SPC-16/40, SPC-16/60, SPC-16/80
SPC-16/45, SPC-16/65, SPC-16/85
maintenance manual

88A00234A-C

SPC-16/40, SPC-16/45,
SPC-16/60, SPC-16/65,
SPC-16/80 SPC-16/85
maintenance

GENERAL AUTOMATION, INC.
1055 East Street
Anaheim, California 92805
(714) 778-4800

REVISION

Symbol	Description	Manager Publications	Approved	Date
A	Original Issue			Oct. 73
B	Added RCSR/RCSM Instructions.			Mar. 74
C	Major Revision			Aug. 74

SUPPORTING DOCUMENTS AND PROGRAMS

Title	Number
SPC-16/40+ System Reference Manual	88A00243A
128K Extended Memory Option Technical Manual	88A00432A
SPC-16 Processor T&V Program	6T000
SPC-16 Memory Test Programs	6T500, 6T020
SPC-16 Teletype T&V Program	6T100
SPC-16 Processor and Memory T&V Program Manual	88A00184A
SPC-16 Teletype T&V Program Manual	88A00185A
SPC-16 Power Supply Function Test Specification	83S0036A

CONTENTS

Contents (Continued)

Contents (Continued)

Contents (Continued)

Contents (Continued)

Section Title Page

Figures

Contents (Continued)

Contents (Continued)

Tables

Contents (Continued)

PREFACE

This document is a maintenance manual for use with General Automation's SPC-16 Automation Computer models 40/45, 60/65, 80/85. The included material is intended to guide both the General Automation field service representative and the customer in the proper procedures of on-site fault isolation and correction and preventive maintenance.

All procedures are given in an easily followed step-by-step format. Fault isolation is to the processor board level.

Usefulness of this manual is enhanced by a previous reading of the SPC-16/40 series Reference Manual (88A00243A).

The SPC-16/40 series Maintenance Manual is organized as follows:

- o Section 1 includes a general description of the SPC-16/40 series system.
- o Section 2 presents drawings intended to give the reader a basic orientation as to the physical layout of the computer.
- o Section 3 describes the functional organization of the SPC-16 Central Processing Unit.
- o Section 4 describes data flow within the computer.
- o Section 5 describes the basic timing signals in the SPC-16 processor.
- o Section 6 presents sequence diagrams for the SPC-16 instruction set.
- o Section 7 describes the SPC-16 memory.
- o Section 8 describes the SPC-16 processor power supply.
- o Section 9 discusses preventive maintenance procedures.
- o Section 10 presents procedures to isolate an existing malfunction in the SPC-16 processor and Input/Output system.
- o Appendix A gives the definitions of signal mnemonics.
- o Appendix B is a guide to processor board interchangeability.
- o Appendix C includes lists of parts used in the computers mainframe assembly.
- o Appendix D is a description of the switches and indicators on the SPC-16 console.

SECTION 1
INTRODUCTION

The SPC-16 computer is a stored-program digital computer with 16-bit word length organization, a high-speed memory, and a versatile arithmetic and control unit.

1.1 SYSTEM ORGANIZATION

The SPC-16 processor family consists of six different computers offering a choice of three memory cycle speeds and two packaging configurations:

Model Number	Memory Cycle Time	Packaging Configuration
SPC-16/40	1440 Nanoseconds	Internal I/O
SPC-16/45	1440 Nanoseconds	External I/O
SPC-16/60	960 Nanoseconds	Internal I/O
SPC-16/65	960 Nanoseconds	External I/O
SPC-16/80	800 Nanoseconds	Internal I/O
SPC-16/85	800 Nanoseconds	External I/O

All models of the SPC-16 utilize the same efficient modular concept which enables the user to maintain, troubleshoot and remove and replace subassemblies with ease.

SPC-16/40, 60, 80

In the SPC-16/40, 60 and 80 models the I/O logic is housed in the computer mainframe. This configuration provides eight card slots for peripheral controller boards, four card slots for memory logic boards and four card slots for processor logic boards.

