (disable boot ROM) instruction has been executed. This prevents re-entry of the VCP program once it has been entered.

The starting address of the VCP* program is set during the self-test with an OTA/B 3,C instruction with the A- or B-register set to the address. This address can also be read back with an LIA/B 3,C instruction.

---

* If break is not enabled on any card, then the instruction is ignored.

PROGRAMMING INFORMATION

SECTION III

This section describes the software data formats and machine-language instruction coding required to operate the computer and its associated input/output system.

3-1. DATA FORMATS

As shown in figure 3-1, the basic data format is a 16-bit word in which bit positions are numbered from 0 through 15 in order of increasing significance. Bit position 15 of the data format is used for the sign bit; a logic 0 in this position indicates a positive number and a logic 1 in this position indicates a negative number. The data is assumed to be a whole number and the binary point is therefore assumed to be to the right of the number.

The basic word can also be divided into two 8-bit bytes or combined to form a 32-bit double word. The byte format is used for character-oriented input/output devices; packing two bytes of data into one 16-bit word is accomplished by software drivers. In I/O operations, the higher-order byte (byte 1) is the first to be transferred.

The integer double-word format is used for extended arithmetic in conjunction with the extended arithmetic instructions described under paragraphs 3-19 and 3-20. Bit position 15 of the most-significant word is the sign bit and the binary point is assumed to be to the right of the least-significant word. The integer value is expressed by the remaining 31 bits. When addressing a double word in memory, the address refers to the least-significant word location; the next higher memory address contains the most-significant word. When loaded into the accumulators, the B-register contains the most-significant word and the A-register contains the least-significant word.

The floating-point double-word format is used with floating-point software. Bit position 15 of the most-significant word is the mantissa sign and bit position 0 of the least-significant word is the exponent sign. Bits 1 through 7 of the least-significant word express the exponent and the remaining bits (bits 8 through 15 of the least-significant word and bits 0 through 14 of the most-significant word) express the mantissa. Since the mantissa is assumed to be a fractional value, the binary point appears to the left of the mantissa. Software drivers convert decimal numbers to this binary form and normalize the quantity expressed (sign and leading mantissa differ). If either the mantissa or the exponent is negative, that part is stored in two's complement form.

The number must be in the approximate range of 10^{-38} to 10^{+38}. When loaded into the accumulators, the A-register contains the most-significant word and the B-register contains the least-significant word.

Figure 3-1 also illustrates the octal notation for both single-length (16-bit) and double-length (32-bit) words. Each group of three bits, beginning at the right, is combined to form an octal digit. A single-length (16-bit) word can therefore be fully expressed by six octal digits and a double-length (32-bit) word can be fully expressed by 11 octal digits. Octal notation is not shown for byte or floating-point formats, since bytes normally represent characters and floating-point numbers are given in decimal form.

The range of representable number for single-word data is +32,767 to −32,768 (decimal) or +77,777 to −100,000 (octal). The range of representable numbers for double-word integer data is +2,147,483,647 to −2,147,483,648 (decimal) or +17,777,777,777 to −20,000,000,000 (octal).

3-2. MEMORY ADDRESSING

3-3. PAGING

The computer memory is logically divided into pages of 1,024 words each. A page is defined as the largest block of memory that can be directly addressed by the address bits of a single-length memory reference instruction. (Refer to paragraph 3-8.) These memory reference instructions use 10 bits (bits 0 through 9) to specify a memory address; thus, the page size is 1,024 locations (2000 octal). Octal addresses for each page, up to a maximum memory size of 32K, are listed in table 3-1.

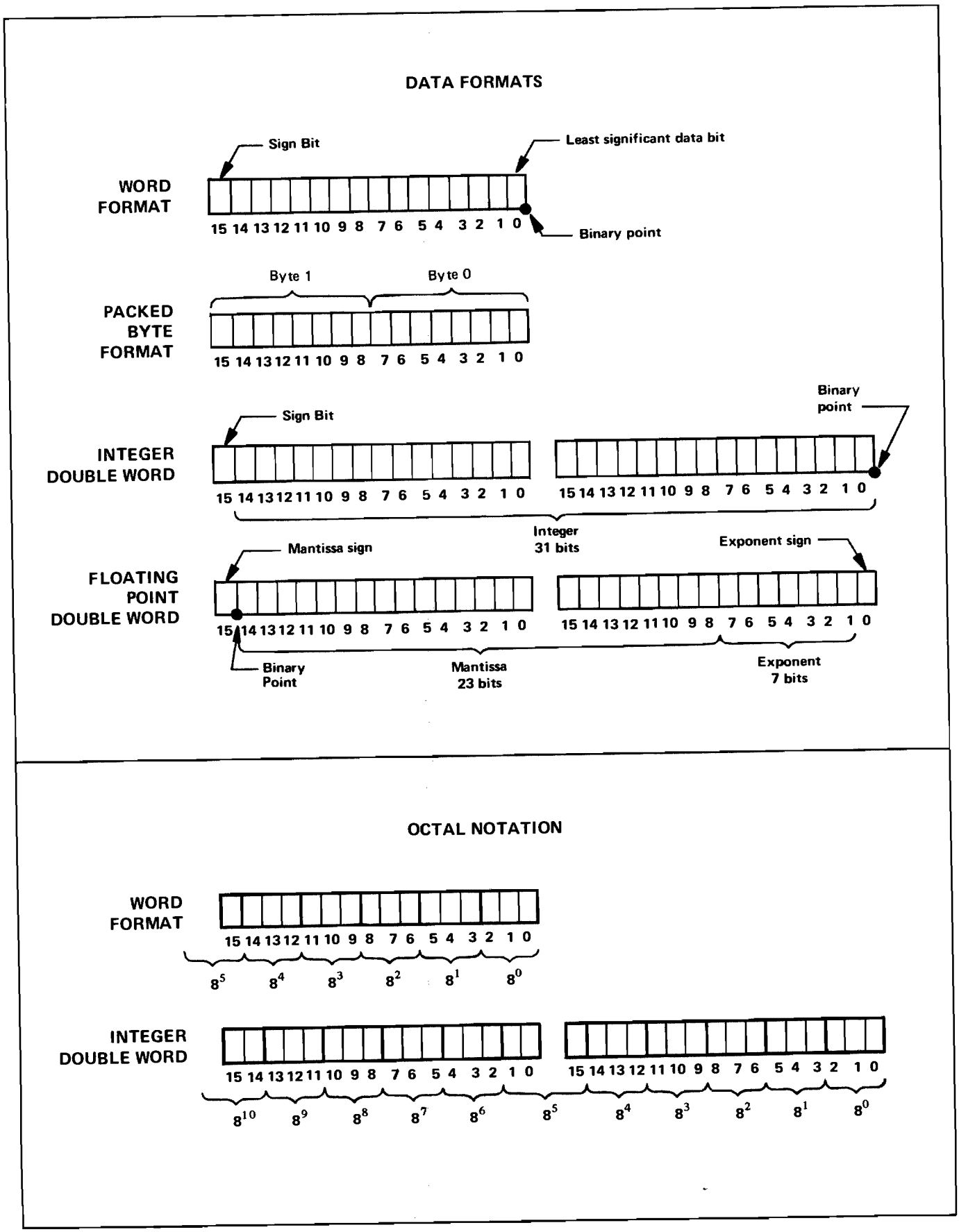
Provision is made to directly address one of two pages: page zero (the base page consisting of locations 00000 through 01777) and the current page (the page in which the instruction itself is located). Memory reference instructions reserve bit 10 to specify one or the other of these two pages. To address locations on any other page, indirect addressing is used as described in following paragraphs. Page references are specified by bit 10 as follows:

a. Logic 0 = Page Zero (Z).

b. Logic 1 = Current Page (C).

3-4. DIRECT AND INDIRECT ADDRESSING

All memory reference instructions reserve bit 15 to specify either direct or indirect addressing. For single-length memory reference instructions, bit 15 of the instruction word is used; for extended arithmetic memory reference instructions, bit 15 of the address word is used. Indirect addressing uses the address part of the instruction to access another word in memory, which is taken as the new



2270-2

Figure 3-1. Data Formats and Octal Notation

Table 3-1. Memory Paging

MEMORY SIZE (BYTES)	PAGE	OCTAL ADDRESSES
	0	00000 to 01777
	1	02000 to 03777
	•	
	•	
32Kb ↓	15	36000 to 37777
	16	40000 to 41777
	•	
	•	
	•	
64Kb	31	76000 to 77777

memory reference for the same instruction. This new address word is a full 16 bits long: 15 address bits plus another direct/indirect bit. The 15-bit length of the address permits access to any location in memory. If bit 15 again specifies indirect addressing, still another address is obtained; thus, multistep indirect addressing may be done to any number of levels. The first address obtained that does not specify another indirect level becomes the effective address for the instruction. Direct or indirect addressing is specified by bit 15 as follows:

a. Logic 0 = Direct (D).

b. Logic 1 = Indirect (I).

3-5. MAPPING

In the standard L-Series computer, the 15-bit address bus can address all of main memory. With option 011 or 012, the L-Series computer has up to 512k bytes of main memory and mapping is used to address the additional memory. With these options, main memory locations 100 through 137 are reserved for the mapping registers and locations 140 through 177 are reserved for the DMA relocation registers. The 32 map locations each correspond to a 1k area in logical memory. The physical memory which responds to the logical address is determined by the contents of the corresponding map register location. Location 0001XY points to the 1k page of physical memory that will respond to logical addresses in page XY. Note that the lower 10 address bits are always unaltered by the mapping.

3-6. RESERVED MEMORY LOCATIONS

The first 64 memory locations of the base page (octal addresses 00000 through 00077) are reserved as listed in table 3-2. The first two locations are reserved as addresses for the two 16-bit accumulators (the A- and B-registers). Locations 00004 through 00077 are reserved for priority interrupts; as long as locations 00020 through 00077 do not have actual priority interrupt assignments, as determined by the input/output devices included in the system configuration, these locations can be used for programming purposes.

In the standard L-Series computer, the uppermost 64 locations of memory are reserved for use by the loaders, self-test, and virtual control panel program; these locations should not be used for programming purposes. With expanded memory (option 001 or 012), 64 words of memory are reserved in physical page 0 (base page) for use by the loaders, self-test, and virtual control panel program. In addition, with option 011 or 012 certain memory locations are reserved for mapping registers as previously explained.

3-7. NONEXISTENT MEMORY

Nonexistent memory is defined as those locations not physically implemented in the machine. Any attempt to write into a nonexistent memory location will be ignored (no operation). Any attempt to read from a nonexistent memory location will return an all-ones word (177777 octal); no parity error occurs.

3-8. BASE SET INSTRUCTION FORMATS

The base set of instructions are classified according to format. The five formats used are illustrated in figure 3-2 and described in following paragraphs. In all cases where a single bit is used to select one of two cases (e.g., D/I), the choice is made by coding a logic 0 or logic 1, respectively.

Table 3-2. Reserved Memory Locations

MEMORY LOCATION	PURPOSE
00000	A-register address.
00001	B-register address.
00002-00003	Reserved.
00004	Power-fail interrupt.
00005	Memory parity interrupt.
00006	Time base generator interrupt.
00007	Memory protect interrupt.
00010	Unimplemented instruction interrupt.
00011-00017	Reserved.
00020-00077	Interrupt locations corresponding to interface card select codes.
01700-01777	Expanded memory systems reserve the last 64 locations of the base page for use by the loaders, self-tests, and virtual control panel program.
77700-77777	Standard memory systems reserve the last 64 locations of memory for use by the loaders, self-tests, and virtual control panel program.

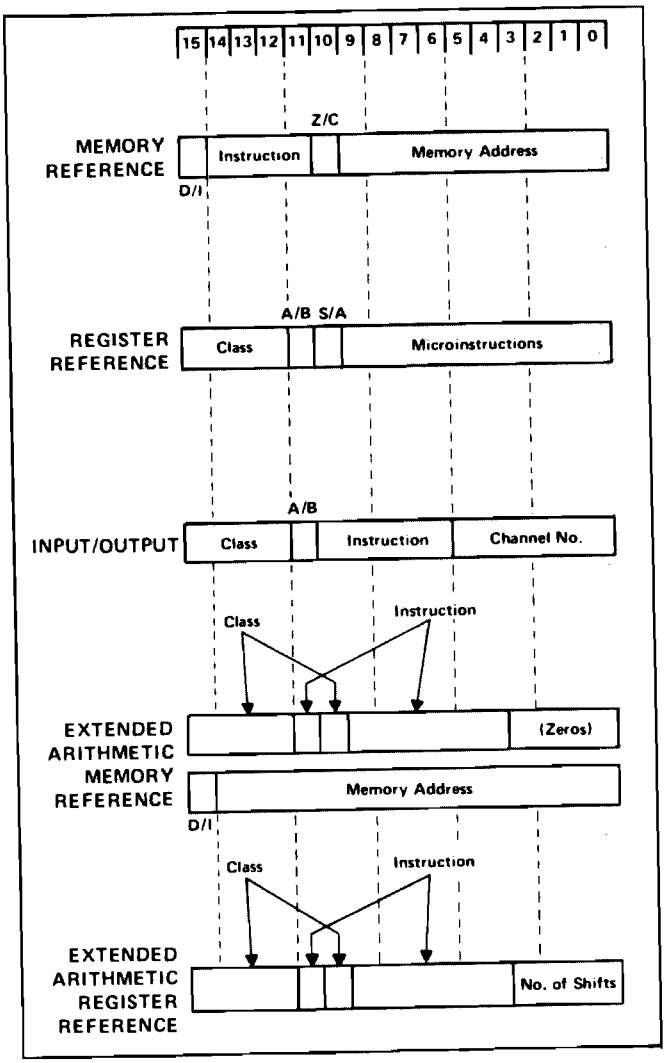


Figure 3-2. Base Set Instruction Formats

3-9. MEMORY REFERENCE INSTRUCTIONS

This class of instructions, which combines an instruction code and a memory address into one 16-bit word, is used to execute some function involving data in a specific memory location. Examples are storing, retrieving, and combining memory data to and from the accumulators (A- and B-registers) or causing the program to jump to a specified location in memory.

The memory cell referenced (i.e., the absolute address) is determined by a combination of 10 memory address bits (0 through 9) in the instruction word and 5 bits (10 through 14) assumed from the current contents of the PAGE-register. This means that memory reference instructions can directly address any word in the current page; additionally, if the instruction is given in some location other than the base page (page zero), bit 10 (Z/C) of the instruction doubles the addressing range to 2,048 locations by allowing the selection of either page zero or the current page. (This causes bits 10 through 14 of the address contained in the P-register to be set to zero instead of assuming the current contents of the PAGE-register.) This feature provides a convenient linkage between all pages of memory, since page zero can be reached directly from any other page.

As discussed under paragraph 3-4, bit 15 is used to specify direct or indirect memory addressing. Note also that since the A- and B-registers are addressable, any single-word memory reference instruction can apply to either of these registers as well as to memory cells. For example, an ADA 0001 instruction adds the contents of the B-register (address 0001) to the contents currently held in the A-register; specify page zero for these operations since the addresses of the A- and B-registers are on page zero.

3-10. REGISTER REFERENCE INSTRUCTIONS

In general, the register reference instructions manipulate bits in the A-register, B-register, and E-register; there is no reference to memory. This group includes 39 basic microinstructions which may be combined to form a one-word multiple instruction that can operate in various ways on the contents of the A-, B-, and E-registers. These 39 instructions are divided into two subgroups: the shift/rotate group (SRG) and the alter/skip group (ASG). The appropriate subgroup is specified by bit 10 (S/A). Typical operations are clear and/or complement a register, conditional skips, and register increment.

3-11. INPUT/OUTPUT INSTRUCTIONS

The input/output instructions use bits 6 through 11 for a variety of I/O instructions and bits 0 through 5 to apply the instructions either to a specific I/O channel (if the global register is disabled) or to an I/O card register. This provides the means of controlling all peripherals connected to the I/O channels and for transferring data to and from these peripherals. Included also in this group are instructions to control the interrupt system, overflow bit, and computer halt.

3-12. EXTENDED ARITHMETIC MEMORY REFERENCE INSTRUCTIONS

As the single-word memory reference instruction described previously, the extended arithmetic memory reference instructions include an instruction code and a memory address. In this case, however, two words are required. The first word specifies the extended arithmetic class (bits 12 through 15 and 10) and the instruction code (bits 4 through 9 and 11); bits 0 through 3 are not needed and are coded with zeros. The second word specifies the memory address of the operand. Since the full 15 bits are used for the address, this type of instruction may directly address any location in memory. As with all memory reference instructions, bit 15 is used to specify direct or

indirect addressing. Operations performed by this class of instructions are integer multiply and divide (using double-length product and dividend) and double load and double store.

3-13. EXTENDED ARITHMETIC REGISTER REFERENCE INSTRUCTIONS

This class of instructions provides long shifts and rotates on the combined contents of the A- and B-registers. Bits 12 through 15 and 10 identify the instruction class; bits 4 through 9 and 11 specify the direction and type of shift; and bits 0 through 3 control the number of shifts, which can range from 1 to 16 places.

3-14. BASE SET INSTRUCTION CODING

Machine language coding for the base set of instructions are provided in following paragraphs. Definitions for these instructions are grouped according to the instruction type: memory reference, register reference, input/output, extended arithmetic memory reference, and extended arithmetic register reference.

Directly above each definition is a diagram showing the machine language coding for that instruction. The gray shaded bits code the instruction type and the gold shaded bits code the specific instruction. Unshaded bits are further defined in the introduction to each instruction type. The mnemonic code and instruction name are included above each diagram.

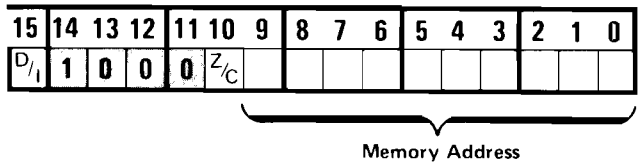
In all cases where an additional bit is used to specify a secondary function (D/I, Z/C, or H/C), the choice is made by coding a logic 0 or logic 1, respectively. In other words, a logic 0 codes D (direct addressing), Z (zero page), or H (hold flag); a logic 1 codes I (indirect addressing), C (current page), or C (clear flag).

3-15. MEMORY REFERENCE INSTRUCTIONS

The following 14 memory reference instructions execute a function involving data in memory. Bits 0 through 9 specify the affected memory location on a given memory page or, if indirect addressing is specified, the next address to be referenced. Indirect addressing may be continued to any number of levels; when bit 15 (D/I) is a logic 0 (specifying direct addressing), that location will be taken as the effective address. The A- and B-registers may be addressed as locations 00000 and 00001 (octal), respectively.

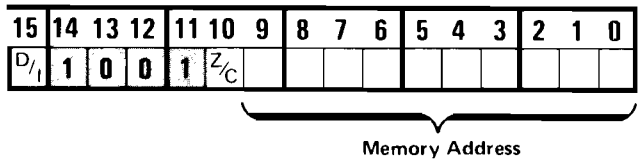
If bit 10 (Z/C) is a logic 0, the memory address is on page zero; if bit 10 is a logic 1, the memory address is on the current page. If the A- or B-register is addressed, bit 10 must be a logic 0 to specify page zero, unless the current page is page zero.

ADA ADD TO A



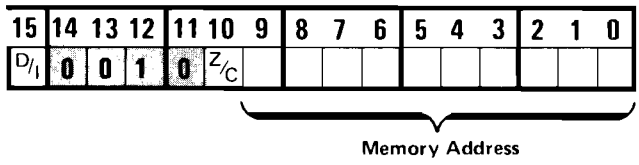
Adds the contents of the addressed memory location to the contents of the A-register. The sum remains in the A-register and the contents of the memory cell are unaltered. The result of this addition may set the extend bit or the overflow bit. (Extend and overflow examples are illustrated on page A-9.)

ADB ADD TO B



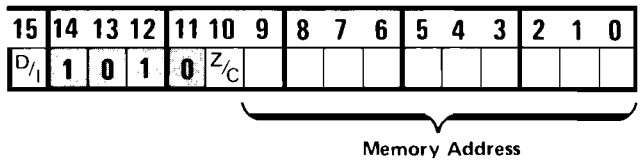
Adds the contents of the addressed memory location to the contents of the B-register. The sum remains in the B-register and the contents of the memory cell are unaltered. The result of this addition may set the extend bit or the overflow bit. (Extend and overflow examples are illustrated on page A-9.)

AND "AND" TO A



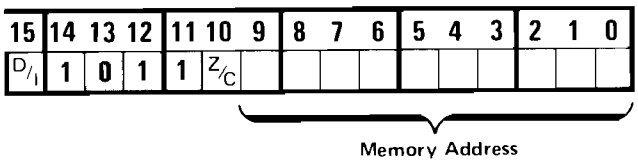
Combines the contents of the addressed memory location and the contents of the A-register by performing a logical "and" operation. The contents of the memory cell are unaltered.