A memory slot will accept any of the following memory modules:

- A Read/Write Memory module with 4K, 8K or 16K word capacity.
- If 4K, 8K and/or 16K modules are intermixed, the 4K must be in highest address slot.
- A Read-Only Memory (ROM) module with 512-, 1024-, 2048- or 4096-word capacity.
- A Read/Write Memory module that also includes a 32- or 64-word ROM or a Memory Protect unit (option).

This series of computer utilizes the integral I/O packaging configuration, which is arranged as shown in Figure 1-1.

SPC-16/45, 65, 85

The SPC-16/45, 65 and 85 model computers are designed for users with applications requiring more than 32K of memory. These models provide optionally up to 128K of memory. In these models the I/O controller modules are housed in an external I/O enclosure, which allows the mainframe to accommodate the additional memory modules.

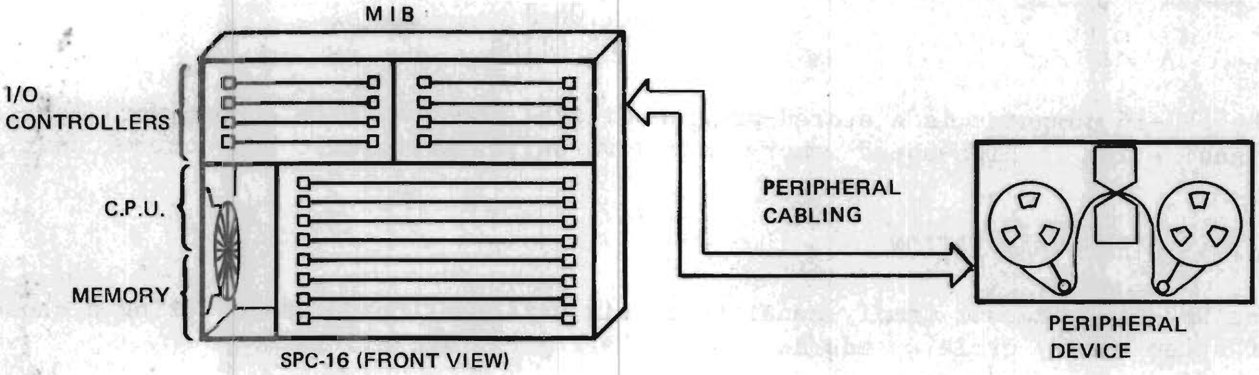


Figure 1-1. SPC-16 40/60/80 Typical System

As shown in Figure 1-2, the mainframe contains the four basic CPU boards and up to eight memory boards. The external I/O bus is cabled to the external I/O enclosure, which contains the controllers. The external I/O enclosure has 18 card slots available for peripheral controller cards and one slot reserved for a cable interface card.