CPA COMPARE TO A



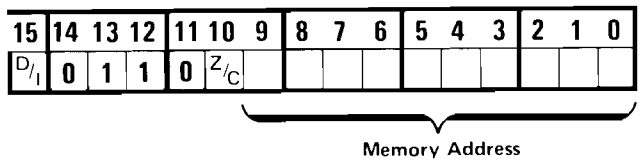
Compares the contents of the addressed memory location with the contents of the A-register. If the two 16-bit words are not identical, the next instruction is skipped; i.e., the P-register advances two counts instead of one count. If the two words are identical, the next sequential instruction is executed. Neither the A-register contents nor memory cell contents are altered.

CPB COMPARE TO B



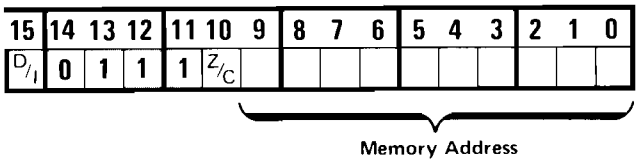
Compares the contents of the addressed memory location with the contents of the B-register. If the two 16-bit words are not identical, the next instruction is skipped; i.e., the P-register advances two counts instead of one count. If the two words are identical, the next sequential instruction is executed. Neither the B-register contents nor memory cell contents are altered.

IOR "INCLUSIVE OR" TO A



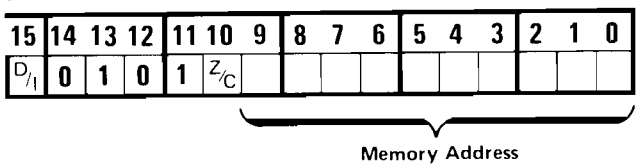
Combines the contents of the addressed memory location and the contents of the A-register by performing a logical "inclusive or" operation. The contents of the memory cell are unaltered.

ISZ INCREMENT AND SKIP IF ZERO



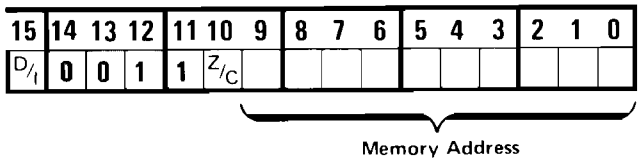
Adds one to the contents of the addressed memory location. If the result of this operation is zero (memory contents incremented from 177777 to 000000), the next instruction is skipped; i.e., the P-register is advanced two counts instead of one count. If the result of this operation is not zero, the next sequential instruction is executed. In either case, the incremented value is written back into the memory cell. An ISZ instruction referencing locations 0000 or 0001 (A- or B-register) cannot set the extend bit or the overflow bit.

JMP JUMP



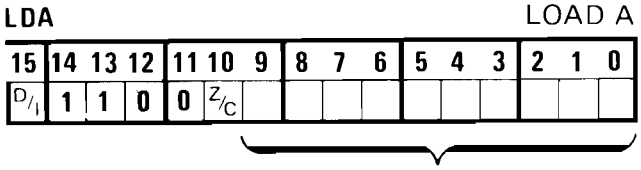
Transfers control to the addressed memory location. That is, a JMP causes the P-register count to set according to the memory address portion of the JMP instruction so that the next instruction will be read from that location.

JSB JUMP TO SUBROUTINE



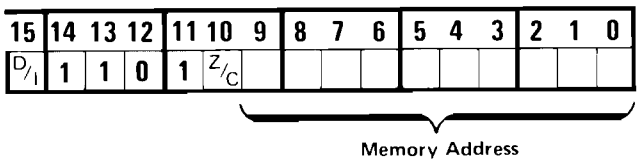
This instruction, executed in location P (P-register count), causes the computer control to jump unconditionally to the memory location (m) specified by the memory address portion of the JSB instruction. The contents of the P-register plus one (return address) is stored in memory location m, and the next instruction to be executed will be that contained in the next sequential memory location (m +1). A return to the main program sequence at P + 1 will be affected by a JMP indirect through location m.

LDA LOAD A



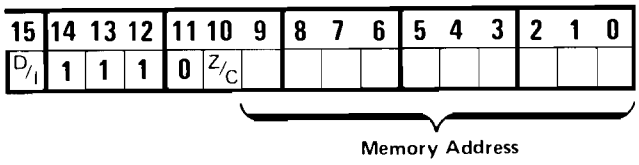
Loads the contents of the addressed memory location into the A-register. The contents of the memory cell are unaltered.

LDB LOAD B



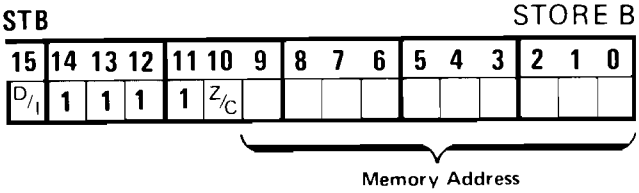
Loads the contents of the addressed memory location into the B-register. The contents of the memory cell are unaltered.

STA STORE A



Stores the contents of the A-register in the addressed memory location. The previous contents of the memory cell are lost; the A-register contents are unaltered.

STB STORE B



Stores the contents of the B-register in the addressed memory location. The previous contents of the memory cell are lost; the B-register contents are unaltered.

XOR "EXCLUSIVE OR" TO A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D/I	0	1	0	0	Z/C										

Memory Address

Combines the contents of the addressed memory location and the contents of the A-register by performing a logical "exclusive or" operation. The contents of the memory cell are unaltered.

3-16. REGISTER REFERENCE INSTRUCTIONS

The 39 register reference instructions execute functions on data contained in the A-register, B-register, and E-register. These instructions are divided into two groups: the shift/rotate group (SRG) and the alter/skip group (ASG). In each group, several instructions may be combined into one word and are thus individually termed "microinstructions." Since the two groups perform separate and distinct functions, microinstructions from the two groups cannot be mixed. Unshaded bits in the coding diagrams are available for combining other microinstructions.

3-17. SHIFT/ROTATE GROUP. The 20 instructions in the shift/rotate group (SRG) are defined first; this group is specified by setting bit 10 to a logic 0. A comparison of the various shift/rotate functions are illustrated in figure 3-3. Rules for combining microinstructions in this group are as follows (refer to table 3-3):

a. Only one microinstruction can be chosen from each of the two multiple-choice columns.

b. References can be made to either the A-register or B-register, but not both.

c. Sequence of execution is from left to right.

d. In machine code, use zeros to exclude unwanted microinstructions.

e. Code a logic 1 in bit position 9 to enable shifts or rotates in the first position; code a logic 1 in bit position 4 to enable shifts or rotates in the second position.

f. The extend bit is not affected unless specifically stated. However, if a "rotate-with-E" instruction (ELA, ELB, ERA, or ERB) is coded but disabled by a logic 0 in bit position 9 and/or position 4, the E-register will be updated even though the A- or B-register contents are not affected; to avoid this situation, code a "no operation" (three zeros) in the first and/or second positions.

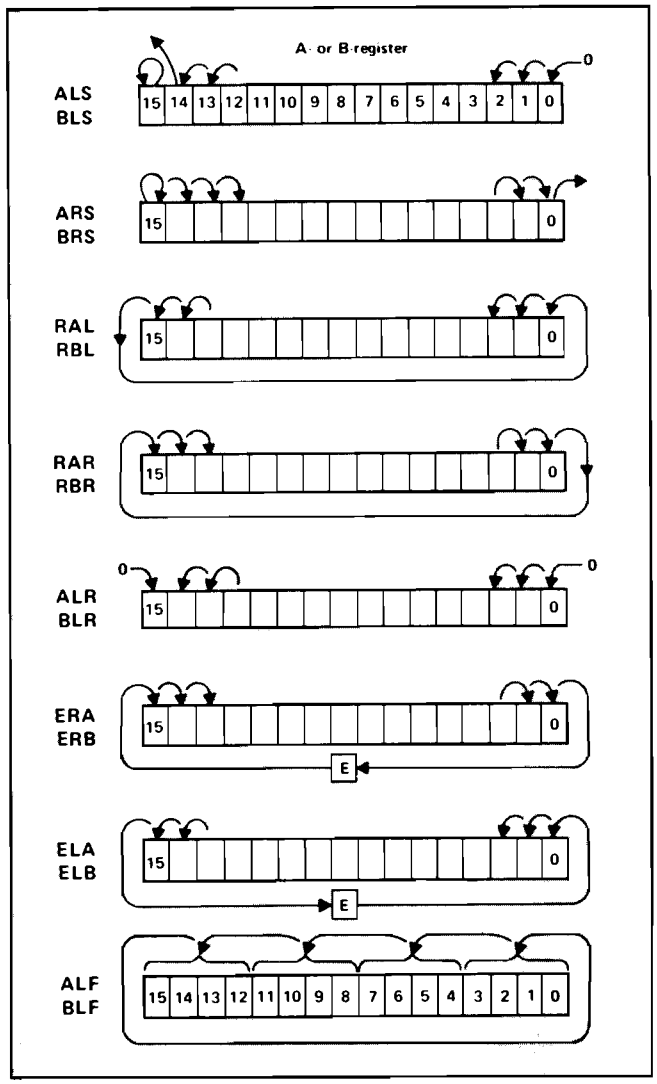
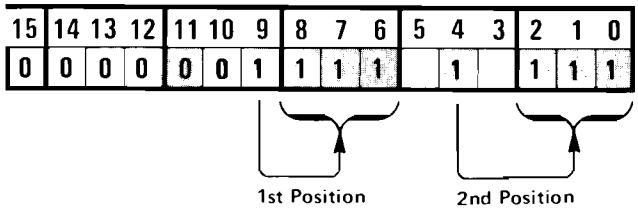


Figure 3-3. Shift and Rotate Functions

Table 3-3. Shift/Rotate Group Combining Guide

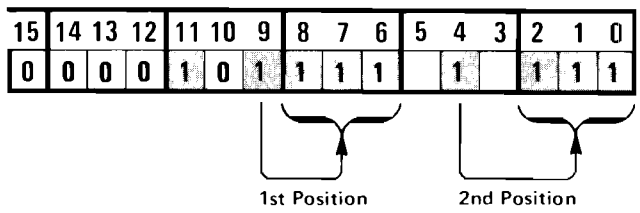
[ALS ARS RAL RAR ALR ALF ERA ELA]	[,CLE]	[,SLA]	[ALS ARS RAL RAR ALR ALF ERA ELA]
[BLS BRS RBL RBR BLR BLF ERB ELB]	[,CLE]	[,SLB]	[BLS BRS RBL RBR BLR BLF ERB ELB]

ALF ROTATE A LEFT FOUR



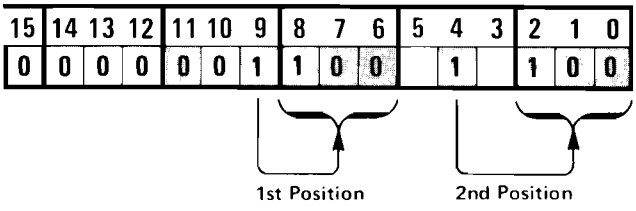
Rotates the A-register contents (all 16 bits) left four places. Bits 15, 14, 13, and 12 rotate around to bit positions 3, 2, 1, and 0, respectively. Equivalent to four successive RAL instructions.

BLF ROTATE B LEFT FOUR



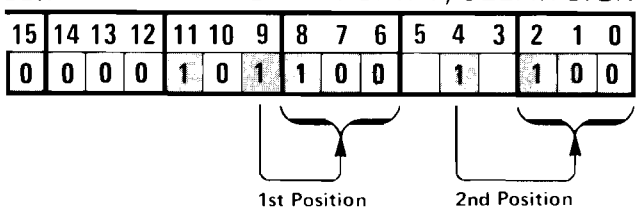
Rotates the B-register contents (all 16 bits) left four places. Bits 15, 14, 13, and 12 rotate around to bit positions 3, 2, 1, and 0, respectively. Equivalent to four successive RBL instructions.

ALR A LEFT SHIFT, CLEAR SIGN



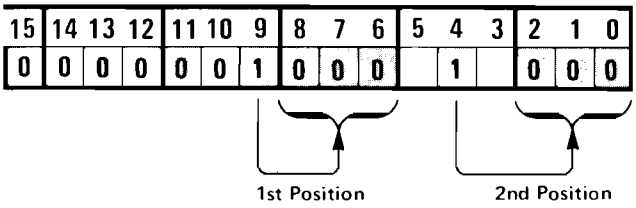
Shifts the A-register contents left one place and clears sign bit 15.

BLR B LEFT SHIFT, CLEAR SIGN



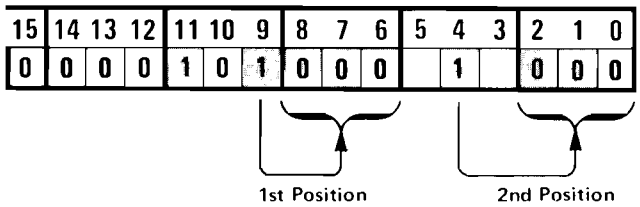
Shifts the B-register contents left one place and clears sign bit 15.

ALS A LEFT SHIFT



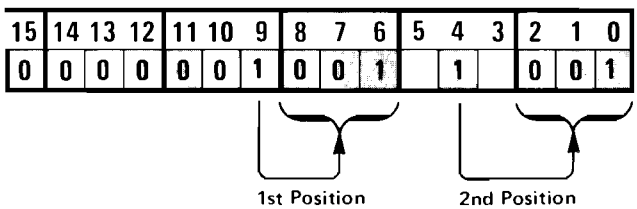
Arithmetically shifts the A-register contents left one place, 15 magnitude bits only; bit 15 (sign) is not affected. The bit shifted out of bit position 14 is lost; a logic 0 replaces vacated bit position 0.

BLS B LEFT SHIFT



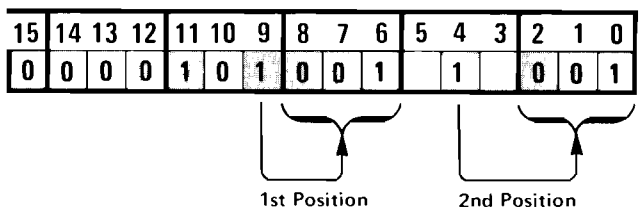
Arithmetically shifts the B-register contents left one place, 15 magnitude bits only; bit 15 (sign) is not affected. The bit shifted out of bit position 14 is lost; a logic 0 replaces vacated bit position 0.

ARS A RIGHT SHIFT



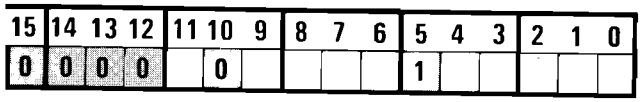
Arithmetically shifts the A-register contents right one place, 15 magnitude bits only; bit 15 (sign) is not affected. A copy of the sign bit is shifted into bit position 14; the bit shifted out of bit position 0 is lost.

BRS B RIGHT SHIFT



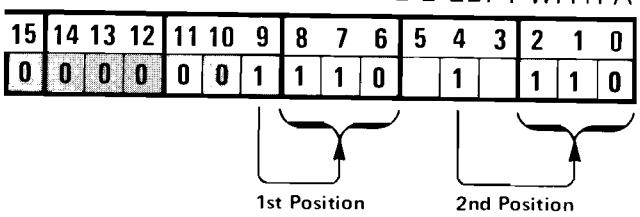
Arithmetically shifts the B-register contents right one place, 15 magnitude bits only; bit 15 (sign) is not affected. A copy of the sign bit is shifted into bit position 14; the bit shifted out of bit position 0 is lost.

CLE CLEAR E



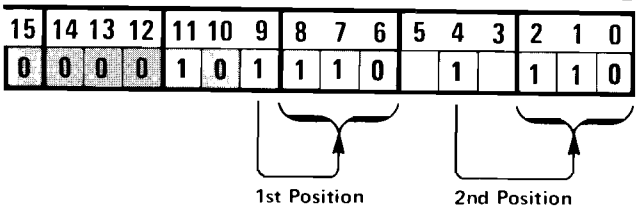
Clears the E-register; i.e., the extend bit becomes a logic 0.

ELA ROTATE E LEFT WITH A



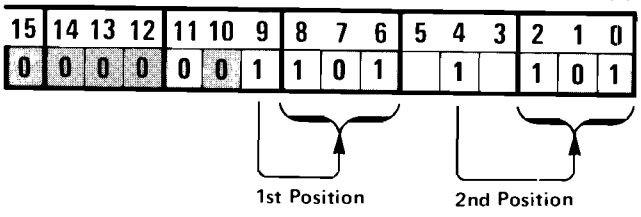
Rotates the E-register content left with the A-register contents (one place). The E-register content rotates into bit position 0; bit 15 rotates into the E-register.

ELB ROTATE E LEFT WITH B



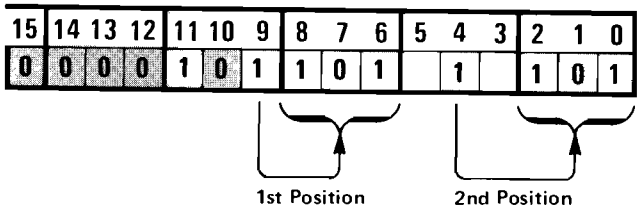
Rotates the E-register content left with the B-register contents (one place). The E-register content rotates into bit position 0; bit 15 rotates into the E-register.

ERA ROTATE E RIGHT WITH A



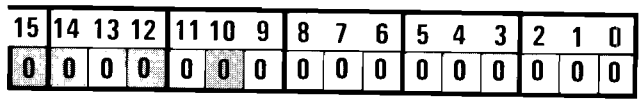
Rotates the E-register content right with the A-register contents (one place). The E-register content rotates into bit position 15; bit 0 rotates into the E-register.

ERB ROTATE E RIGHT WITH B



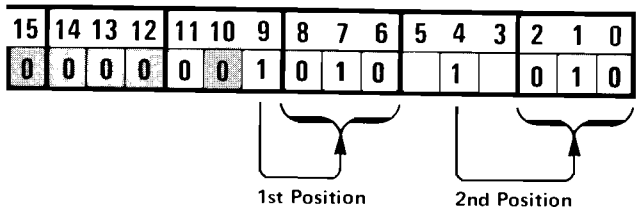
Rotates the E-register content right with the B-register contents (one place). The E-register content rotates into bit position 15; bit 0 rotates into the E-register.

NOP NO OPERATION



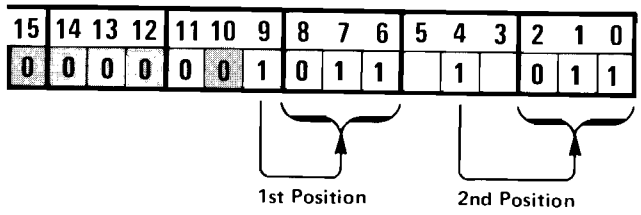
This all-zeros instruction causes a no-operation cycle.

RAL ROTATE A LEFT



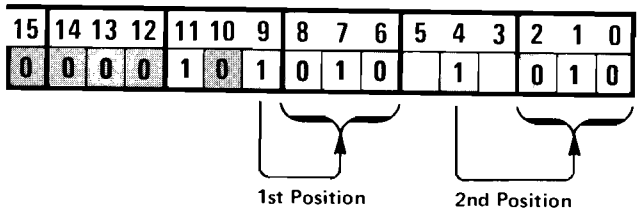
Rotates the A-register contents left one place (all 16 bits). Bit 15 rotates into bit position 0.

RAR ROTATE A RIGHT



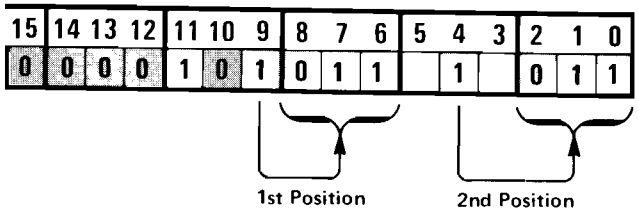
Rotates the A-register contents right one place (all 16 bits). Bit 0 rotates into bit position 15.

RBL ROTATE B LEFT



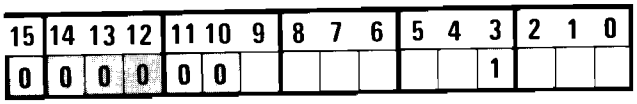
Rotates the B-register contents left one place (all 16 bits). Bit 15 rotates into bit position 0.

RBR ROTATE B RIGHT



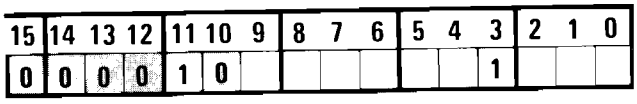
Rotates the B-register contents right one place (all 16 bits). Bit 0 rotates into bit position 15.

SLA SKIP IF LSB OF A IS ZERO



Skips the next instruction if the least-significant bit (bit 0) of the A-register is a logic 0.

SLB SKIP IF LSB OF B IS ZERO



Skips the next instruction if the least-significant bit (bit 0) of the B-register is a logic 0.

3-18. ALTER/SKIP GROUP. The 19 instructions comprising the alter-skip group (ASG) are defined next. This group is specified by setting bit 10 to a logic 1. Rules for combining microinstructions are as follows (refer to table 3-4):

a. Only one microinstruction can be chosen from each of the two multiple-choice columns.

b. References can be made to either the A-register or B-register, but not both.

c. Sequence of execution is from left to right.

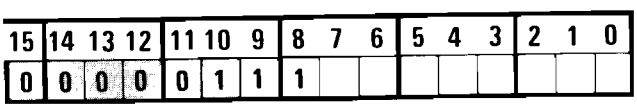
d. If two or more skip functions are combined, the skip function will occur if either or both conditions are met. One exception exists: refer to the RSS instruction.

e. In machine code, use zeros to exclude unwanted microinstructions.

Table 3-4. Alter/Skip Group Combining Guide

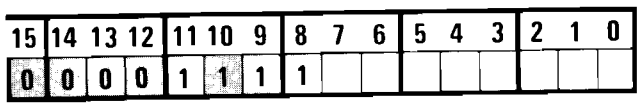
[CLA / CMA / CCA]	[,SEZ]	[CLE / CME / CCE]	[,SSA] [,SLA] [,INA] [,SZA] [,RSS]
[CLB / CMB / CCB]	[,SEZ]	[CLE / CME / CCE]	[,SSB] [,SLB] [,INB] [,SZB] [,RSS]

CCA CLEAR AND COMPLEMENT A



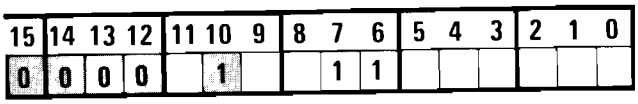
Clears and complements the A-register contents; i.e., the contents of the A-register become 177777 (octal). This is the two's complement form of −1.

CCB CLEAR AND COMPLEMENT B



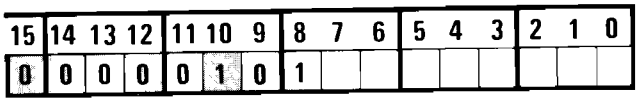
Clears and complements the B-register contents; i.e., the contents of the B-register become 177777 (octal). This is the two's complement form of −1.

CCE CLEAR AND COMPLEMENT E



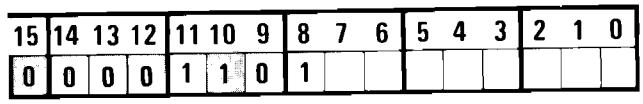
Clears and complements the E-register content (extend bit); i.e., the extend bit becomes a logic 1.

CLA CLEAR A



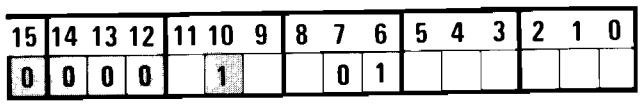
Clears the A-register; i.e., the contents of the A-register become 000000 (octal).

CLB CLEAR B



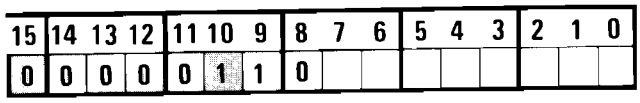
Clears the B-register; i.e., the contents of the B-register become 000000 (octal).

CLE CLEAR E



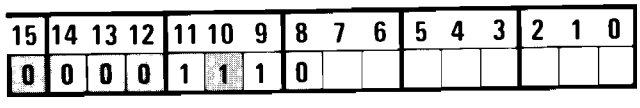
Clears the E-register; i.e., the extend bit becomes a logic 0.

CMA COMPLEMENT A



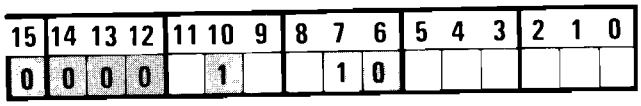
Complements the A-register contents (one's complement).

CMB COMPLEMENT B



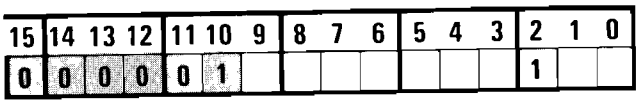
Complements the B-register contents (one's complement).

CME COMPLEMENT E



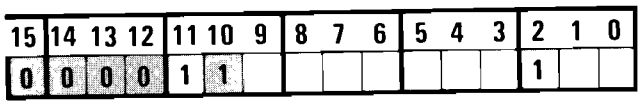
Complements the E-register content (extend bit).

INA INCREMENT A



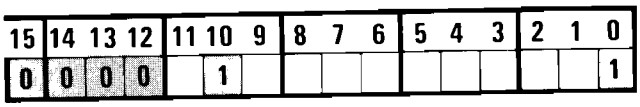
Increments the A-register by one. Can result in setting the extend bit or the overflow bit.

INB INCREMENT B



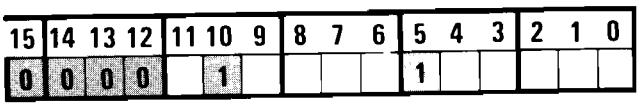
Increments the B-register by one. Can result in setting the extend bit or the overflow bit.

RSS REVERSE SKIP SENSE



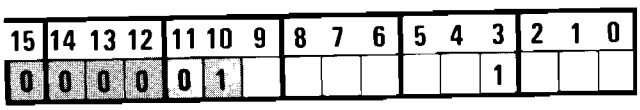
Skip occurs for any of the following skip instructions, if present, when the non-zero condition is met. An RSS without a skip instruction in the word causes an unconditional skip. If a word with RSS also includes both SSA and SLA (or SSB and SLB), bits 15 and 0 must both be logic 1's for a skip to occur; in all other cases, a skip occurs if one or more skip conditions are met.

SEZ SKIP IF E IS ZERO



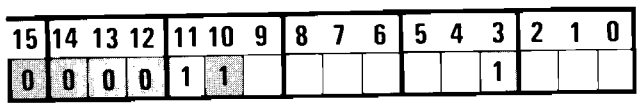
Skips the next instruction if the E-register content (extend bit) is a logic 0.

SLA SKIP IF LSB OF A IS ZERO



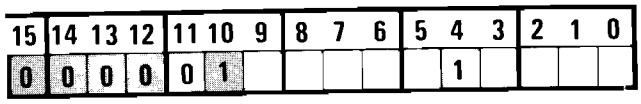
Skips the next instruction if the least-significant bit (bit 0) of the A-register is a logic 0; i.e., skips if an even number is in the A-register.

SLB SKIP IF LSB OF B IS ZERO



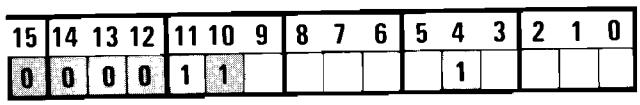
Skips the next instruction if the least-significant bit (bit 0) of the B-register is a logic 0; i.e., skips if an even number is in the B-register.

SSA SKIP IF SIGN OF A IS ZERO



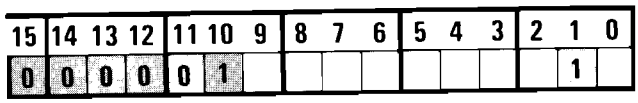
Skips the next instruction if the sign bit (bit 15) of the A-register is a logic 0; i.e., skips if a positive number is in the A-register.

SSB SKIP IF SIGN OF B IS ZERO



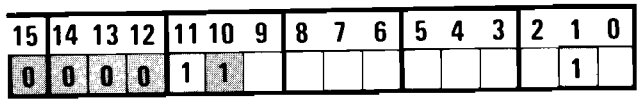
Skips the next instruction if the sign bit (bit 15) of the B-register is a logic 0; i.e., skips if a positive number is in the B-register.

SZA SKIP IF A IS ZERO



Skips the next instruction if the A-register contents are zero (16 zeros).

SZB SKIP IF B IS ZERO



Skips the next instruction if the B-register contents are zero (16 zeros).

3-19. INPUT/OUTPUT INSTRUCTIONS

The following input/output instructions provide the capability of setting, clearing or testing the flag and control bits associated with DMA, programmed I/O, interrupts, memory protect, time base generator, parity error, global register, and overflow. These I/O instructions permit data transfer between the A- and B-registers and either specific I/O devices or between registers associated with memory

protect, parity error, or interrupts. The various registers and I/O devices are addressed by means of their register numbers and select codes.

Bit 11, where relevant, specifies the A- or B-register or distinguishes between set control and clear control; otherwise, bit 11 may be a logic 0 or a logic 1 without affecting the instruction (although the assembler will assign zeros in this case). In those instructions where bit position 9 includes the letters H/C, the programmer has the choice of holding (logic 0) or clearing (logic 1) the device flag after executing the instruction. (Exception: the H/C bit associated with instructions SOC and SOS holds or clears the overflow bit instead of the device flag.)

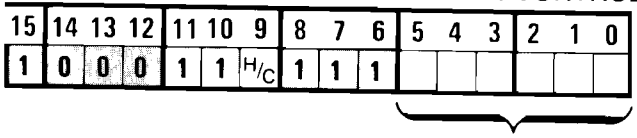
Bits 8, 7, and 6, specify the appropriate I/O instruction. When the global register is disabled, bits 5 through 0 form a two-digit octal select code (address) to apply the instruction to one of up to 64 input/output devices or functions. When the global register is enabled, bits 5 through 0 apply the instruction to a register on the I/O card whose select code is in the global register. (The global register is discussed further in paragraph 5-4.)

NOTE

Execution of I/O instructions is inhibited when the memory protect feature is enabled. Refer to paragraph 4-3.

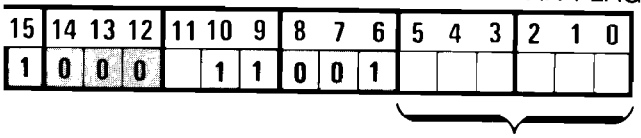
The following instruction descriptions assume that the global register is disabled and, therefore, the instructions are addressed to a select code.

CLC CLEAR CONTROL



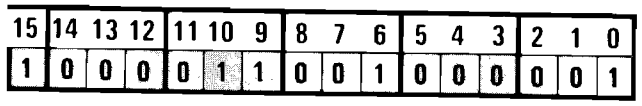
Clears the control bit (Control 30) of the selected I/O channel or function. This turns off the specific device channel and prevents it from interrupting. A CLC 00 instruction clears the control bits from select code 20 upward, effectively turning off all I/O devices.

CLF CLEAR FLAG



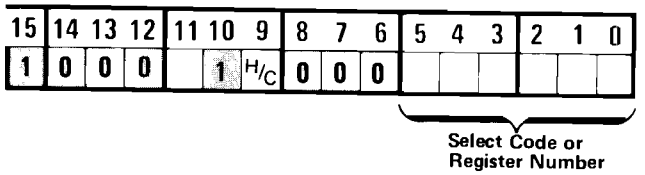
Clears the flag (Flag 30) of the selected I/O channel or function. A CLF 00 instruction disables the interrupt system for the time base generator and all interface cards; this does not affect the status of the individual channel flags.

CLO CLEAR OVERFLOW



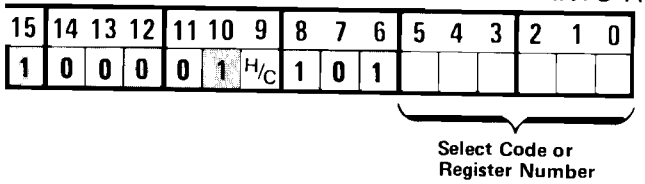
Clears the overflow bit.

HLT HALT



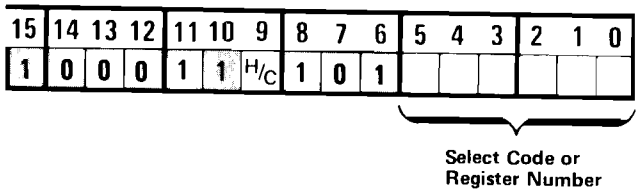
Halts the computer, holds or clears the flag of the selected I/O channel, and invokes the virtual control panel program. The HLT instruction will be contained in the T-register, which is displayed on the VCP when the VCP program starts executing. The P-register (also displayed) will normally contain the HLT location plus one.

LIA LOAD INTO A



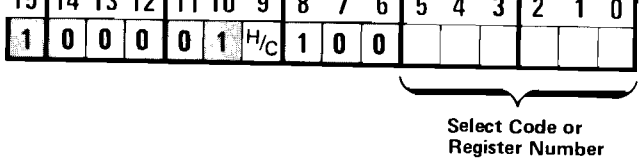
Loads the contents of the addressed I/O buffer or special function register into the A-register.

LIB LOAD INTO B



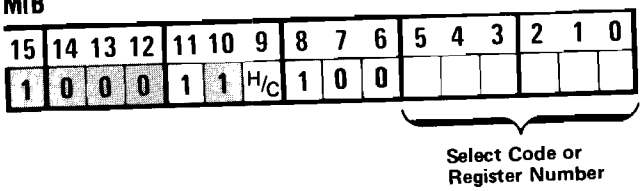
Loads the contents of the addressed I/O buffer or special function register into the B-register.

MIA MERGE INTO A



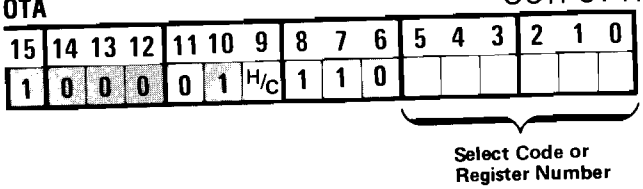
By executing a logical "inclusive or" function, merges the contents of the addressed I/O buffer or special function register into the A-register.

MIB MERGE INTO B



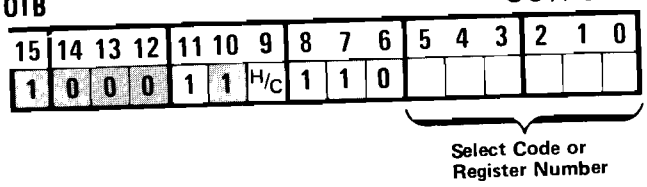
By executing a logical "inclusive or" function, merges the contents of the addressed I/O buffer or special function register into the B-register.

OTA OUTPUT A



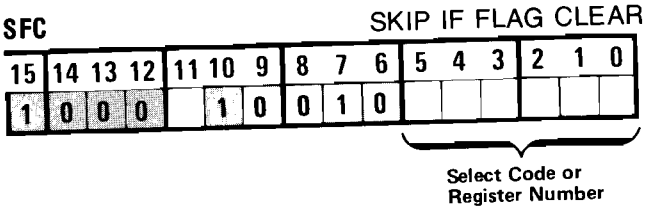
Outputs the contents of the A-register to the addressed I/O buffer or special function register. The contents of the A-register are not altered.

OTB OUTPUT B



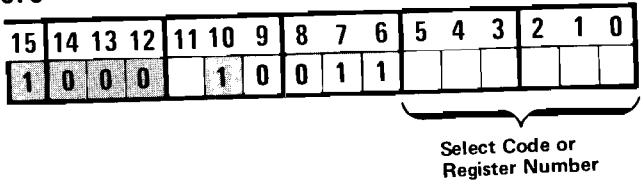
Outputs the contents of the B-register to the addressed I/O buffer or special function register. The contents of the B-register are not altered.

SFC SKIP IF FLAG CLEAR



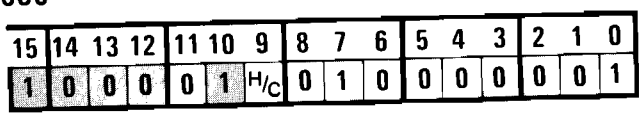
Skips the next programmed instruction if the flag (Flag 30) of the selected channel is clear (device busy).

SFS SKIP IF FLAG SET



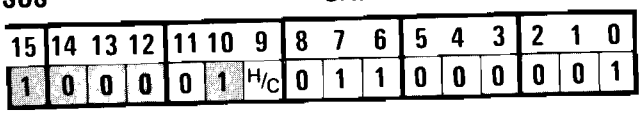
Skips the next programmed instruction if the flag (Flag 30) of the selected channel is set (device ready).

SOC SKIP IF OVERFLOW CLEAR



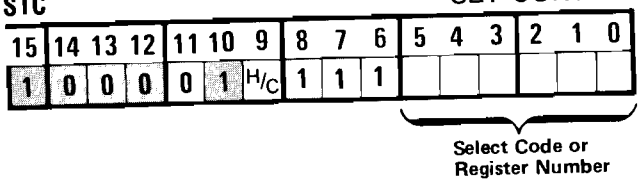
Skips the next programmed instruction if the overflow bit is clear. Use the H/C bit (bit 9) to either hold or clear the overflow bit following the completion of this instruction (whether the skip is taken or not).

SOS SKIP IF OVERFLOW SET



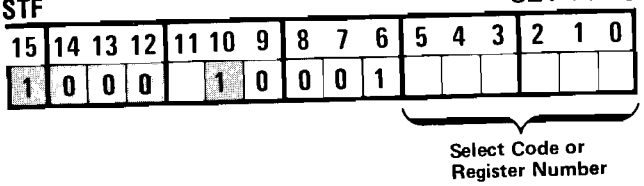
Skips the next programmed instruction if the overflow bit is set. Use the H/C bit (bit 9) to either hold or clear the overflow bit following the completion of this instruction (whether the skip is taken or not).

STC SET CONTROL



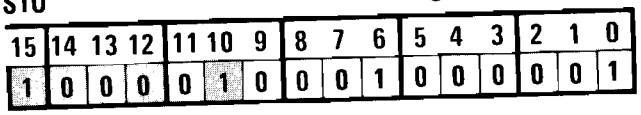
Sets the control bit (Control 30) of the selected I/O channel or function.

STF SET FLAG



Sets the flag (Flag 30) of the selected I/O channel or function. An STF 00 instruction enables the interrupt system for the time base generator and all interface cards.

STO SET OVERFLOW



Sets the overflow bit.

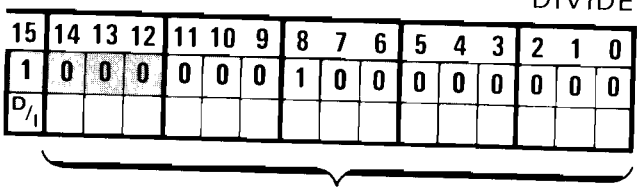
3-20. EXTENDED ARITHMETIC MEMORY REFERENCE INSTRUCTIONS

The four extended arithmetic memory reference instructions provide for integer multiply and divide and for loading and storing double-length words to and from the A- and B-registers. The complete instruction requires two

words: one for the instruction code and one for the address. When stored in memory, the instruction word is the first to be fetched; the address word is in the next sequential location.

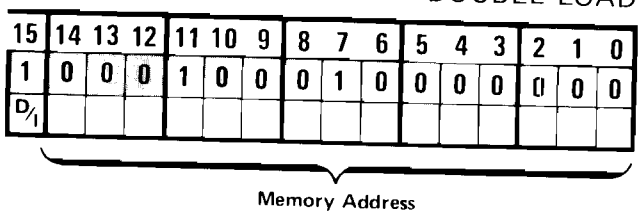
Since 15 bits are available for the address, these instructions can directly address any location in memory. As for all memory reference instructions, indirect addressing to any number of levels may also be used. A logic 0 in bit position 15 specifies direct addressing; a logic 1 specifies indirect addressing.

DIV DIVIDE



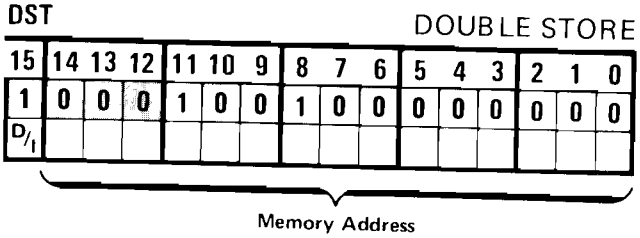
Divides a double-word integer in the combined B- and A-registers by a 16-bit integer in the addressed memory location. The result is a 16-bit integer quotient in the A-register and a 16-bit integer remainder in the B-register. Overflow can result from an attempt to divide by zero, or from an attempt to divide by a number too small for the dividend. In the former case (divide by zero), the division will not be attempted and the B- and A-register contents will be unchanged except that a negative quantity will be made positive. In the latter case (divisor too small), the execution will be attempted with unpredictable results left in the B- and A-registers. If there is no divide error, the overflow bit is cleared.

DLD DOUBLE LOAD



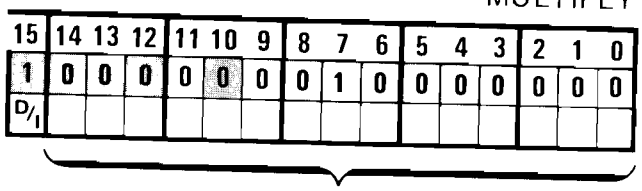
Loads the contents of addressed memory location m (and m + 1) into the A- and B-registers, respectively.

DST DOUBLE STORE



Stores the double-word quantity in the A- and B-registers into addressed memory locations m (and m + 1), respectively.