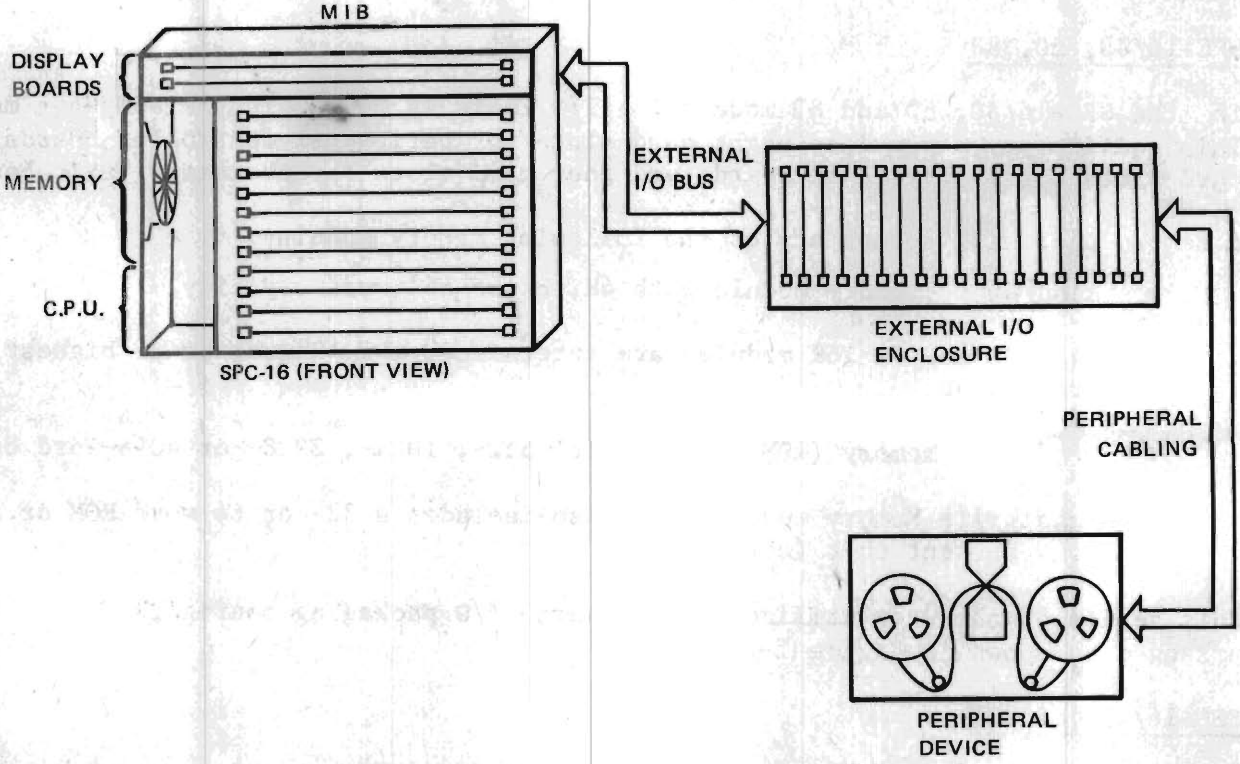


Figure 1-2. SPC-16/45/65/85 System with 64K Expanded Memory

1.2 GENERAL SPECIFICATIONS

General specifications for the SPC-16 40/45, 60/65 and 80/85 series are summarized in Table 1-1.

Table 1-1. SPC-16 Specification Summary

Characteristic	Specification
Memory	o Random access o 16-bit word organization. o Expandable to 32K 16-bit words (using 8K memory modules). o Extendable to 128K 16- bit words (Models 45, 65, 85 only). o Interchangeable Read/Write and Read Only Memory boards. o Read-Only Memory modules available in 512-, 1024-, 2048- and 4096- word increments. o Read/Write Memory module available in 4096- and 8192- word sizes. o Read/Write Memory module may include 32- or 64- word ROM. o Lithium ferrite core stacks (with wide temperature tolerance) used on Read/Write Memory; each stack has an independent current regulator. o Metal-oxide semiconductor ICs used on Read Only Memory boards.
Addressing	o Single and double-word addressing. o Eleven addressing schemes, including: Direct and direct indexed; indirect and indirect indexed; program relative and program relative indirect; base relative and base relative indexed; base relative indirect and base relative indirect indexed; literal.

Table 1-1. (continued)

Characteristic	Specification
Arithmetic	o Bit, byte and word operations (logical and arithmetic). o Parallel, binary, fixed point two's complement.
Instructions	o There are 9 standard instruction groups and one optional group (number of instructions in a group is shown in parentheses following group name): - Memory reference (4) - Memory reference, indexed (12) - Skip (±256 locations on 8 conditions) (8) - Register operate and compare (11) - Register operate literal and compare (11) - Register change (16) - Shift (0 to 16 bits) (4) - Control (9) - Input/Output for up to 64 devices (single word instruction addressing) (6) - Hardware Multiply and Divide (optional) (2) o All instructions may be stored in either Read/Write or Read Only Memory modules.
Instruction Execution Time	o Execution times vary with the complexity of the instruction and are a multiple of memory cycle time. Values below are execution times for a one-cycle instruction. o SPC-16/40 & 45 1440 ns (Read/Write Memory) 720 ns (Read Only Memory) o SPC-16/60 & 65 960 ns (Read/Write Memory) 480 ns (Read Only Memory) o SPC-16/80 & 85 800 ns (Read/Write Memory) 400 ns (Read Only Memory)

Table 1-1. (continued)

Characteristic	Specification
Input/Output	o 16-bit parallel bus. o Allows communication between SPC-16 registers/memory and up to 64 external peripheral units (via peripheral controllers). o Programmed I/O transfer rates, full 16 bits, for the three model pairs are, using back-to-back instructions: SPC-16 40/45 347.2 KHz SPC-16 60/65 520.8 KHz SPC-16 80/85 625.0 KHz o DMA port allows direct access of memory by peripheral unit(s) using a cycle-stealing technique. Transfer rates for the three model pairs are: SPC-16 40/45 694.4 KHz SPC-16 60/65 1041.6 KHz SPC-16 80/85 1250.0 KHz o Sixteen DMA channels are provided as standard. The controller physically nearest to the CPU has priority. o Sixteen memory locations are dedicated to the standard DMA channels.
System Safety Interrupts	o Power Fail Memory protection (standard for all models). o Relative Time Clock (standard for Models 40/60/80; optional for Models 45/65/85). o Operations Monitor Alarm (standard for Models 40/60/80; optional for Models 45/65/85). o Power Fail Interrupt (standard for Models 40/60/80; optional for Models 45/65/85).

Table 1-1. (continued)

Characteristic	Specification
Other Interrupts	o Auto Restart Interrupt (standard for Models 40/60/80; optional for Models 45/65/85). o Block Protect (optional on all models). o Teletype (optional for models 45/65/85) o Console o External (I/O)
Physical	Console Panel Programmer's console. Sixteen data switches and indicators. Register select switches for 19 registers and status indicators. Console lock-out switch, key operated. Operations monitor alarm. Console interrupt. General Dimensions SPC-16 40/60/80 and 45/65/85 without Memory Expansion options: Height = 10.5 inches (26.7 centimeters) Width = 19.0 inches (48.3 centimeters) Depth = 21.6 inches (54.9 centimeters) Weight = max. approx. 55 lbs. (25 kilograms) SPC-16 45/65/85 with 64K Memory Expansion option: Height = 12.2 inches (31.0 centimeters) Width = 19.0 inches (48.3 centimeters) Depth = 22.7 inches (57.6 centimeters) Weight = max. approx. 75 lbs. (34 kilograms) Environment Temperature = 0°C to +50°C (+32°F to +122°F) Humidity = up to 90% relative (no condensation)

Table 1-1. (continued)

Characteristic	Specification
	AC Power 115 vac, ±10%, 47 to 63 HZ, single phase 220 vac, ±10%, 47 to 63 HZ, single phase 700 watts (4K core), add 40 watts per additional 4K. DC Current Requirements See Tables 1-2 and 1-3 for current requirements for all boards in system. Site Preparation No special wiring, subflooring, air conditioning, or other site preparation is required.

Table 1-2. DC Current Requirements of CPU and Memory Boards

Board Type		+5V	+15V	-15V	+20V
MIO	CPU Boards	10.00A	0.60A	—	—
Arithmetic					
Macro					
Timing					
Console					
MIB					
Expanded MIB					
MLD Board (SPC-16/45/65/85 only)		0.25A	0.12A	—	—
4K Memory Board	Operating	0.92A	1.70A	1.25A	6.00A
8K Memory Board		0.90A	1.40A	1.40A	6.00A
4K Memory Board	Idle	0.68A	0.15A	0.30A	—
8K Memory Board		0.80A	0.20A	0.40A	—