MPY MULTIPLY



Multiplies a 16-bit integer in the A-register by a 16-bit integer in the addressed memory location. The resulting double-length integer product resides in the B- and A-registers, with the B-register containing the sign bit and the most-significant 15 bits of the quantity. The A-register may be used as an operand (i.e., memory address 0), resulting in an arithmetic square. Overflow cannot occur; hence this instruction always clears the overflow bit.

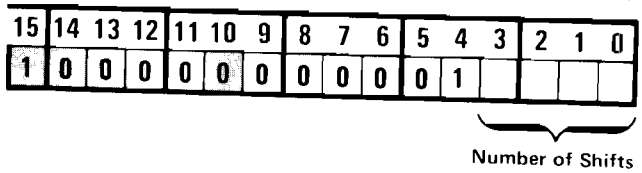
3-21. EXTENDED ARITHMETIC REGISTER REFERENCE INSTRUCTIONS

The six extended arithmetic register reference instructions provide various types of shifting operations on the combined contents of the B- and A-registers. The B-register is considered to be to the left (most-significant word) and the A-register is considered to be to the right (least-significant word). An example of each type of shift operation is illustrated in figure 3-4.

The complete instruction is given in one word and includes four bits (unshaded) to specify the number of shifts (1 to 16). By viewing these four bits as a binary-coded number, the number of shifts is easily expressed; i.e., binary-coded 1 = 1 shift, binary-coded 2 = 2 shifts . . . binary-coded 15 = 15 shifts. The maximum number of 16 shifts is coded with four zeros, which essentially exchanges the contents of the B- and A-registers.

The extend bit is not affected by any of the following instructions. Except for the arithmetic shifts, overflow also is not affected.

ASL ARITHMETIC SHIFT LEFT



Arithmetically shifts the combined contents of the B- and A-registers left n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated low-order positions of the A-register. The sign bit is not affected, and data bits are lost out of bit position 14 of the B-register. If any one of the lost bits is a significant data bit ("1" for positive numbers, "0" for negative numbers), the overflow bit will be set; otherwise, overflow will be cleared during execution. See ASL example in figure 3-4. Note that two additional shifts in this example would cause an error by losing a significant '1'.

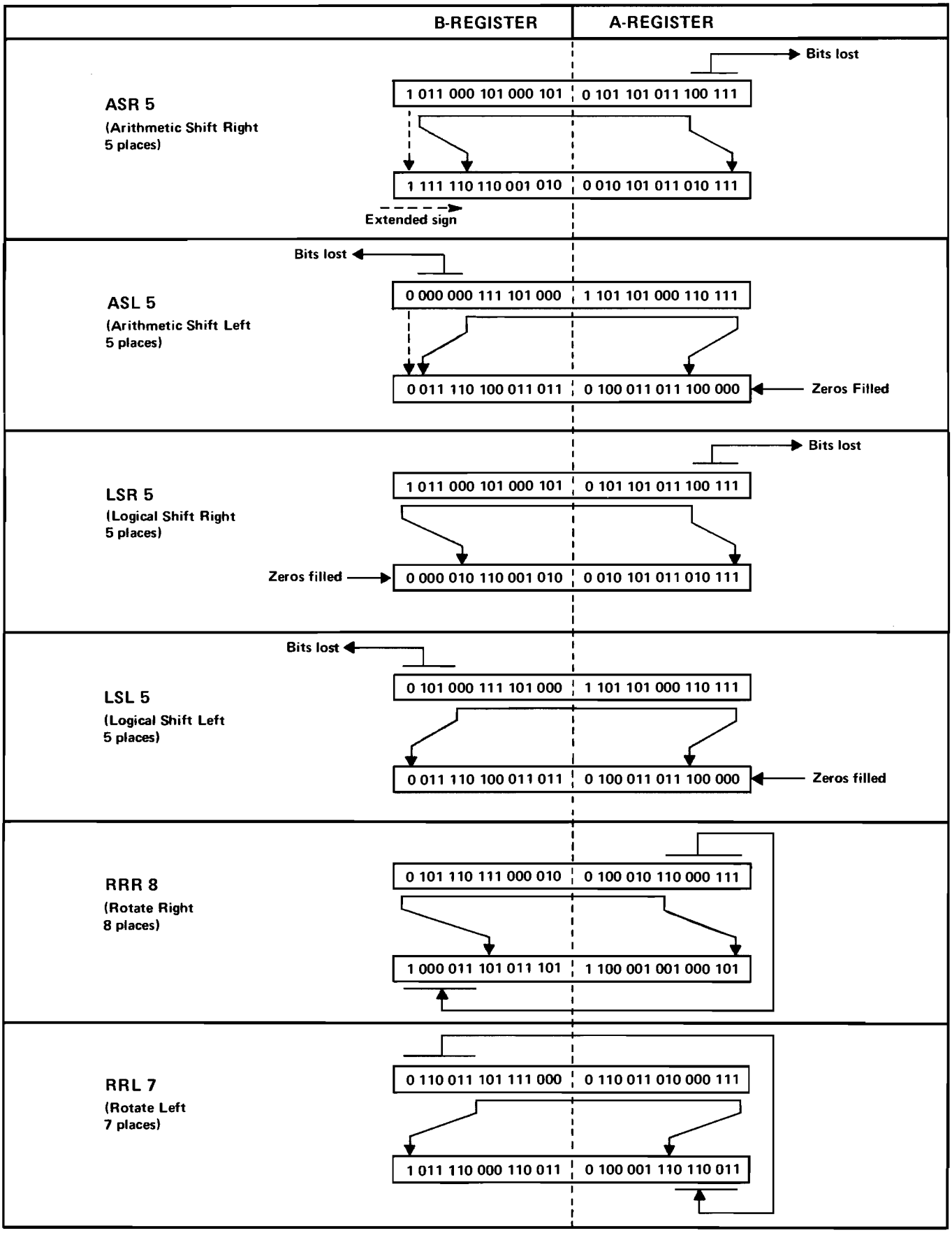
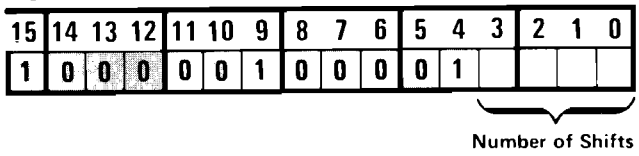


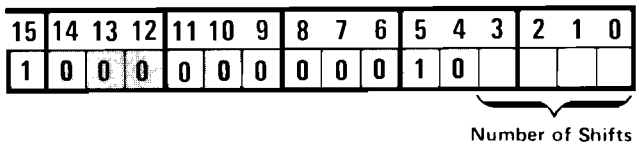
Figure 3-4. Examples of Double-Word Shifts and Rotates

ASR ARITHMETIC SHIFT RIGHT



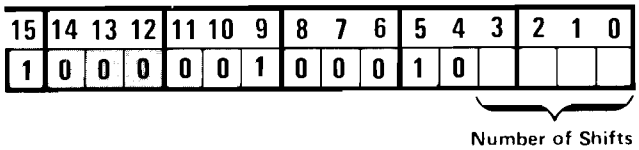
Arithmetically shifts the combined contents of the B- and A-registers right n places. The value of n may be any number from 1 through 16. The sign bit is unchanged and is extended into bit positions vacated by the right shift. Data bits shifted out of the least-significant end of the A-register are lost. Overflow cannot occur because the instruction clears the overflow bit.

LSL LOGICAL SHIFT LEFT



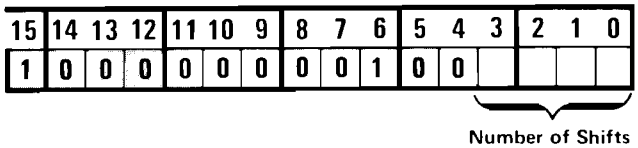
Logically shifts the combined contents of the B- and A-registers left n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated low-order bit positions of the A-register; data bits are lost out of the high-order bit positions of the B-register.

LSR LOGICAL SHIFT RIGHT



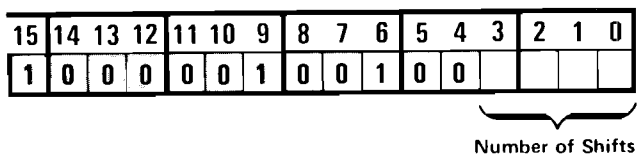
Logically shifts the combined contents of the B- and A-registers right n places. The value of n may be any number from 1 through 16. Zeros are filled into vacated high-order bit positions of the B-register; data bits are lost out of the low-order bit positions of the A-register.

RRL ROTATE LEFT



Rotates the combined contents of the B- and A-registers left n places. The value of n may be any number from 1 through 16. No bits are lost or filled in. Data bits shifted out of the high-order end of the B-register are rotated around to enter the low-order end of the A-register.

RRR ROTATE RIGHT



Rotates the combined contents of the B- and A-registers right n places. The value of n may be any number from 1 through 16. No bits are lost or filled in. Data bits shifted out of the low-order end of the A-register are rotated around to enter the high-order end of the B-register.

3-22. INSTRUCTION EXECUTION TIMES

Table 3-5 lists the execution times required for the computer instructions. Since the timing requirements of the input/output group instructions depend on the response time of the peripheral device involved, programs should not rely on the execution times for accurate, real-time measurements.

Table 3-5. Instruction Execution Times

INSTRUCTION	EXECUTION TIME (μS)*
Memory Reference Group	
ADA/B, AND, IOR, XOR	4.5
LDA/B, STA/B	4.1
CPA/B (no skip)	4.5
(skip)	5.0
ISZ (no skip)	5.9
(skip)	6.4
JMP	2.7
JSB	4.1
indirect (per level)	1.8
Shift/Rotate Group	3.2
Alter/Skip Group	3.6
Input/Output Group	
STC, CLC, STF, CLF, STO, CLO, SOS, SOC	3.6
LIA/B, MIA/B, OTA/B	6.4
HLT	15.2
SFS, SFC (no skip)	3.6
(skip)	5.0
Extended Arithmetic Memory Reference Instructions	
DLD, DST	7.7
MPY	28.1
DIV	8.6 - 33.1
Extended Arithmetic Register Reference Instructions	
(with one shift)	2.3
(per additional shift)	0.45

*Assumes no DMA intervention; concurrent DMA intervention may increase execution time.

INTERRUPT SYSTEM

SECTION IV

The vectored priority interrupt system has up to 60 distinct interrupt levels, each of which has a unique priority assignment. In the L-Series computer, the interrupt priority of an I/O card is based on the card's proximity to the processor card and is independent of the card's select code. The I/O card in the slot directly below the processor card has the highest interrupt priority. Each I/O card has higher interrupt priority than other I/O cards farther from the processor card and lower priority than cards closer to the processor card. As shown in table 4-1, the select code of an interrupt level is associated with an interrupt location in memory.

Any device can be selectively enabled or disabled under program control, thus switching the device into or out of the interrupt structure. In addition, the interrupt system is divided into types of interrupts (table 4-1). Interrupt Type 3 can be enabled or disabled under program control using a single instruction and interrupt Types 2 and 3 combined can be enabled or disabled using a single instruction.

4-1. POWER FAIL INTERRUPT

The computer power supply is equipped with power-sensing circuits. When primary line power fails or drops below a predetermined level while the computer is running, an interrupt to memory location 00004 is automatically generated. Memory location 00004 is intended to contain a jump-to-subroutine (JSB) instruction referencing the entry point of a user-written power fail subroutine. The interrupt capability of lower-priority operations is automatically inhibited while a power fail subroutine is in process.

A minimum of five milliseconds is available between the detection of a power failure and the loss of usable power supply power to execute a power fail subroutine; the purpose of such a routine is to transfer the current state of the computer system into memory and then halt the computer. A sample power fail subroutine is given in table 4-2. The battery module will supply enough power to preserve the contents of memory for a sustained line power outage of up to 60 minutes.

The user has a switch-selectable option of what action the computer will take upon restoration of primary power. When switch U1S8 is closed, the computer will execute either a loader or the virtual control panel routine, depending on the setting of the Start-Up switches.

NOTE

Switch U1S8 is mounted on the processor card and is not an operator control. The setting of this switch is normally determined prior to or during system installation.

When switch U1S8 is open, the automatic restart feature is enabled. After the self-test is executed following the return to normal power levels, an interrupt to location 00004 occurs. This time the power-down portion of the subroutine is skipped and the power-up portion begins. (Refer to table 4-2.) Those conditions existing at the time of the power fail interrupt are restored and the computer continues the program from the point of the interruption.

Note that an auto-restart interrupt to location 00004 occurs only if that location's contents are not zero; otherwise, the system is re-booted. This is done so that if power fails and is restored during a boot, an attempt to restart a partially loaded program can be avoided. To enable this to happen the program being loaded should initially load location 00004 with zero and load the power-fail JSB instruction only when the load is otherwise complete.

If the computer memory does not contain a subroutine to service the power fail interrupt, location 00004 should contain a NOP instruction (00 octal).

At the end of a restart routine, consideration should be given to re-initializing the power-fail logic and to restoring the interrupt capability of the lower priority functions.

4-2. PARITY ERROR INTERRUPT

Parity checking of memory is a standard feature in the L-Series computer. The parity logic continuously generates correct parity for all words written into memory and monitors the parity of all words read out of memory. Parity can be programmatically set (STF 05) to even parity or cleared (CLF 05) to odd parity. Correct odd parity is defined as having the total number of "1" bits in a 17-bit memory word (16 data bits plus the parity bit) equal to an odd value. If a "1" bit (or any odd number of "1" bits) is either dropped or added in the transfer process, a Parity Error signal is generated when that word is read out of memory.

The Parity Error signal will generate an interrupt to memory location 00005. This location may contain either a JSB instruction referencing the entry point of a user-

Table 4-1. L-Series Interrupt Assignments

SELECT CODE (OCTAL)	INTERRUPT LOCATION	ASSIGNMENT	INTER-RUPT TYPE
04	00004	Power Fail Interrupt	2
05	00005	Memory Parity Interrupt	1
06	00006	Time Base Generator Interrupt	3
07	00007	Memory Protect Interrupt	2
10	00010	Unimplemented Instruction Interrupt	1
11-16	00011-00016	Reserved	
17	00017	Special Interrupt	2
20-77	00020-00077	I/O Card Interrupts	3

written parity error subroutine or a HLT instruction. A parity error during a DMA transfer causes an interrupt to the memory location corresponding to the select code of the I/O card making the transfer.

The memory address containing the parity error will be loaded automatically into the parity register and from there it is accessible to the user by a programmed LIA 05 or LIB 05 instruction. Parity register bit 15 equals 1 if the error occurred during an instruction fetch and equals 0 if the error occurred during a data transfer.

If a parity error occurs during a read of an instruction, that instruction is not executed. When a parity error occurs, it is recommended that the entire program or set of data containing the error location be reloaded.

4-3. MEMORY PROTECT INTERRUPT

The memory protect feature provides the capability of protecting a selected block of memory of any size, from a settable fence address downward, against alteration or entry by programmed instructions except those involving the A- and B-registers.

The memory protect logic, when enabled by an STC 07 instruction, also prohibits the execution of all I/O instructions except those referencing I/O select code 01 (the processor card status register and the overflow register). (Execution of *all* HLTs is prohibited.) This feature limits control of I/O operations to interrupt control only. Thus, an executive program residing in protected memory can have exclusive control of the I/O system.

The memory protect logic is disabled automatically by any interrupt (except when the interrupt location contains an I/O instruction) and must be re-enabled by an STC 07 instruction at the end of each interrupt subroutine.

Programming rules pertaining to the use of memory protect are as follows (assuming that an STC 07 instruction has been given):

a. Location 00002 is the lower boundary of protected memory. (Locations 00000 and 00001 are the A- and B-register addresses.)

b. JMP instructions may not reference the A- or B-register; however, a JSB instruction may do so.

c. The upper protected memory boundary address is loaded into the fence register from the A- or B-register by an OTA 07 or OTB 07 instruction, respectively. Memory locations below but not including this address are protected.

d. Execution will be inhibited and an interrupt to location 07 will occur if a JMP, JSB, ISZ, STA, STB, or DST instruction directly addresses a location in protected memory, or if any I/O instruction is attempted (excluding those addressing select code 01). After three successive levels of indirect addressing, the logic will allow a pending I/O interrupt.

e. Any instruction not mentioned in step d of this paragraph is legal even if the instruction directly references a protected memory address. In addition, indirect addressing through protected memory by those instructions listed in step d is legal provided that the ultimate effective address is outside the protected memory area.

Following a memory protect interrupt, the address of the illegal instruction will be present in the violation register. This address is made accessible to the programmer by an LIA 07 or LIB 07 instruction, which loads the address into the A- or B-register.

Note that DMA operation is not affected by memory protect.

4-4. UNIMPLEMENTED INSTRUCTION INTERRUPT

An unimplemented instruction interrupt (to memory location 00010) is requested when the CPU chip signals that the last instruction fetched was not recognized by itself or any other system card. This interrupt provides a straightforward entry to software routines for the execution of instruction codes not recognized by the computer's hardware. The unimplemented instruction interrupt must receive immediate service in order to recover the instruction code that caused it. For this reason, and because it is desirable to permit the use of unimplemented instructions anywhere, the unimplemented instruction interrupt is never inhibited.

Table 4-2. Sample Power Fail Subroutine

LABEL	OPCODE	OPERAND	COMMENTS
PFAR	NOP		Power Fail/Auto Restart Subroutine
	SFC	4B	Skip if interrupt was caused by a power failure
	JMP	UP	Power being restored; reset state of system
DOWN	STA	SAVA	Save A-register contents
	CCA		Set flag indicating that computer was running when power failed
	STA	PFFLG	
	STB	SAVB	Save B-register contents
	ERA,ALS		Transfer E-register content to A-register bit 15
	SOC		Increment A-register if Overflow is set
	INA		
	STA	SAVEO	Save E- and O-register contents
	LDA	PFAR	Save contents of P-register at time of power failure
	STA	SAVP	
	.		
	.		Insert user-written routine to save I/O device states
	.		
	CLC	0B	Shut down any DMA or I/O
	SFS	4B	
	JMP	*−1	Wait in case power comes back up
UP	LDA	PFFLG	Was computer running when power failed?
	SZA,RSS		
	HLT	4B	No, then halt
	CLA		Yes, reset computer Run flag to initial state
	STA	PFFLG	
	LDA	FENCE	Restore the memory protect fence register contents (if MP used)
	OTA	7B	
	.		
	.		Insert user-written routine to restore I/O device states
	.		
	LDA	SAVEO	Restore the contents of the E-register and O-register
	CLO		
	SLA,ELA		
	STF	1B	
	LDA	SAVA	Restore A-register contents
	LDB	SAVB	Restore B-register contents
	STC	4B	Reset power fail logic for next power failure
	STC	7B	Turn on memory protect (if used)
	JMP	SAVP,I	Transfer to program executing at power failure
FENCE	OCT	2000	Fence address storage (update when fence is changed)
SAVEO	OCT	0	Storage for E and O
SAVA	OCT	0	Storage for A
SAVB	OCT	0	Storage for B
SAVP	OCT	0	Storage for P
PFFLG	OCT	0	Storage for Run flag

Note: The interrupt mask register (if used) must be saved and restored; the memory protect state must be saved and conditionally restored. Also, save the global register if used.

4-5. TIME BASE GENERATOR INTERRUPT

A time base generator interrupt request is made when the CPU chip signals that its internal clock divider chain has rolled over. The clock divider is set to roll over at 10-millisecond intervals for maintaining a real-time clock. The interrupt occurs through location 00006 and can be masked (inhibited) by using bit 1 of the interrupt mask register. (The interrupt mask register allows interrupts from the TBG and the I/O cards to be selectively masked. For details on the interrupt mask register, refer to the *HP 1000 L-Series Computer I/O Interfacing Guide,* part no. 02103-90005.)

4-6. SPECIAL INTERRUPT

The special interrupt is reserved for possible future use by Hewlett-Packard.

4-7. INPUT/OUTPUT INTERRUPT

Interrupt locations 20 through 77 (octal) are reserved for I/O devices. In a typical I/O operation, the computer issues a programmed command such as Set Control/Clear Flag (STC,C) to one or more external devices to initiate an input (read) or an output (write) operation, via either programmed I/O or DMA. While the I/O card is in the process of transferring data, the computer may be either running a program or looping, waiting for a flag to get set. Upon completion of the read or write operation, the interface flag is set. If the corresponding control bit is set, the interface will interrupt. Its request will be passed through a priority network so that only the highest priority interrupting device will receive service. The computer will acknowledge the interrupt and the highest priority device will receive service when the current instruction has finished executing, except under the following circumstances:

a. Interrupt system disabled or interface card interrupt disabled (or masked).

b. JMP indirect or JSB indirect instruction not sufficiently executed. These instructions inhibit all interrupts except power fail or memory protect until the succeeding instruction is executed. After three successive levels of indirect addressing, the logic will allow a pending I/O interrupt.

c. A DMA (direct memory access) data transfer is in process.

d. Current instruction is any I/O instruction. The interrupt in this case must wait until the succeeding instruction is executed.