Table 1-3. DC Current Requirements of General Automation Controller Boards

Model Number	No. of BDS	+5 VDC	+15 VDC	-15 VDC
1615 - 0016 HSMD	2	2.55A	-	-
1615 - 0203 CIT	1	1.18A	-	-
1615 - 0202 CID	1	.72A	-	-
1615 - 0209 MHSDC	1	1.43A	-	-
1615 - 0210 CCIF	2	2.17A	-	1.0A
1615 - 0215 GAARD	1 Δ	1.00A	.25A	.25A
1615 - 0216 GAARD	1 Δ	1.00A	.25A	.25A
1615 - 0220 HSAPU	3	Not Avail.	-	-
1615 - 0221 HSAPU	3	Not Avail.	-	-
3315 - 6200 CR	1	.75A	-	-
3316 - 6200 CR	1	.75A	-	-
3317 - 6200 CR	1	.75A	-	-
3318 - 6200 CR	1	.75A	-	-
3314 - 6200 CP	1	1.15A	-	-
3321 - 6200 PTR	1	.50A	-	-
3322 - 6200 PTP	1	.50A	-	-
3323 - 6200 PTR & PTP	1	.97A	-	-
3325 - 6200 PTR/PTP	1	.97A	-	-
3331 - 6200 MTU	3	3.76A	-	-
3332 - 6200 MTU	3	3.76A	-	-
3333 - 6200 MTU	3	3.76A	-	-
3334 - 6200 MTU	3	3.89A	-	-
3335 - 6200 MTU	3	3.89A	-	-
3336 - 6200 MTU	3	3.89A	-	-
3341 - 6200 DISK	3	4.46A	-	-
3342 - 6200 HPTSD	2	2.35A	-	-
3343 - 6200 DISK	3	4.46A	-	-
3346 - 6200 DISK	2	2.81A	-	-
3347 - 6200 DISK	2	2.81A	-	-
3349 - 6200 DISK	2	2.25A	-	-
3353 - 6200 LP	1	.75A	-	-
3354 - 6200 LP	1	.75A	-	-
3355 - 6200 LP & CR	1	1.50A	-	-
3356 - 6200 LP & CR	1	1.50A	-	-
3357 - 6200 LP & CR	1	1.50A	-	-
3358 - 6200 LP & CR	1	1.50A	-	-

Δ Occupies four (4) card slots.

SECTION 2
COMPONENT IDENTIFICATION

This section contains drawings of the external arrangement of the SPC-16 40/60/80 and 45/65/85 computers as an aid in locating components on these machines.

The SPC-16 40/60/80 processor can be ordered in three basic versions:

- Model 1640 (or 60 or 80) - 1159 is supplied with one 4K memory board and can accommodate up to three additional 4K memory boards.
- Model 1640 (or 60 or 80) - 1259 is supplied with one 8K memory board and can accommodate up to three additional memory boards. Any memory slot can accommodate either a 4K, 8K or 16K board, but only one 4K board can be used in any one - 1259 memory configuration.
- Model 1640 (or 60 or 80) - 1359 is identical to the -1259 above, except that its basic configuration consists of one 4K memory board.

NOTE

Due to their double thickness, 16K memory boards occupy two slots.

Figure 2-1 is an installation drawing of the SPC-16 40/60/80. Figure 2-2 shows the internal board arrangement of the SPC-16 40/60/80.

The SPC-16 45/65/85 processor can be ordered in three basic versions:

- Model 1645 (or 65 or 85) -1100 is supplied with one 4K memory board and can accommodate up to three additional 4K memory boards without purchasing additional options. Memory can be expanded to accommodate four additional 4K memory boards by installing the optional Internal Memory Expansion Board, 1645 (or 65 or 85) -0097.
- Model 1645 (or 65 or 85) -1200 is supplied with one 8K memory board and can accommodate up to three additional memory boards without purchasing additional options. Memory can be expanded to accommodate four additional memory boards by installing the optional Memory Expansion Chassis, 1645 (or 65 or 85) -0095. Any memory slot can accommodate either a 4K, 8K or 16K board, but only one 4K board can be used in any one -1200 memory configuration.
- Model 1645 (or 65 or 85) -1300 is identical to the -1200 above, except that its basic configuration consists of one 4K memory board.