After an interface card has been issued a Set Control (STC instruction) and its flag bit becomes set, all interrupt requests from lower-priority devices are inhibited until this flag bit is cleared by a Clear Flag (CLF) instruction. A service subroutine in process for any device can be interrupted only by a higher-priority device; then, after the higher-priority device is serviced, the interrupted service subroutine may continue. In this way it is possible for several service subroutines to be in the interrupt state at one time; each of these service subroutines will be allowed to continue after the higher-priority device is serviced. All such service subroutines normally end with a JMP indirect instruction to return the computer to the point of interrupt.

4-8. INTERRUPT PRIORITY

The interrupt servicing priority among the system interrupts is as follows:

a. Parity error
b. Unimplemented instruction
c. Memory protect
d. Special interrupt
e. Power fail
f. Time base generator
g. I/O interrupts

4-9. CENTRAL INTERRUPT REGISTER

Each time an interrupt occurs, the address of the interrupt location is stored in the central interrupt register. The contents of this register are accessible at any time by executing an LIA 04 or LIB 04 instruction. This loads the address of the most recent interrupt into the A- or B-register.

4-10. PROCESSOR STATUS REGISTER

The processor card status register is two registers: one for input and one for output. The input register shows the status of the processor card switches, MLOST signal, and the TBG mask bit. The input register is read into the upper eight bits of the A- or B-register by an LIA/B 01 instruction. The switch, bit, and function relationships are as follows:

SWITCH (U1)	BIT	MEANING
S1	8	Boot select
S2	9	Boot select
S3	10	Boot select
S4	11	Reserved
S5	12	VCP program select
S6	13	Not used
S8	14	MLOST
	15	Mask bit for TBG

Note that switch S7 is not read.

The output register drives the processor card LED's. The contents of the lower seven bits of the A- or B-register are inverted (1 = off) and sent to the LED's by an OTA/B 01 instruction.

4-11. INTERRUPT TYPE CONTROL

I/O address 00 is the master control address for Type 3 interrupts (TBG and I/O cards). An STF 00 instruction enables Type 3 interrupts and a CLF 00 disables Type 3 interrupts. (Type 3 interrupts are disabled when power is initially applied.) I/O address 04 is the master control address for Type 2 interrupts (power fail, memory protect, and special) and Type 3 interrupts combined. An STC 04 instruction enables Type 2 and 3 interrupts and a CLC 04 disables Type 2 and 3 interrupts.

4-12. INSTRUCTION SUMMARY

Table 4-3 is a summary of instructions for select codes 00 through 07. For a summary of instructions used with the I/O cards, refer to an I/O card reference manual.

Table 4-3. Instructions for Select Codes 00 through 07

INSTRUCTION	FUNCTION	INSTRUCTION	FUNCTION
STC 0	NOP	STC 4	Enable Type 2 and 3 interrupts
CLC 0	System reset	CLC 4	Disable Type 2 and 3 interrupts
STF 0	Enable Type 3 interrupts	STF 4	NOP
CLF 0	Disable Type 3 interrupts	CLF 4	NOP
SFS 0	Skip if Type 3 interrupts enabled	SFS 4	Skip if power coming up
SFC 0	Skip if Type 3 interrupts disabled	SFC 4	Skip if power going down
LI* 0	Load from interrupt mask register	LI* 4	Load from central interrupt register
MI* 0	Merge from interrupt mask register	MI* 4	Merge from central interrupt register
OT* 0	Output to interrupt mask register	OT* 4	Output to central interrupt register
STC 1	NOP	STC 5	Enable parity error interrupts
CLC 1	NOP	CLC 5	Disable parity error interrupts
STF 1	Same as Set Overflow (STO)	STF 5	Set parity sense to even parity
CLF 1	Same as Clear Overflow (CLO)	CLF 5	Clear parity sense to odd parity
SFS 1	Same as Skip If Overflow Set (SOS)	SFS 5	Skip if parity sense is even
SFC 1	Same as Skip If Overflow Clear (SOC)	SFC 5	Skip if parity sense is odd
LI* 1	Load from processor card status register	LI* 5	Load from parity register
MI* 1	Merge from processor card status register	MI* 5	Merge from parity register
OT* 1	Output to processor card status register	OT* 5	Output to parity register
STC 2	Disable boot ROM	STC 6	Turn on time base generator
CLC 2	Enable boot ROM	CLC 6	Turn off time base generator
STF 2	Disable global register	STF 6	Set time base generator flag
CLF 2	Enable global register	CLF 6	Clear time base generator flag
SFS 2	Skip if global register disabled	SFS 6	Skip if time base generator flag set
SFC 2	Skip if global register enabled	SFC 6	Skip if time base generator flag clear
LI* 2	Load from global register	LI* 6	NOP
MI* 2	Merge from global register	MI* 6	NOP
OT* 2	Output to global register (Note 1)	OT* 6	NOP
STC 3	NOP	STC 7	Turn on memory protect
CLC 3	Transfer control to VCP after three fetches	CLC 7	NOP
STF 3	NOP	STF 7	NOP
CLF 3	NOP	CLF 7	NOP
SFS 3	NOP	SFS 7	NOP
SFC 3	NOP	SFC 7	NOP
LI* 3	Load from P SAVE	LI* 7	NOP
MI* 3	Merge from P SAVE	MI* 7	NOP
OT* 3	Output to P SAVE	OT * 7	Output to memory protect fence register
LI* 3,C	Load from ROM P	LI* 7,C	Load from violation register
MI* 3,C	Merge from ROM P	MI* 7,C	Merge from violation register
OT* 3,C	Output to ROM P	OT* 7,C	Output to violation register

* = A or B.

Note 1. An OTA/B 2 with A/B equal to one through seven establishes a diagnose mode; refer to paragraph 5-23 for details.

INPUT/OUTPUT SYSTEM

SECTION V

The purpose of the input/output system is to transfer data between the computer and external devices. As shown in figure 5-1, data can be transferred either by a direct memory access (DMA) feature or through the A- or B-register in the CPU chip (non-DMA). Each L-series I/O card has DMA logic and DMA is normally used for most I/O data transfers. Once the DMA logic has been initialized, no programming is involved and the transfer occurs in two distinct steps as follows:

a. Between the external device and its I/O interface card in the computer;

b. Between the I/O card and memory via the backplane data bus.

This two-step process also applies to a DMA output transfer except in reverse order.

As mentioned above, data may be transferred under program control without using the DMA feature. This type of transfer allows the computer to manipulate the data during the transfer process. A non-DMA input transfer is a three-step process as follows:

a. Between the external device and its I/O card;

b. Between the I/O card and the A- or B-register via the data bus and the processor card; and

c. Between the A- or B-register and memory via the processor and the data bus.

Note that in the DMA transfer the processor card is bypassed. Since a DMA transfer eliminates programmed loading and storing via the accumulators, the time involved is very short. Further information on the DMA feature is given in paragraph 5-9.

5-1. INPUT/OUTPUT ADDRESSING

As shown in figure 5-2, an external device is connected by cable directly to an interface card located in the computer mainframe. The interface card, in turn, plugs into one of the input/output slots, each of which is assigned a fixed interrupt priority. Note, however, that the select code of the L-Series interface cards is independent of the priority. The computer communicates with a specific device on the basis of its select code which is set by switches on the interface card.

Figure 5-2 shows an interface card inserted in the I/O slot having the highest priority. If it is decided that the associated device should have lower priority, its interface card and cable may simply be exchanged with those occupying some other I/O slot. This will change the priority but not the I/O address (select code). Due to priority chaining, there can be no vacant slots from the highest priority slot to the lowest priority slot used. Only select codes 20 through 77 (octal) are available for input/output cards; the lower select codes (00 through 17) are reserved for other features.

5-2. INPUT/OUTPUT PRIORITY

The plug-in card slots of the L-Series computers are numbered 1 through 10 in the box computer and 1 through 16 in the computer system. Generally, slots 1 and 2 are used for the memory and processor cards and the remaining slots are available for I/O cards, with slot 3 having the highest I/O interrupt priority. An I/O channel consists of an I/O device (or devices) and its I/O card and is assigned the number of the card slot.

When an input/output device is ready to be serviced, it causes its interface card to request an interrupt so that the computer will interrupt the current program and service the device. Since many device interface cards will be requesting service at random times, it is necessary to establish an orderly sequence for granting interrupts. Also, it is desirable that high-speed devices should not have to wait for low-speed device transfers. Both of these requirements are met by a series-linked priority structure illustrated by figure 5-3. The bold line, representing a priority enabling signal, is routed in series through each card capable of causing an interrupt. The card cannot interrupt unless this enabling signal is present at its input.

Each device (or other interrupt function) can break the enabling line when it requests an interrupt. If two devices simultaneously request an interrupt, the device with the highest priority will be the first one that can interrupt because it has broken the enable line for the lower-priority device. The other device cannot begin its service routine until the first device is finished. However, a still higher-priority device (one interfaced through a lower-numbered slot) may interrupt the service routine of the first device. Figure 5-4 illustrates a hypothetical case in which several devices request service by interrupting a CPU program. Both simultaneous and time-separated interrupt requests are considered.

Assume that the computer is running a CPU program when an interrupt from I/O channel 5 occurs (at reference time t1), and that the card in slot 5 is assigned select code 22. With the I/O card supplying the select code, a JSB instruction in the interrupt location for select code 22 causes a program jump to the service routine for the

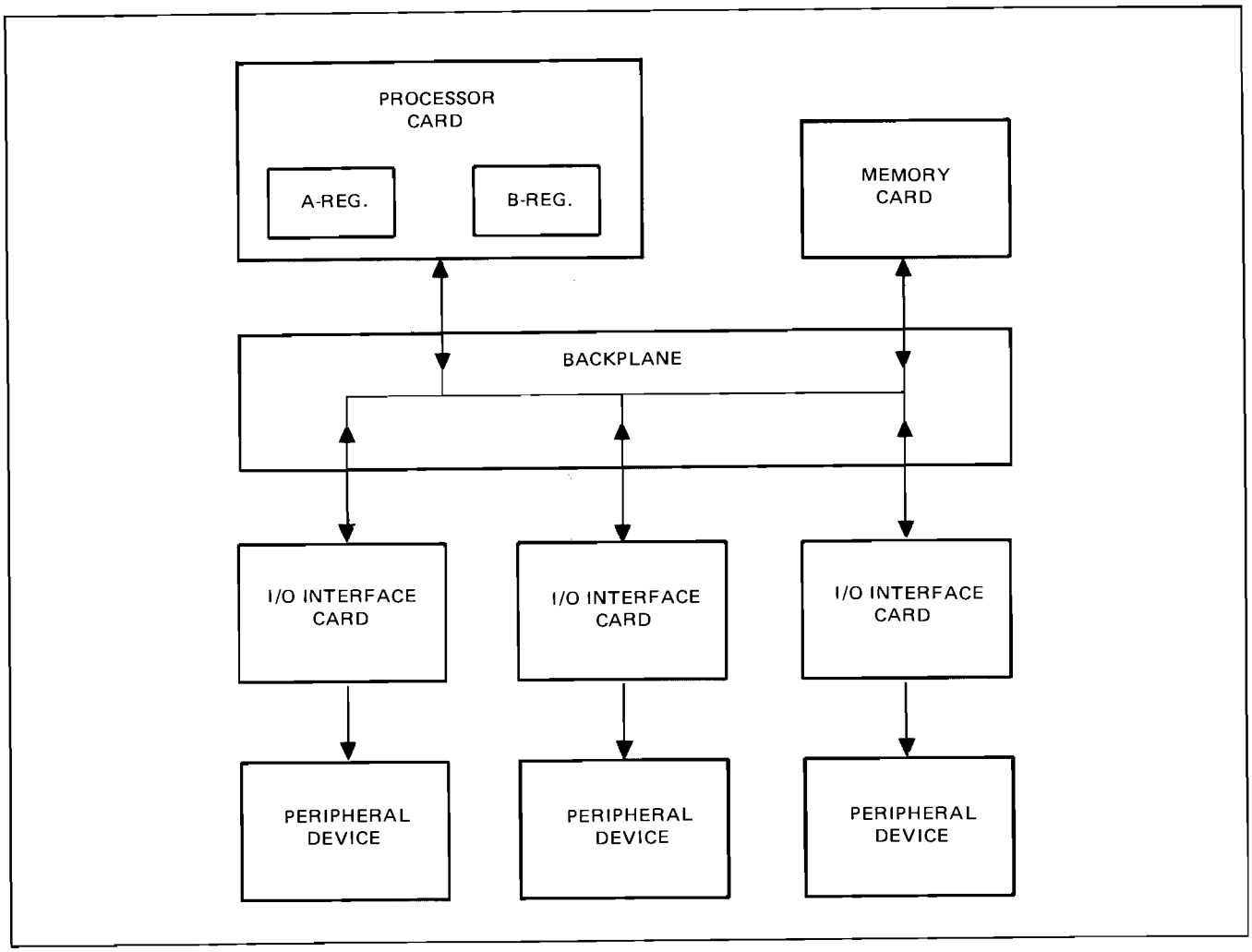
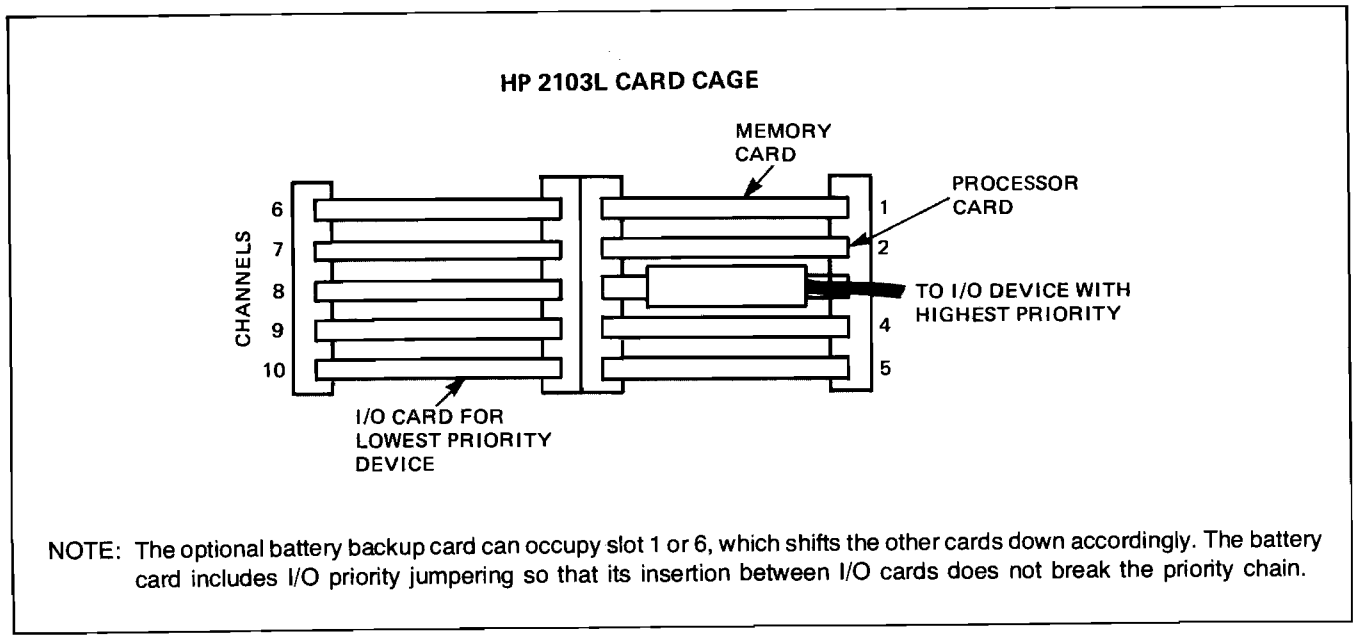


Figure 5-1. Input/Output System



NOTE: The optional battery backup card can occupy slot 1 or 6, which shifts the other cards down accordingly. The battery card includes I/O priority jumpering so that its insertion between I/O cards does not break the priority chain.

Figure 5-2. I/O Priority Assignments

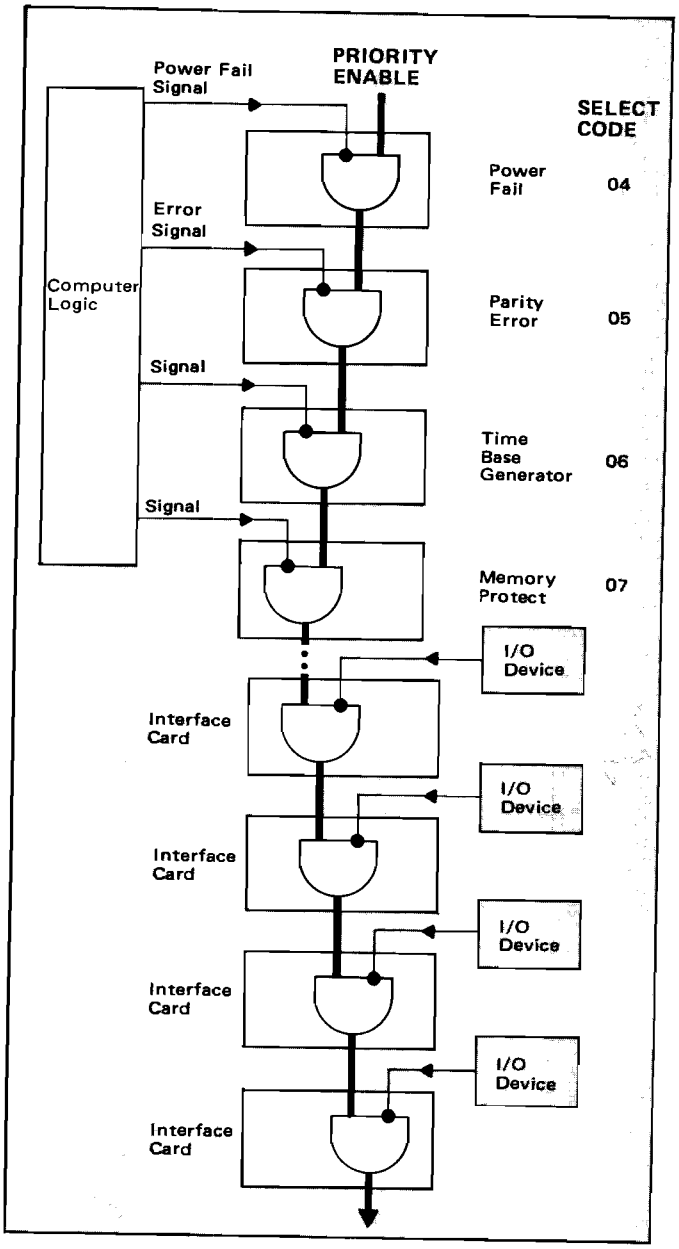


Figure 5-3. Priority Linkage (Simplified)

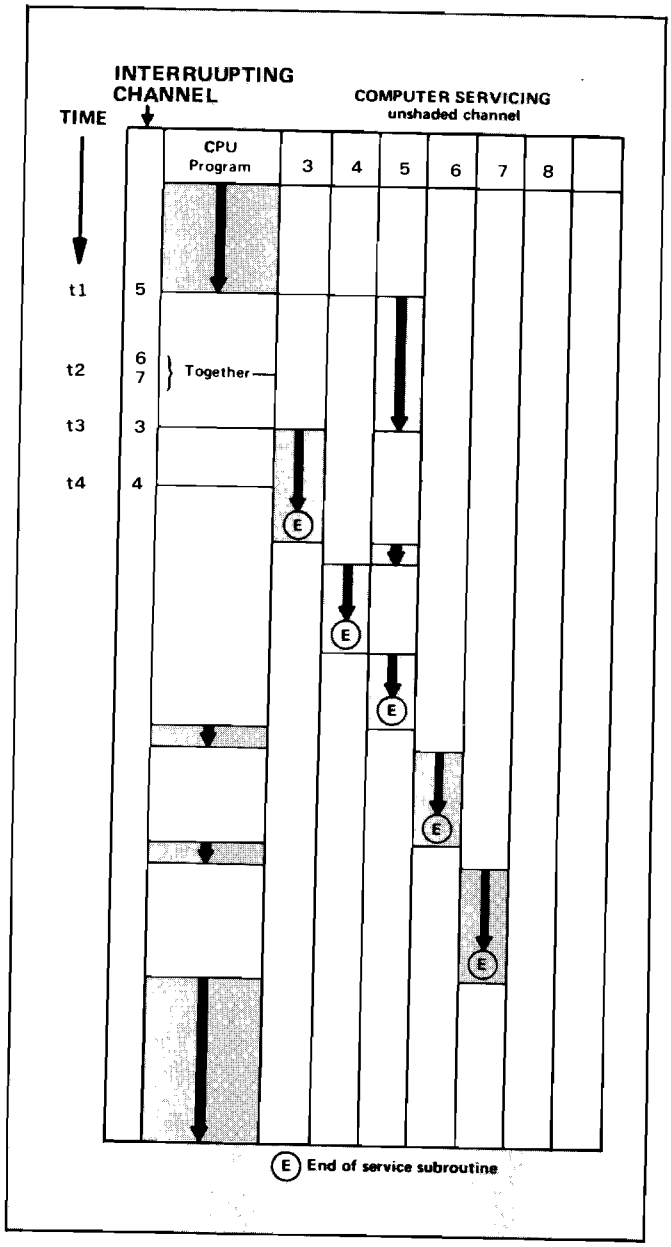


Figure 5-4. Interrupt Sequences

channel-5 device (select code 22). The JSB instruction automatically saves the return address (in a location which the programmer must reserve in his routine) for a later return to the CPU program.

The routine for channel 5 (select code 22) is still in progress when several other devices request service (set flag). First, channels 6 and 7 request simultaneously at time t2; however, since neither one has priority over channel 5, their flags are ignored and channel 5 continues transfer. But at t3, a higher priority device on channel 3 requests service. This request interrupts the channel 5 transfer and causes the channel 3 transfer to begin. The JSB instruction saves the return address for return to the channel 5 routine.

During the channel 3 transfer, the channel 4 flag is set (t4). Since it has lower priority than channel 3, channel 4 must wait until the end of the channel 3 routine. And since the channel 3 routine, when it ends, contains a return address to the channel 5 routine, program control temporarily returns to channel 5 (even though the waiting channel 4 has higher priority). The JMP,I instruction used for the return inhibits all interrupts until fully executed. At the end of this short interval, the channel 4 interrupt request is granted.

When channel 4 has finished its routine, control is returned to channel 5, which at last has sufficient priority to complete its routine. Since channel 5 has been saving a return address in the main CPU program, it returns control to this point.

The two waiting interrupt requests from channels 6 and 7 are now enabled. Channel 6 has the higher priority and goes first. At the end of the channel 6 routine control is temporarily returned to the CPU program. Then the lowest priority channel (channel 7) interrupts and completes its transfer. Finally, control is returned to the CPU program, which resumes processing.

5-3. INTERFACE ELEMENTS

The interface card provides the communication link between the computer and one or more external devices. The interface card includes several basic elements which either the computer or the device can control in order to effect the necessary communication. These basic elements are the global register, control bits, flag bits, data buffer register, and control register. Other registers, associated only with DMA, are discussed in paragraph 5-9. The control and flag bits and the data buffer and control registers of an interface card can be addressed directly when the card's select code is in the global register (GR) and the GR is enabled. Refer to the interface card reference manuals for specific information on the data and control registers.

5-4. GLOBAL REGISTER

In the L-Series computer there are two ways of specifying which select code is being addressed by an I/O instruction. The first is a method where the select code is contained in the lower six bits of the I/O instruction. The second is a method utilizing a new design feature called the "global register." The global register is a register on each I/O card that can be loaded with the select code of any one of the I/O cards. When the global register (GR) is enabled, any I/O instruction is executed only by the I/O card whose select code is in the GR. Also, the GR allows other registers on the selected I/O card to be accessed programmatically by I/O instructions. The global register on all I/O cards may be simultaneously loaded with an OTA/B 02 instruction, enabled with a CLF 02 instruction, and disabled with an STF 02 instruction.

5-5. CONTROL BITS

The control bits on an interface card are used to turn on a specific I/O function. In addition, a control bit must be set to allow the corresponding flag bit to interrupt. There are three control bits associated with each I/O select code: Control 20, 21, and 30. Control 30 is the only control bit that can be accessed with or without the global register being enabled. When Control 30 is set it generates an action command, allowing one word or character to be read or written. Control 20 and 21 can only be accessed when the global register is enabled. When Control 20 is set it turns on DMA self-configuration. The setting of Control 21 turns on DMA transfers.

5-6. FLAG BITS

The flag bits (when set) are used primarily to interrupt or to signal completion of a task. Flag 30, the only flag bit accessible without using the global register, signals either one data element has been transferred or that an interrupting condition has been detected. There are three other flags, all of which must be accessed with the global register enabled. Flag 21 signals DMA transfer complete; Flag 20 signals DMA self-configuring transfer complete; and Flag 22 signals parity error during DMA. The device cannot clear the flag bit. If the corresponding control bit is set, priority is high, and the interrupt system is enabled, then setting the flag bit will cause an interrupt to the location corresponding to the I/O card's select code.

5-7. DATA BUFFER REGISTER

The data buffer register (designated Register 30) is used for the intermediate storage of data during an I/O transfer. Typically, the data capacity is 16 bits.

5-8. CONTROL REGISTER

The control register (designated Register 31) enables a general purpose interface card to be configured for compatibility with a specific I/O device or to be programmed for particular modes of operation. The control register must be programmatically set up for a particular application.

5-9. DIRECT MEMORY ACCESS

The direct memory access (DMA) capability of each L-Series interface card provides a direct data path between memory and a peripheral device, making it practical to use DMA for most data transfers. The use of DMA to perform I/O data transfers reduces the number of interrupts from one per byte or word to one per complete DMA block transfer. (Maximum DMA block size is 65,536 bytes.)

The maximum DMA transfer rate is 2.7 million bytes per second; this is also the combined limit for DMA transfers by two or more I/O cards. Except when the DMA feature is operating at full bandwidth, the central processor can interleave memory cycles with the DMA operation. The DMA feature is provided by the following elements:

a. The common backplane that links the processor, memory, and I/O cards;

b. The capability of the I/O cards to execute I/O instructions; and

c. The global register which:

   1. Enables only the I/O card whose select code is in the global register to execute I/O instructions, freeing the address bits of the I/O instruction; and

2. Enables the I/O-instruction address bits to be used to access registers on the I/O card specified by the global register.

Each I/O card has four registers associated with DMA. Three of them must be loaded with control words that specify the DMA operation. The fourth register is used for a special type of DMA operation called self-configured DMA which is discussed later. All of these registers can be accessed only when the select code of the desired I/O card is in the global register. The DMA registers and their functions are as follows:

a. Register 20, DMA Self-Configuration Register;

b. Register 21 (for Control Word 1), DMA Control Register;

c. Register 22 (for Control Word 2), DMA Address Register; and

d. Register 23 (for Control Word 3), Word/Byte Count Register.

5-10. CONTROL WORD 1

Control Word 1 (CW1) must be loaded into Register 21 of the desired I/O card as part of the DMA initialization process. The general definitions of the bits in Control Word 1 are given in figure 5-5. Note that the requirements of individual I/O cards may vary slightly from the general definitions and that it is necessary to refer to the I/O card reference manuals for specific programming information.

5-11. CONTROL WORD 2

Control Word 2 (CW2) loads into Register 22 the address of the first memory location to be read from or stored into when the DMA operation is initiated. The most significant bit, bit 15, is not used by the DMA control logic; when CW2 is read for status, bit 15 is the complement of bit 7 in CW1 (figure 5-5).

5-12. CONTROL WORD 3

Control Word 3 (CW3) loads into Register 23 the two's-complement number of data elements to be transferred by DMA. Data elements may be either words or bytes as specified by bit 13 of CW1 (figure 5-5). The end of a DMA data transfer is indicated by the transition from −1 to 0 of the value in Register 23 (the Word/Byte Count Register); this causes the I/O card to generate a completion interrupt. (A DMA transfer can also be terminated in other ways as described in the interface card manuals.)

5-13. DMA TRANSFER INITIALIZATION

A DMA data transfer is started by:

a. Loading the global register with the select code of the desired I/O card;

b. Loading the three DMA registers: DMA control into Register 21, DMA address into Register 22, and word/byte count into Register 23;

c. Loading the control register (Register 31) of the I/O card (described in the individual interface card reference manuals); and

d. Issuing an STC instruction to Register 21 (DMA Control Register).

A typical programming sequence to configure the DMA logic for a DMA transfer is as follows:

```
LDA SC          Load select code
OTA 2,C         Set up global register
LDA CW1
OTA 21B         Output DMA control word
LDA CW2
OTA 22B         Output DMA starting address
LDA CW3
OTA 23B         Output DMA word/byte count
LDA CNT
OTA 31B         Output I/O card control word
STC 21B,C       Start DMA and device
<continue any other processing>
```

5-14. SELF-CONFIGURED DMA

Each I/O card also has logic that can automatically load the DMA registers discussed previously with the DMA control words from sequential locations in memory. This process is performed by using the I/O card's Register 20, the Self-Configuration Register. The DMA self-configuration feature is initialized by setting the value of Register 20 to the memory address of a list of DMA "triplets" or "quadruplets".

A triplet is of the form control bits, DMA transfer address, and word/byte count. The triplet words are the words to be loaded into Registers 21, 22, and 23, respectively. A quadruplet is of the form control bits, I/O-card control word, transfer address, and word/byte count. Bit 8 of the control bits (Control Word 1) determines whether a triplet or quadruplet is loaded. (A quadruplet is used only when the I/O-card control word must be changed; refer to the interface card manuals for detailed information.) As each register is loaded, the contents of Register 20 are incremented, leaving it pointing to the memory location to be loaded into the next register.

DMA self-configuration can be chained to enable consecutive DMA transfers via the same I/O card with a minimum of interrupts. If bit 15 of Control Word 1 in a triplet (or

15	14	13	12	11	10	9	8	7	6 5	4 0
CONT	DVCMD	BYTE	RES	CINT	REM	FOUR	AUTO	IN	Various	EXT ADDR BUS

CONT (Continue), bit 15.

Bit 15 = 1: Enable a DMA re-configuration upon completion of a self-configured DMA transfer.

Bit 15 = 0: Stop DMA after current transfer.

DVCMD (Device Command), bit 14.

Bit 14 = 1: Issue a Device Command signal for each data element transferred.

Bit 14 = 0: No Device Command signal issued.

BYTE (Byte/word transfer), bit 13.

Bit 13 = 1: Conduct DMA transfer in byte mode.

Bit 13 = 0: Conduct DMA transfer in word mode.

RES (Residue), bit 12.

Bit 12 = 1: Write word/byte count back into memory.

Bit 12 = 0: Word/byte count is not written.

CINT (Completion Interrupt), bit 11.

Bit 11 = 1: Inhibit DMA completion interrupt.

Bit 11 = 0: Request completion interrupt when word/byte count goes from −1 to 0 and bit 15 equals 0.

REM (Remote), bit 10.

Bit 10 = 1: Enable remote (non-standard) memory for DMA transfer.

Bit 10 = 0: Remote memory not enabled.

FOUR (Fetch four control words), bit 9.

Bit 9 = 1: Causes DMA self-configuration to fetch four control words; i.e., three DMA control words and one I/O card control word.

Bit 9 = 0: Fetch three control words for DMA self-configuration.

AUTO (Automatic), bit 8.

Bit 8 = 1: Initiate first data transfer once DMA is configured to output, without waiting for an SRQ. For input transfers, enable a Device Command signal after the last data element is transferred.

Bit 8 = 0: For output transfers, wait for a Service Request (SRQ) signal before performing the first transfer. For input transfers, the last data element is not followed by a Device Command.

IN (Transfer In), bit 7.

Bit 7 = 1: Perform DMA transfer from I/O device to memory.

Bit 7 = 0: Perform DMA transfer from memory to I/O device.

Various, bits 5 and 6, User definable.

EXT ADDR BUS, bits 4-0.

These five bits allow DMA accesses to extended memory by referencing one of 32 offset registers. These offset registers are stored in memory locations 140-177. Adding the contents of the offset register to the upper 5 bits of the address bus produces the 18 bit DMA address.

Figure 5-5. General Bit Definitions for Control Word 1

quadruplet) is a logic 1, the DMA registers will be loaded with the next triplet in memory (as pointed to by Register 20) upon completion of the current DMA block transfer. When bit 15 (and bit 11) is a logic 0, the current DMA block transfer is followed by a completion interrupt.

5-15. DMA DATA TRANSFER

Figure 5-6 illustrates, in general, the sequence of operations for a DMA input data transfer (the minor differences for an output transfer are explained in text). Note that the global register has been enabled and loaded with the I/O card's select code.

The initialization routine sets up the DMA control registers on the I/O card (1) and issues the start command (STC 21,C) to the DMA Control Bit (Control 21). (If the operation is an output, the I/O card buffer is also loaded at this time.) The DMA logic is now turned on and the computer program continues with other instructions.

Setting the DMA Control bit (2) causes the I/O card to send a Start signal (with a data word if it is an output transfer) to the external device (3). The device goes through a read or write cycle and returns a Done signal (with a data word if it is an input transfer). The Done signal (4) requests the DMA logic (5) to transfer a word into (or out of) memory (6). The process now loops back to step 3 to transfer the next word.

After the specified number of words have been transferred, the DMA logic generates a completion interrupt (7). The program control is now forced to a completion routine (8), the content of which is the programmer's responsibility.

For more detailed information on DMA, refer to the I/O interfacing guide, part no. 02103-90005.

5-16. NON-DMA DATA TRANSFER

The following paragraphs describe how data is transferred between memory and input/output devices without using DMA. The sequences presented are simplified in order to present an overall view without the involvement of software operating systems or device drivers.

5-17. INPUT DATA TRANSFER (INTERRUPT METHOD)

Figure 5-7 illustrates the sequence of events required to input data using the interrupt method. Note that some operations are under control of the computer program (programmer's responsibility) and some of the operations are automatic. Note also that the global register has been loaded and enabled and the I/O card's control register has been loaded.

The operations begin (1) with the programmed instruction STC 30,C which sets the Control bit (Control 30) and clears the Flag bit (Flag 30) on the I/O card. Since the next few operations are under control of the hardware, the computer program may continue the execution of other instructions. Setting the Control bit causes the card to output a Start signal (2) to the device, which reads out a data character and asserts the Done signal (3).

The device Done signal sets the Flag bit, which in turn generates an interrupt (4) provided that the interrupt conditions are met; i.e., the interrupt system must be on (STF 00 previously given), no higher priority interrupt is pending, and the Control bit is set (done in step 1).

The interrupt causes the current computer program to be suspended and control is transferred to a service subroutine (5). It is the programmer's responsibility to provide the linkage between the interrupt location (which agrees with the select code) and the service subroutine. It is also the programmer's responsibility to include in his service subroutine the instructions for processing the data (loading into an accumulator, manipulating if necessary, and storing into memory).

The subroutine may then issue further STC 30,C instructions to transfer additional data characters. One of the final instructions in the service subroutine must be CLC 30,C. This step (6) restores the interrupt capability to lower priority devices and returns the I/O card to its static "reset" condition (Control clear and Flag clear). This condition is initially established by the computer at power turn-on and it is the programmer's responsibility to return the I/O card to the same condition on the completion of each data transfer operation. At the end of the subroutine, control is returned to the interrupted program via previously established linkages.

5-18. OUTPUT DATA TRANSFER (INTERRUPT METHOD)

Figure 5-8 illustrates the sequence of events required to output data using the interrupt method. Again note the distinction between programmed and automatic operations. Note also that the global register has been loaded and enabled and that the I/O card's control register has been loaded. It is assumed that the data to be transferred has been loaded into the A-register and is in a form suitable for output.

The output operation begins with a programmed instruction (OTA 30) to transfer the contents of the A-register to the I/O card buffer (1). This is followed (2) by the instruction STC 30,C which sets the Control bit (Control 30) and clears the Flag bit (Flag 30) on the I/O card. Since the next few instructions are under control of the hardware, the computer program may continue the execution of other instructions. Setting the Control bit causes the card to output the buffered data and a Start signal (3) to the device, which writes (e.g., records, stores, etc.) the data character and asserts the Done signal (4).

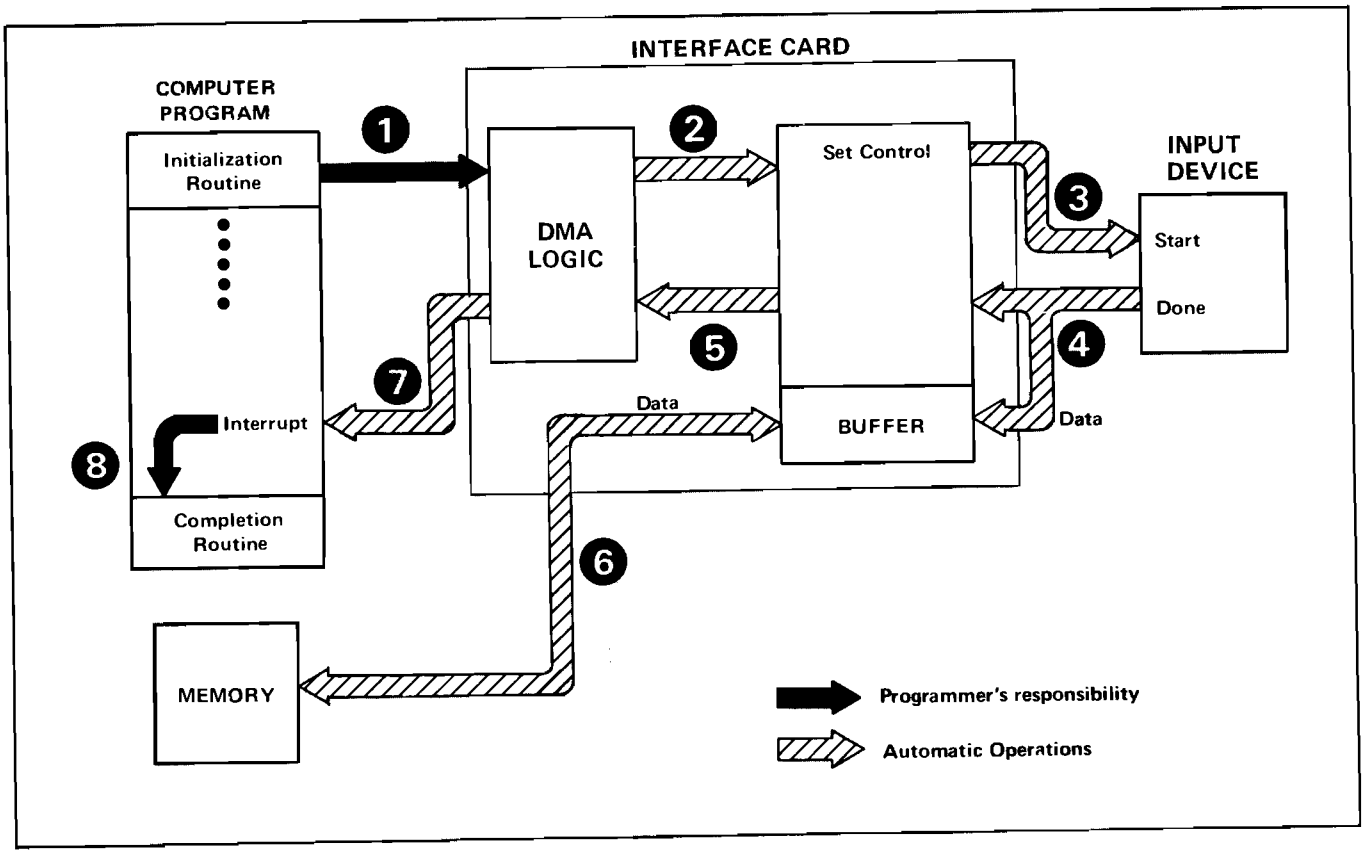


Figure 5-6. DMA Input Data Transfer

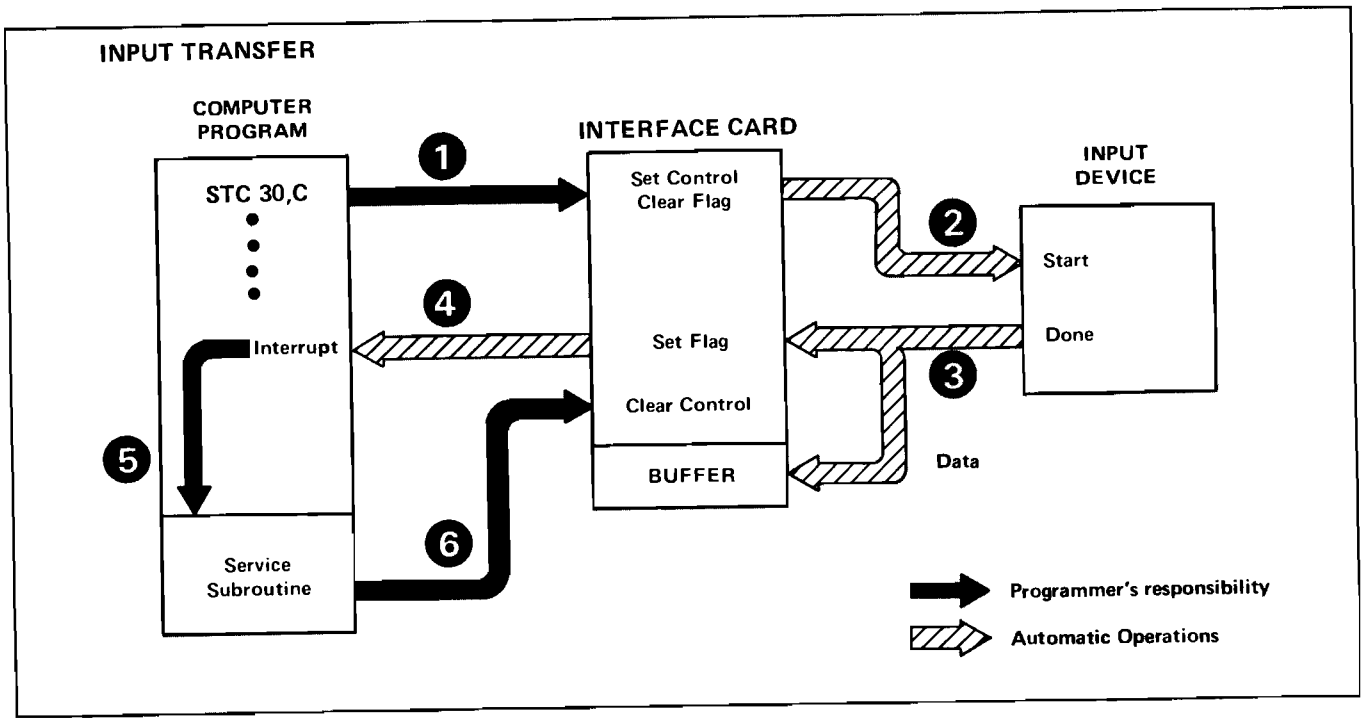


Figure 5-7. Input Data Transfer (Interrupt Method)

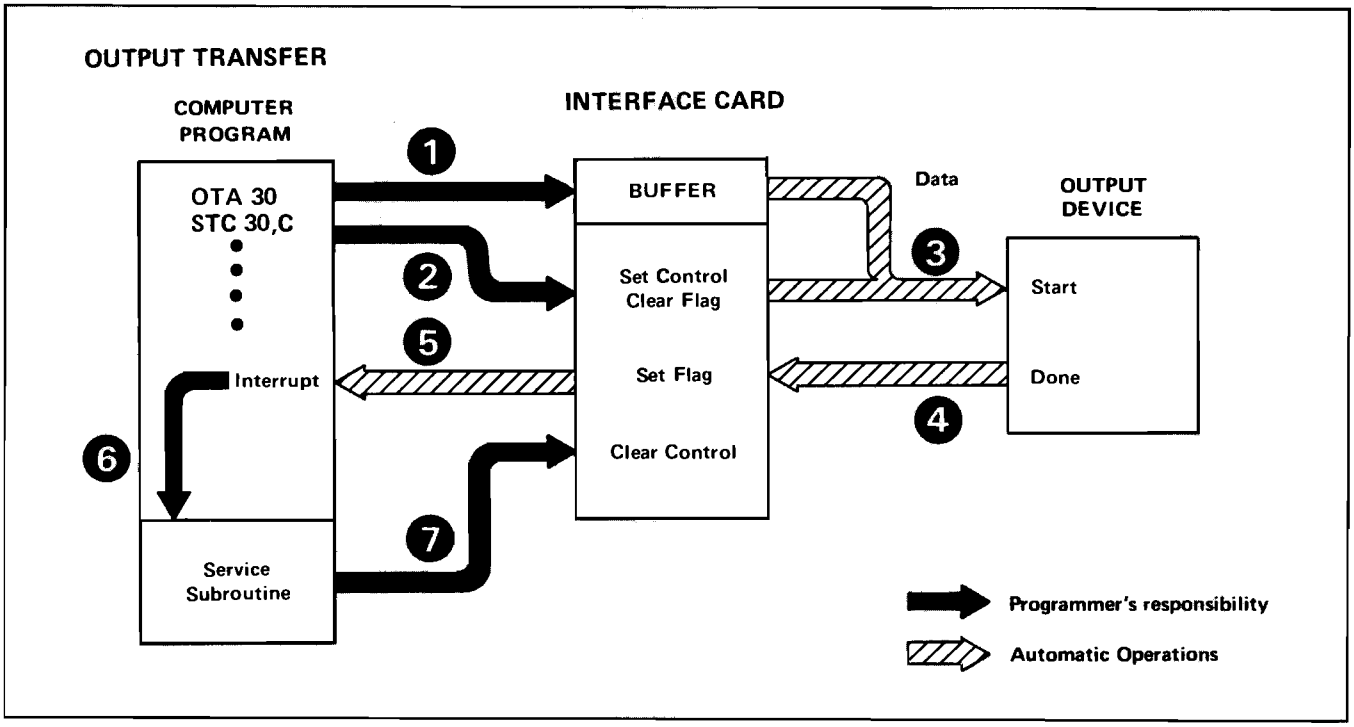


Figure 5-8. Output Data Transfer (Interrupt Method)

The device Done signal sets the card's Flag bit, which in turn generates an interrupt (5) provided that the interrupt system is on, priority is high, and the Control bit is set (done in step 2). The interrupt causes the current computer program to be suspended and control is transferred to a service subroutine (6). It is the programmer's responsibility to provide the linkage between the interrupt location (which agrees with the select code) and the service subroutine. The detailed contents of the subroutine are also the programmer's responsibility and the contents will vary with the type of device.

The subroutine may then output further data to the I/O card and reissue the STC 30,C command for additional data character transfers. One of the final instructions in the service subroutine must be a clear control (CLC 30,C). This step (7) allows lower priority devices to interrupt and restores the I/O card to its static "reset" condition (Control clear and Flag clear). At the end of the subroutine, control is returned to the interrupted program via the previously established linkages.

5-19. NON-INTERRUPT DATA TRANSFER

It is also possible to transfer data without using the interrupt system. This involves a "wait-for-flag" method in which the computer commands the device to operate and then waits for the completion response. In using this method to transfer data, computer time is relatively unimportant. It is assumed that the interrupt system is turned off (STF 00 not previously given). It is also assumed that the global register has been loaded and enabled and that the I/O card's control register has been loaded. As shown in table 5-1, the programming is very simple; each of the routines will transfer one word or character of data.

5-20. INPUT. As described in paragraph 5-17, an STC 30,C instruction begins the operation by commanding the device to read one word or character. The computer then goes into a waiting loop, repeatedly checking the status of the Flag bit (Flag 30). If the Flag bit is not set, the JMP *−1 instruction causes a jump back to the SFS instruction. (The *−1 operand is assembler notation for "this location minus one.") When the Flag bit is set, the skip condition for SFS is met and the JMP instruction is skipped. The computer thus exits from the waiting loop and the LIA 30 instruction loads the device input data into the A-register.

5-21. OUTPUT. The first step, which transfers the data to the I/O card buffer, is the OTA 30 instruction. Then STC 30,C commands the device to operate and accept the data. The computer then goes into a waiting loop as described in the preceding paragraph. When the Flag bit becomes set, indicating that the device has accepted the output data, the computer exits from the loop. (The final NOP is for illustration purposes only.)

5-22. TRANSFERS WITHOUT USING GLOBAL REGISTER

Non-DMA data transfers can be made without using the global register if the I/O card's control register does not have to be reloaded. The global register must be disabled and the I/O instructions must be addressed to the select code of the I/O card. For example, assuming that the card's select code is 25, the input routine shown in table 5-1 would be:

```
STC 25,C
SFS 25
JMP *-1
LIA 25
```

Table 5-1. Noninterrupt Transfer Routines

INSTRUCTIONS	COMMENTS
INPUT ROUTINE	
STC 30,C	Start device
SFS 30	Is input ready?
JMP *-1	No, repeat previous instruction
LIA 30	Yes, load input into A-register
	OUTPUT ROUTINE
OTB 30	Output data to I/O card's data register
STC 30,C	Start device
SFS 30	Has device accepted the data?
JMP *-1	No, repeat previous instruction
NOP	Yes, proceed

5-23. DIAGNOSE MODES

A diagnose mode allows the I/O cards to be accessed for diagnostic or test purposes. A diagnose mode is established when an OTA/B 2 instruction (output to the global register) is executed with the A- or B-register value equal to one through seven. (The diagnose mode is terminated when an OTA/B 2 instruction is executed with the A- or B-register equal to zero.) When establishing a diagnose mode the current contents of the global register (GR) is not altered. The diagnose mode can be on an individual I/O card or on all I/O cards. If the GR is disabled then all I/O cards accept the diagnose mode. If the GR is enabled, only the I/O card whose select code is in the GR will accept the diagnose mode. Diagnose Mode 7 is used to disable any service request (SRQ) signal coming into the I/O chip which may cause DMA to cycle during a test. (Mode 7 can be disabled only by a CRS signal (CLC 0).) Diagnose Modes 4 through 6 are reserved for future definition. Diagnose Modes 1 through 3 are described in the following paragraphs.

5-24. DIAGNOSE MODE 1

When an OTA/B 2 instruction is executed with the A- or B-register equal to one each I/O card responds by turning off priority to the next I/O card. When the instruction is complete the only I/O card receiving priority will be the highest priority I/O card (i.e., the one directly under the processor card. When a subsequent LIA/B 2 instruction is executed, the I/O card receiving priority sets the A- or B-register equal to its select code and identification data (ID) and passes priority to the next I/O card. Having responded once it will not respond again unless Mode 1 is established again. The next LIA/B 2 executed sets the A- or B-register equal to the second I/O card's select code and ID. The second I/O card at completion of the instruction passes priority to the next I/O card. This process continues until the last I/O card responds. After the last I/O card responds the next LIA/B 2 will not affect the A- or B-register and therefore can be detected as a no response. (An OTA/B 2 with the A- or B-register equal to 0 terminates this sequence.)

Mode 1 can also be used to retrieve the select code and ID of a desired I/O card without going through the priority process. This is accomplished by establishing Mode 1 and then executing an LIA/B xx, where xx is the I/O card select code. This procedure will not modify a priority sequence already in process. The Mode 1 select code and ID format is shown in table 5-2.

5-25. DIAGNOSE MODE 2

Diagnose Mode 2 causes an I/O card to respond to an LIA/B 2 instruction in the same manner as in Mode 1 except that the data set into the A- or B-register is as shown in table 5-3.

Table 5-2. Diagnose Mode 1

A/B BITS	MEANING
15	Intelligent interface
14, 13, 12, 11, 10, 9	Interface card type identification number
8, 7, 6	Interface card revision code
5, 0	Interface card select code

5-26. DIAGNOSE MODE 3

Diagnose Mode 3 allows an I/O chip to do a DMA transfer without affecting the I/O card. When Mode 3 is entered the I/O chip does a DMA input transfer of the data in the configuration address register to the location in memory pointed to by the DMA address register. The configuration address register is incremented after each transfer so that the data can be verified. The transfer continues until the DMA count is incremented to zero. Mode 3 also prevents any STC instructions from generating a device command to the I/O card.

Table 5-3. Diagnose Mode 2

A/B BITS	MEANING
15, 14, 13	Undefined
12	1 = Break feature is enabled
11	1 = Receiving interrupt priority
10	—
9	Control bit
8	Flag bit
7	1 = Global register equals select code of interface card
6	Global register enabled/disabled
5, 4, 3, 2, 1, 0	Current global register value

APPENDIX A

OCTAL ARITHMETIC

ADDITION

TABLE

0	01	02	03	04	05	06	07
1	02	03	04	05	06	07	10
2	03	04	05	06	07	10	11
3	04	05	06	07	10	11	12
4	05	06	07	10	11	12	13
5	06	07	10	11	12	13	14
6	07	10	11	12	13	14	15
7	10	11	12	13	14	15	16

EXAMPLE

Add:

```
  3677   octal
+ 1331   octal
 (111-)  carries
 5230    octal
```

MULTIPLICATION

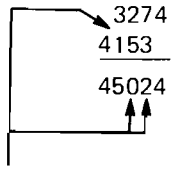
TABLE

1	02	03	04	05	06	07
2	04	06	10	12	14	16
3	06	11	14	17	22	25
4	10	14	20	24	30	34
5	12	17	24	31	36	43
6	14	22	30	36	44	52
7	16	25	34	43	52	61

EXAMPLE

Multiply:

```
  657   octal
X  54   octal
 3274
 4153
45024   octal
```



(Reminder: add in octal)

COMPLEMENT

To find the two's complement form of an octal number. (Same procedure whether converting from positive to negative or negative to positive.)

RULE

1. Subtract from the maximum representable octal value.
2. Add one.

EXAMPLE

Two's complement of 556_8:

```
 177777
-000556
 177221
    + 1
 177222
```

OCTAL/DECIMAL CONVERSIONS

OCTAL TO DECIMAL

TABLE

OCTAL	DECIMAL
0- 7	0- 7
10-17	8-15
20-27	16-23
30-37	24-31
40-47	32-39
50-57	40-47
60-67	48-55
70-77	56-63
100	64
200	128
400	256
1000	512
2000	1024
4000	2048
10000	4096
20000	8192
40000	16384
77777	32767

EXAMPLE

Convert 463_8 to a decimal integer.

$400_8 = 256_{10}$

$60_8 = 48_{10}$

$3_8 = 3_{10}$

307 decimal

DECIMAL TO OCTAL

TABLE

DECIMAL	OCTAL
1	1
10	12
20	24
40	50
100	144
200	310
500	764
1000	1750
2000	3720
5000	11610
10000	23420
20000	47040
32767	77777

EXAMPLE

Convert 5229_{10} to an octal integer.

$5000_{10} = 11610_8$

$200_{10} = 310_8$

$20_{10} = 24_8$

$9_{10} = 11_8$

12155_8

(Reminder: add in octal)

NEGATIVE DECIMAL TO TWO'S COMPLEMENT OCTAL

TABLE

DECIMAL	2's COMP
-1	177777
-10	177766
-20	177754
-40	177730
-100	177634
-200	177470
-500	177014
-1000	176030
-2000	174060
-5000	166170
-10000	154360
-20000	130740
-32768	100000

EXAMPLE

Convert -629_{10} to two's complement octal.

$-500_{10} = 177014_8$

$-100_{10} = 177634_8$

$-20_{10} = 177754_8$ (Add in octal)

$-9_{10} = 177767_8$

176613_8

For reverse conversion (two's complement octal to negative decimal):

1. Complement, using procedure on facing page.
2. Convert to decimal, using OCTAL TO DECIMAL table.

MATHEMATICAL EQUIVALENTS

$2 \pm n$ IN DECIMAL

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.0625
32	5	0.03125
64	6	0.01562 5
128	7	0.00781 25
256	8	0.00390 625
512	9	0.00195 3125
1 024	10	0.00097 65625
2 048	11	0.00048 82812 5
4 096	12	0.00024 41406 25
8 192	13	0.00012 20703 125
16 384	14	0.00006 10351 5625
32 768	15	0.00003 05175 78125
65 536	16	0.00001 52587 89062 5
131 072	17	0.00000 76293 94531 25
262 144	18	0.00000 38146 97265 625
524 288	19	0.00000 19073 48632 8125
1 048 576	20	0.00000 09536 74316 40625
2 097 152	21	0.00000 04768 37158 20312 5
4 194 304	22	0.00000 02384 18579 10156 25
8 388 608	23	0.00000 01192 09289 55078 125
16 777 216	24	0.00000 00596 04644 77539 0625
33 554 432	25	0.00000 00298 02322 38769 53125
67 108 864	26	0.00000 00149 01161 19384 76562 5
134 217 728	27	0.00000 00074 50580 59692 38281 25
268 435 456	28	0.00000 00037 25290 29846 19140 625
536 870 912	29	0.00000 00018 62645 14923 09570 3125
1 073 741 824	30	0.00000 00009 31322 57461 54785 15625
2 147 483 648	31	0.00000 00004 65661 28730 77392 57812 5
4 294 967 296	32	0.00000 00002 32830 64365 38696 28906 25

$10 \pm n$ IN OCTAL

10^n	n	10^{-n}
1	0	1.000 000 000 000 000 000 00
12	1	0.063 146 314 631 463 146 31
144	2	0.005 075 341 217 270 243 66
1 750	3	0.000 406 111 564 570 651 77
23 420	4	0.000 032 155 613 530 704 15
303 240	5	0.000 002 476 132 610 706 64
3 641 100	6	0.000 000 206 157 364 055 37
46 113 200	7	0.000 000 015 327 745 152 75
575 360 400	8	0.000 000 001 257 143 561 06
7 346 545 000	9	0.000 000 000 104 560 276 41
112 402 762 000	10	0.000 000 000 006 676 337 66
1 351 035 564 000	11	0.000 000 000 000 537 657 77
16 432 451 210 000	12	0.000 000 000 000 043 136 32
221 411 634 520 000	13	0.000 000 000 000 003 411 35
2 657 142 036 440 000	14	0.000 000 000 000 000 264 11
34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
434 157 115 760 200 000	16	0.000 000 000 000 000 001 63
5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01

MATHEMATICAL EQUIVALENTS

2^x IN DECIMAL

x	2^x	x	2^x	x	2^x
0.001	1.00069 33874 62581	0.01	1.00695 55500 56719	0.1	1.07177 34625 36293
0.002	1.00138 72557 11335	0.02	1.01395 94797 90029	0.2	1.14869 83549 97035
0.003	1.00208 16050 79633	0.03	1.02101 21257 07193	0.3	1.23114 44133 44916
0.004	1.00277 64359 01078	0.04	1.02811 38266 56067	0.4	1.31950 79107 72894
0.005	1.00347 17485 09503	0.05	1.03526 49238 41377	0.5	1.41421 35623 73095
0.006	1.00416 75432 38973	0.06	1.04246 57608 41121	0.6	1.51571 65665 10398
0.007	1.00486 38204 23785	0.07	1.04971 66836 23067	0.7	1.62450 47927 12471
0.008	1.00556 05803 98468	0.08	1.05701 80405 61380	0.8	1.74110 11265 92248
0.009	1.00625 78234 97782	0.09	1.06437 01824 53360	0.9	1.86606 59830 73615

$n \log_{10} 2$, $n \log_2 10$ IN DECIMAL

n	$n \log_{10} 2$	$n \log_2 10$	n	$n \log_{10} 2$	$n \log_2 10$
1	0.30102 99957	3.32192 80949	6	1.80617 99740	19.93156 85693
2	0.60205 99913	6.64385 61898	7	2.10720 99696	23.25349 66642
3	0.90308 99870	9.96578 42847	8	2.40823 99653	26.57542 47591
4	1.20411 99827	13.28771 23795	9	2.70926 99610	29.89735 28540
5	1.50514 99783	16.60964 04744	10	3.01029 99566	33.21928 09489

MATHEMATICAL CONSTANTS IN OCTAL SCALE

$\pi = (3.11037\ 552421)_{(8)}$

$\pi^{-1} = (0.24276\ 301556)_{(8)}$

$\sqrt{\pi} = (1.61337\ 611067)_{(8)}$

$\ln \pi = (1.11206\ 404435)_{(8)}$

$\log_2 \pi = (1.51544\ 163223)_{(8)}$

$\sqrt{10} = (3.12305\ 407267)_{(8)}$

$e = (2.55760\ 521305)_{(8)}$

$e^{-1} = (0.27426\ 530661)_{(8)}$

$\sqrt{e} = (1.51411\ 230704)_{(8)}$

$\log_{10} e = (0.33626\ 754251)_{(8)}$

$\log_2 e = (1.34252\ 166245)_{(8)}$

$\log_2 10 = (3.24464\ 741136)_{(8)}$

$\gamma = (0.44742\ 147707)_{(8)}$

$\ln \gamma = -(0.43127\ 233602)_{(8)}$

$\log_2 \gamma = -(0.62573\ 030645)_{(8)}$

$\sqrt{2} = (1.32404\ 746320)_{(8)}$

$\ln 2 = (0.54271\ 027760)_{(8)}$

$\ln 10 = (2.23273\ 067355)_{(8)}$

OCTAL COMBINING TABLES

MEMORY REFERENCE INSTRUCTIONS

Indirect Addressing

Refer to octal instruction codes given on the following page.
To combine code for indirect addressing, merge "100000" with octal instruction code.

REGISTER REFERENCE INSTRUCTIONS

Shift-Rotate Group (SRG)

1. select to operate on A or B
2. select 1 to 4 micros, not more than one from each column.
3. combine octal codes (leading zeros omitted) by inclusive or.
4. order of execution is from column 1 to column 4.

A Operations

1	2	3	4
ALS (1000)	CLE (40)	SLA (10)	ALS (20)
ARS (1100)			ARS (21)
RAL (1200)			RAL (22)
RAR (1300)			RAR (23)
ALR (1400)			ALR (24)
ERA (1500)			ERA (25)
ELA (1600)			ELA (26)
ALF (1700)			ALF (27)

B Operations

1	2	3	4
BLS (5000)	CLE (4040)	SLB (4010)	BLS (4020)
BRS (5100)			BRS (4021)
RBL (5200)			RBL (4022)
RBR (5300)			RBR (4023)
BLR (5400)			BLR (4024)
ERB (5500)			ERB (4025)
ELB (5600)			ELB (4026)
BLF (5700)			BLF (4027)

Alter-Skip Group (ASG)

1. select to operate on A or B.
2. select 1 to 8 micros, not more than one from each column.
3. combine octal codes (leading zeros omitted) by inclusive or.
4. order of execution is from column 1 to column 8.

A Operations

1	2	3	4
CLA (2400)	SEZ (2040)	CLE (2100)	SSA (2020)
CMA (3000)		CME (2200)	
CCA (3400)		CCE (2300)	

5	6	7	8
SLA (2010)	INA (6040)	CLE (6100)	RSS (2001)

B Operations

1	2	3	4
CLB (6400)	SEZ (6040)	CLE (6100)	SSB (6020)
CMB (7000)		CME (6200)	
CCB (7400)		CCE (6300)	

5	6	7	8
SLB (6010)	INB (6004)	SZB (6002)	RSS (6001)

INPUT/OUTPUT INSTRUCTIONS

Clear Flag

Refer to octal instruction codes given on the following page.
To clear flag after execution (instead of holding flag), merge "001000" with octal instruction code.

INSTRUCTION CODES IN OCTAL

Memory Reference

AND	01 (0XX) ---
XOR	02 (0XX) ---
IOR	03 (0XX) ---
JSB	01 (1XX) ---
JMP	02 (1XX) ---
ISZ	03 (1XX) ---
ADA	04 (0XX) ---
ADB	04 (1XX) ---
CPA	05 (0XX) ---
CPB	05 (1XX) ---
LDA	06 (0XX) ---
LDB	06 (1XX) ---
STA	07 (0XX) ---
STB	07 (1XX) ---

↑ Binary

Shift-Rotate

NOP	000000
CLE	000040
SLA	000010
SLB	004010
ALS	001000
BLS	005000
ARS	001100
BRS	005100
RAL	001200
RBL	005200
RAR	001300
RBR	005300
ALR	001400
BLR	005400
ERA	001500
ERB	005500
ELA	001600
ELB	005600
ALF	001700
BLF	005700

Alter-Skip

CLA	002400
CLB	006400
CMA	003000
CMB	007000
CCA	003400
CCB	007400
SEZ	002040
CLE	002100
CME	002200
CCE	002300
SSA	002020
SSB	006020
SLA	002010
SLB	006010
INA	002004
INB	006004
SZA	002002
SZB	006002
RSS	002001

Input/Output

HLT	1020 --
STF	1021 --
CLF	1031 --
SFC	1022 --
SFS	1023 --
MIA	1024 --
MIB	1064 --
LIA	1025 --
LIB	1065 --
OTA	1026 --
OTB	1066 --
STC	1027 --
CLC	1067 --
STO	102101
CLO	103101
SOC	102201
SOS	102301

Extended Arithmetic

MPY	100200
DIV	100400
DLD	104200
DST	104400
ASR	1010 (01X) -
ASL	1000 (01X) -
LSR	1010 (10X) -
LSL	1000 (10X) -
RRR	1011 (00X) -
RRL	1001 (00X) -

↑ Binary

*Assuming: no indirect addressing
no combined microinstructions
shifts taken in first position only
hold flag after I/O execution

Refer to preceding page for octal combining tables

BASE SET INSTRUCTION CODES IN BINARY

15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
D/I	AND 001	0 Z/C	← Memory Address →		
D/I	XOR 010	0 Z/C			
D/I	IOR 011	0 Z/C			
D/I	JSB 001	1 Z/C			
D/I	JMP 010	1 Z/C			
D/I	ISZ 011	1 Z/C			
D/I	AD* 100	A/B Z/C			
D/I	CP* 101	A/B Z/C			
D/I	LD* 110	A/B Z/C			
D/I	ST* 111	A/B Z/C			

15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
0	SRG 000	A/B 0 D/E	*LS 000	†CLE D/E ‡SL*	*LS 000
		A/B 0 D/E	*RS 001	D/E	*RS 001
		A/B 0 D/E	R*L 010	D/E	R*L 010
		A/B 0 D/E	R*R 011	D/E	R*R 011
		A/B 0 D/E	*LR 100	D/E	*LR 100
		A/B 0 D/E	ER* 101	D/E	ER* 101
		A/B 0 D/E	EL* 110	D/E	EL* 110
		A/B 0 D/E	*LF 111	D/E	*LF 111
		NOP 000	000	000	000

15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
0	ASG 000	A/B 1	CL* 01	CLE 01	SEZ SS* SL*	IN* SZ* RSS
		A/B	CM* 10	CME 10	
		A/B	CC* 11	CCE 11	

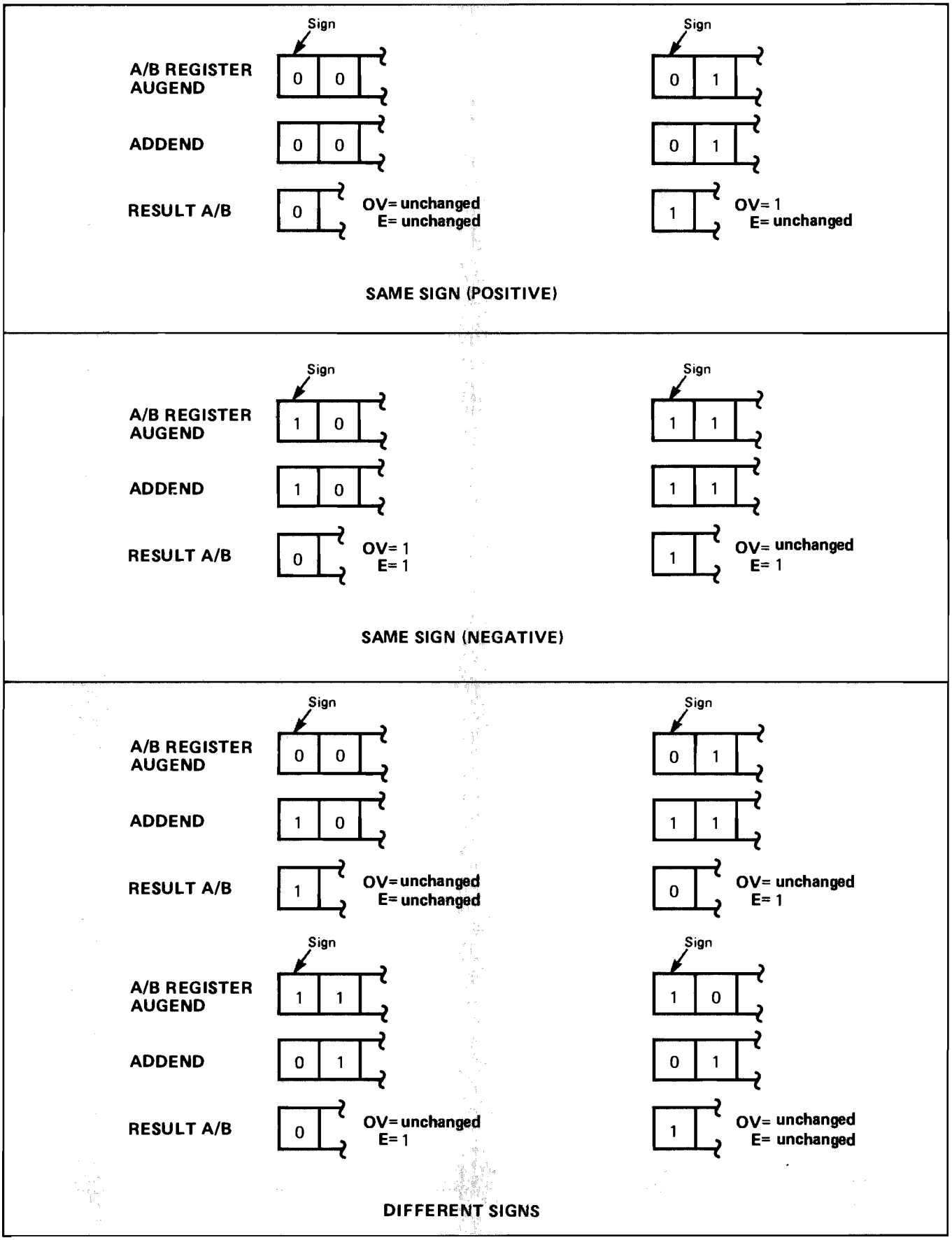
15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
1	IOG 000	1 H/C	HLT 000	← Select Code →	
		1 0	STF 001		
		1 1	CLF 001		
		1 0	SFC 010		
		1 0	SFS 011		
		A/B 1 H/C	MI* 100		
		A/B 1 H/C	LI* 101		
		A/B 1 H/C	OT* 110		
		0 1 H/C	STC 111		
		1 1 H/C	CLC 111		
		1 0	STO 001	000	001
		1 1	CLO 001	000	001
		1 H/C	SOC 010	000	001
		1 H/C	SOS 011	000	001

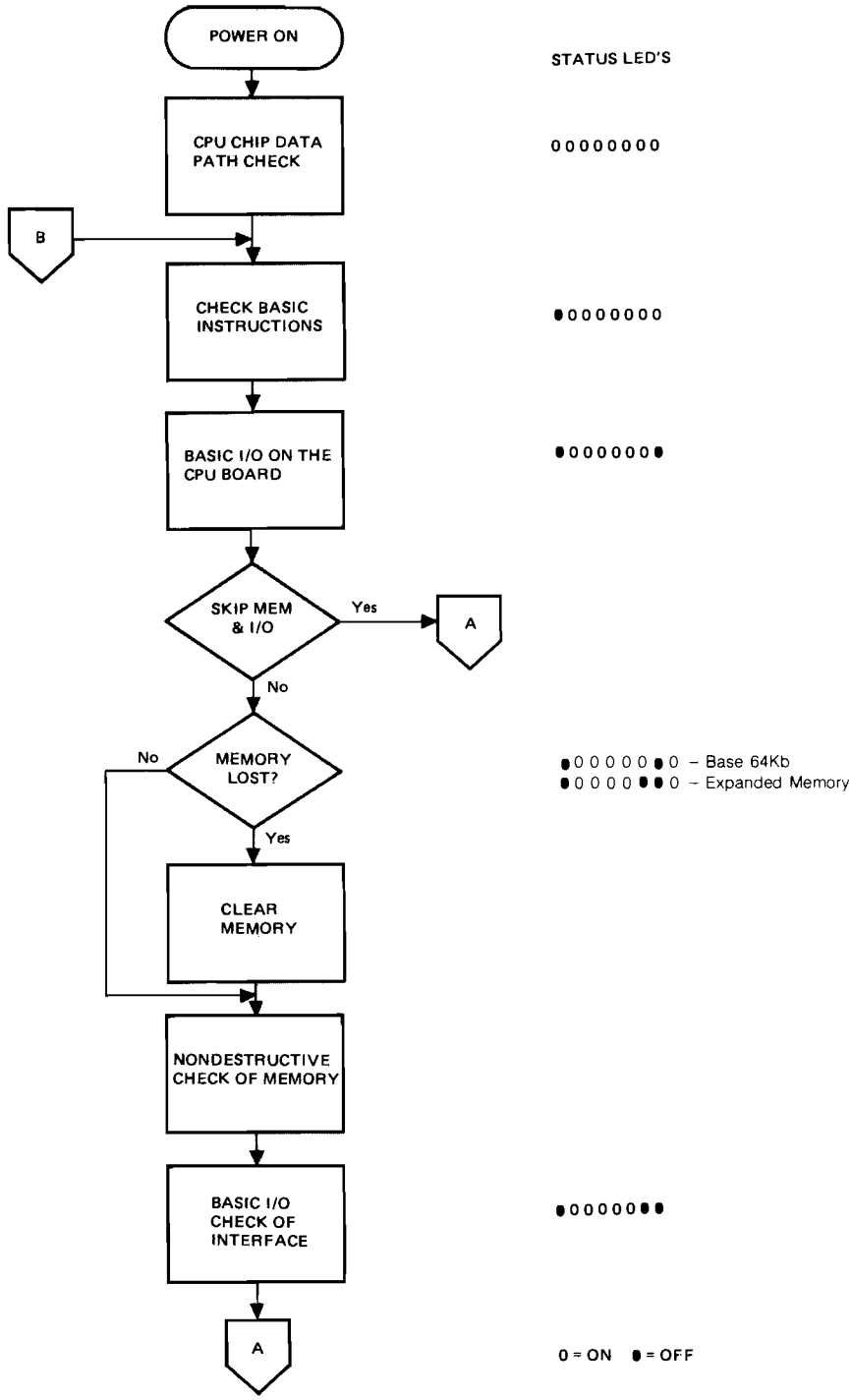
15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
1	EAG 000	MPY** 000	010	000	000
		DIV** 000	100	000	000
		DLD** 100	010	000	000
		DST** 100	100	000	000
		ASR 001	000	0 1	
		ASL 000	000	0 1	
		LSR 001	000	1 0	← number of bits →
		LSL 000	000	1 0	
		RRR 001	001	0 0	
		RRL 000	001	0 0	

Notes: * = A or B, according to bit 11.
D/I, A/B, Z/C, D/E, H/C coded: 0/1.
**Second word is Memory Address.

†CLE: Only this bit is required.
‡SL*: Only this bit and bit 11 (A/B as applicable) are required.

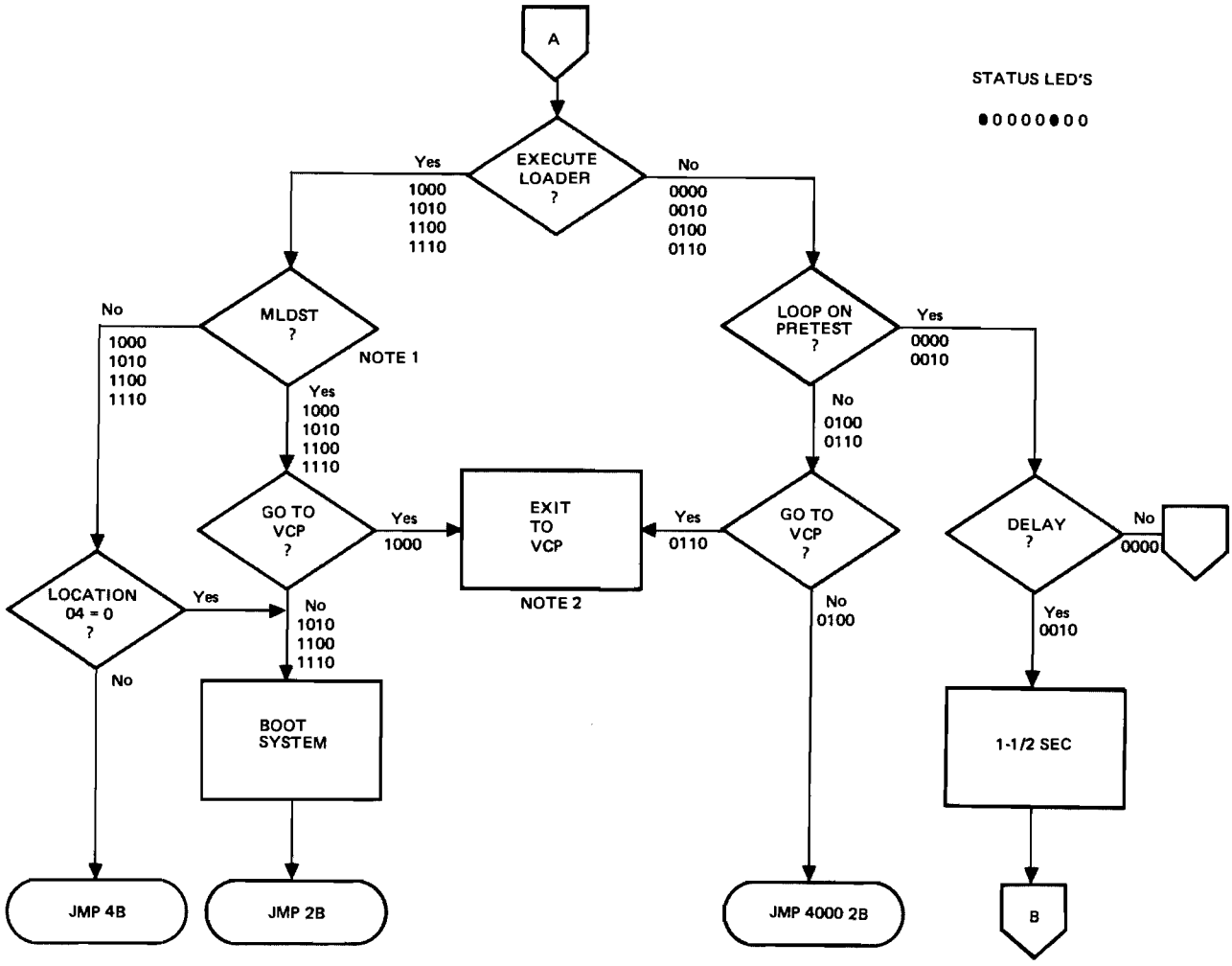
EXTEND AND OVERFLOW EXAMPLES





7700-478-1(1)

Power Up Sequence (Sheet 1 of 2)



Switch 1 2 3 4	Action
0 0 0 0	Loop on self-test regardless of error
0 0 1 0	Loop on self-test and stop on error
0 1 0 0	Start ROM program at location 4 0 0 0 2B
0 1 1 0	Go to VCP on completion of processor test (skips memory test and I/O tests) (MEMLOST) (MEMSAVED)
1 0 0 0	Go to VCP otherwise restart program
1 0 1 0	load from PROM
1 1 0 0	load from network (DS)
1 1 1 0	load from disc

1 = open = up
0 = closed = down

POWER UP SEQUENCE FLOWCHART

NOTES:

1. If a parity error occurs during the memory portion of the self-test, memory is considered lost.
2. When the VCP program is entered, it prints P, A, B, M, and T on the VCP. This information includes the following:
   a. If the P-register is other than 77777 or 37777, a parity error(s) occurred during the self-test and the P-register is set to the address of the error (first error).
   b. The A-register contains 0000xx, where xx is the number of I/O chips that were tested during self test; in addition, the A-register for a computer with expanded memory contains 0nnnxx, where nnn is the page number of the highest page of memory tested.
   c. The B-register contains the revision level of the ROM on the processor card.

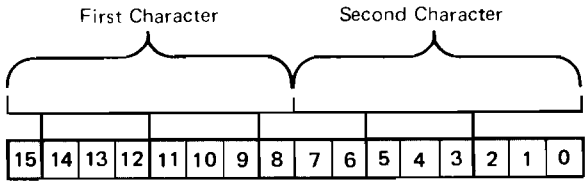
7700-478-2(1)

Power Up Sequence (Sheet 2 of 2)

CHARACTER CODES

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
A	040400	000101
B	041000	000102
C	041400	000103
D	042000	000104
E	042400	000105
F	043000	000106
G	043400	000107
H	044000	000110
I	044400	000111
J	045000	000112
K	045400	000113
L	046000	000114
M	046400	000115
N	047000	000116
O	047400	000117
P	050000	000120
Q	050400	000121
R	051000	000122
S	051400	000123
T	052000	000124
U	052400	000125
V	053000	000126
W	053400	000127
X	054000	000130
Y	054400	000131
Z	055000	000132
a	060400	000141
b	061000	000142
c	061400	000143
d	062000	000144
e	062400	000145
f	063000	000146
g	063400	000147
h	064000	000150
i	064400	000151
j	065000	000152
k	065400	000153
l	066000	000154
m	066400	000155
n	067000	000156
o	067400	000157
p	070000	000160
q	070400	000161
r	071000	000162
s	071400	000163
t	072000	000164
u	072400	000165
v	073000	000166
w	073400	000167
x	074000	000170
y	074400	000171
z	075000	000172
0	030000	000060
1	030400	000061
2	031000	000062
3	031400	000063
4	032000	000064
5	032400	000065
6	033000	000066
7	033400	000067
8	034000	000070
9	034400	000071
NUL	000000	000000
SOH	000400	000001
STX	001000	000002
ETX	001400	000003
EOT	002000	000004
ENQ	002400	000005

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
ACK	003000	000006
BEL	003400	000007
BS	004000	000010
HT	004400	000011
LF	005000	000012
VT	005400	000013
FF	006000	000014
CR	006400	000015
SO	007000	000016
SI	007400	000017
DLE	010000	000020
DC1	010400	000021
DC2	011000	000022
DC3	011400	000023
DC4	012000	000024
NAK	012400	000025
SYN	013000	000026
ETB	013400	000027
CAN	014000	000030
EM	014400	000031
SUB	015000	000032
ESC	015400	000033
FS	016000	000034
GS	016400	000035
RS	017000	000036
US	017400	000037
SPACE	020000	000040
!	020400	000041
"	021000	000042
#	021400	000043
$	022000	000044
%	022400	000045
&	023000	000046
'	023400	000047
(	024000	000050
)	024400	000051
*	025000	000052
+	025400	000053
,	026000	000054
-	026400	000055
.	027000	000056
/	027400	000057
:	035000	000072
;	035400	000073
<	036000	000074
=	036400	000075
>	037000	000076
?	037400	000077
@	040000	000100
[	055400	000133
\	056000	000134
]	056400	000135
Δ	057000	000136
_	057400	000137
`	060000	000140
{	075400	000173
|	076000	000174
}	076400	000175
~	077000	000176
DEL	077400	000177



NOTES

NOTES