Figure 2-3 is an installation drawing of the SPC-16 45/65/85 Model 16xx -1100. Figure 2-4 is an installation drawing of the SPC-16 45/65/85 Models 16xx -1200 and -1300 with the 16xx -0095 option installed. Figure 2-5 shows the internal board arrangement of the SPC-16 45/65/85.

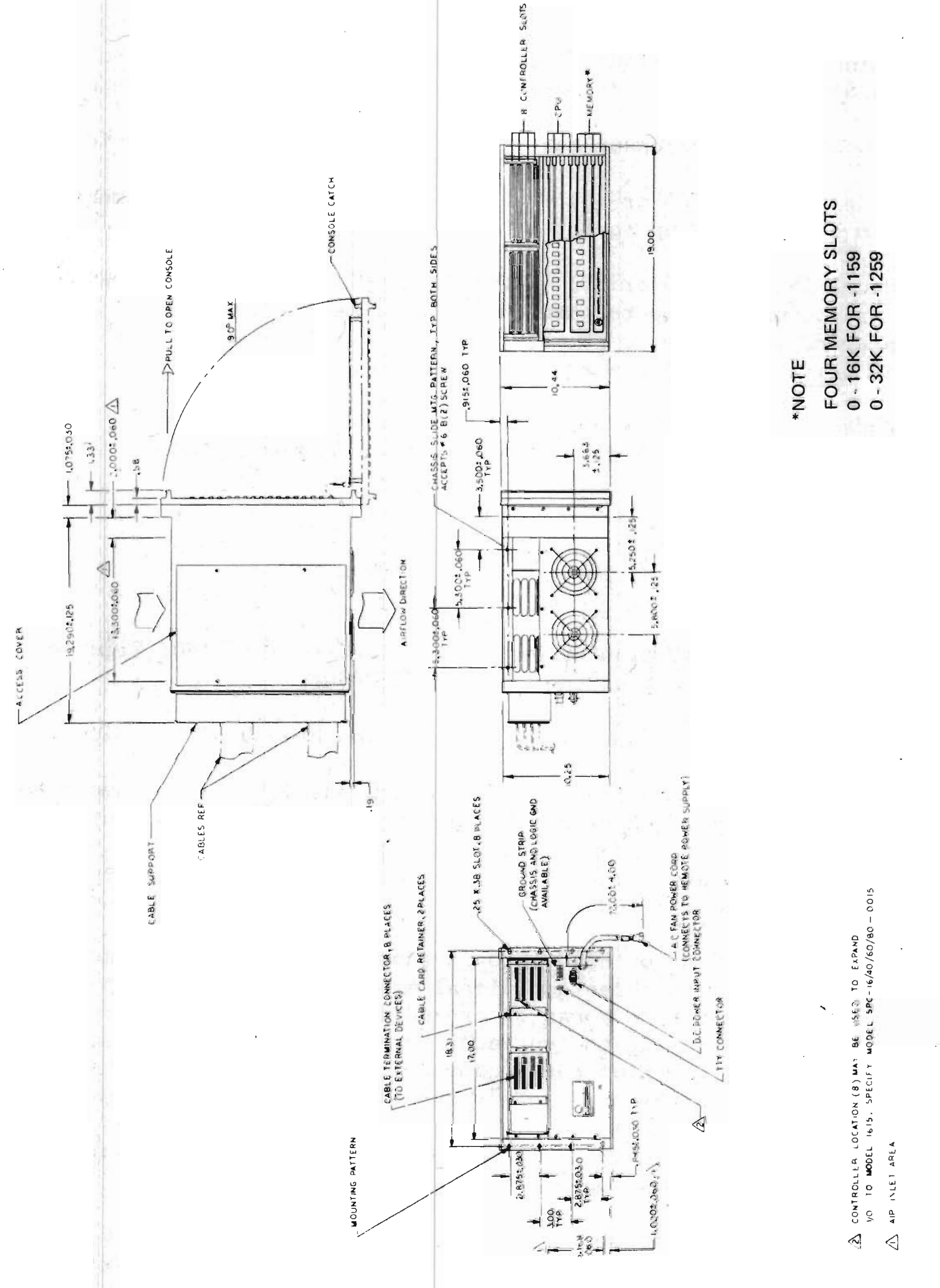
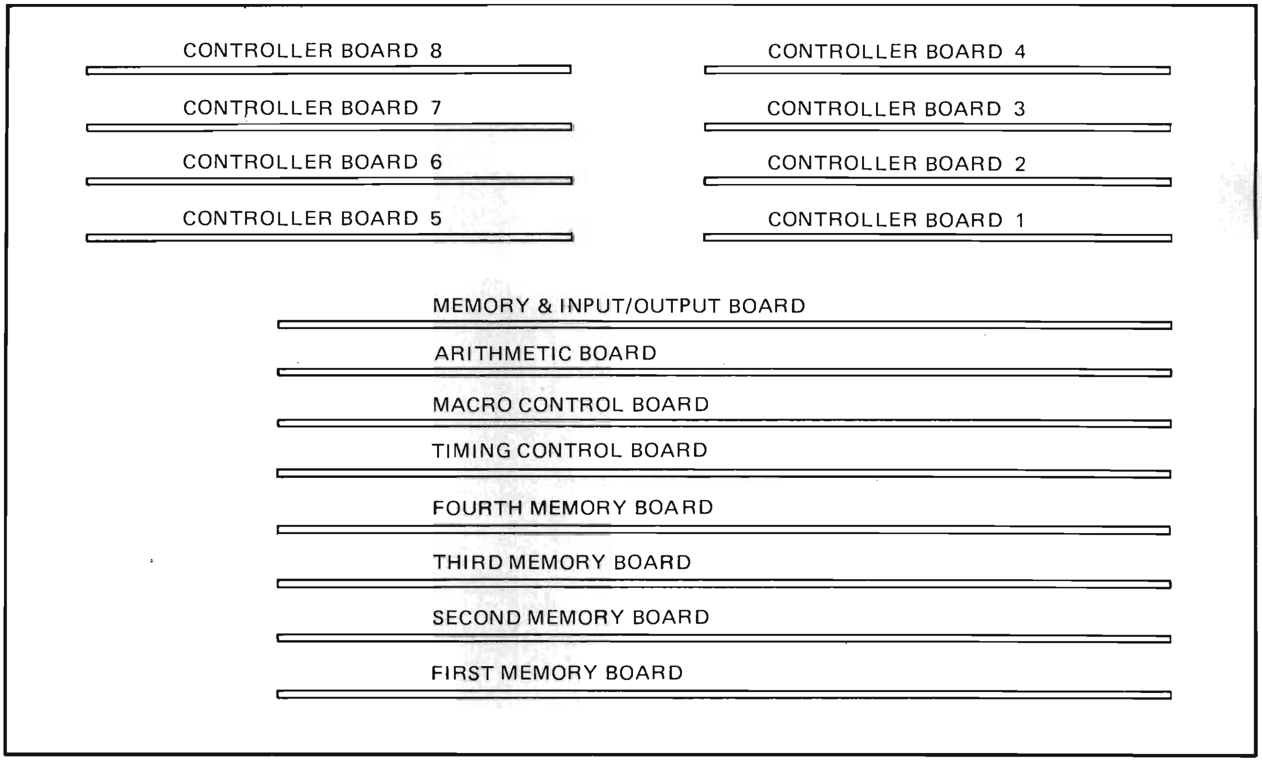


Figure 2-1. Installation Drawing, SPC-16/40/60/80
Model 1640 (or 60 or 80) -1159, -1259 or -1359



NOTES

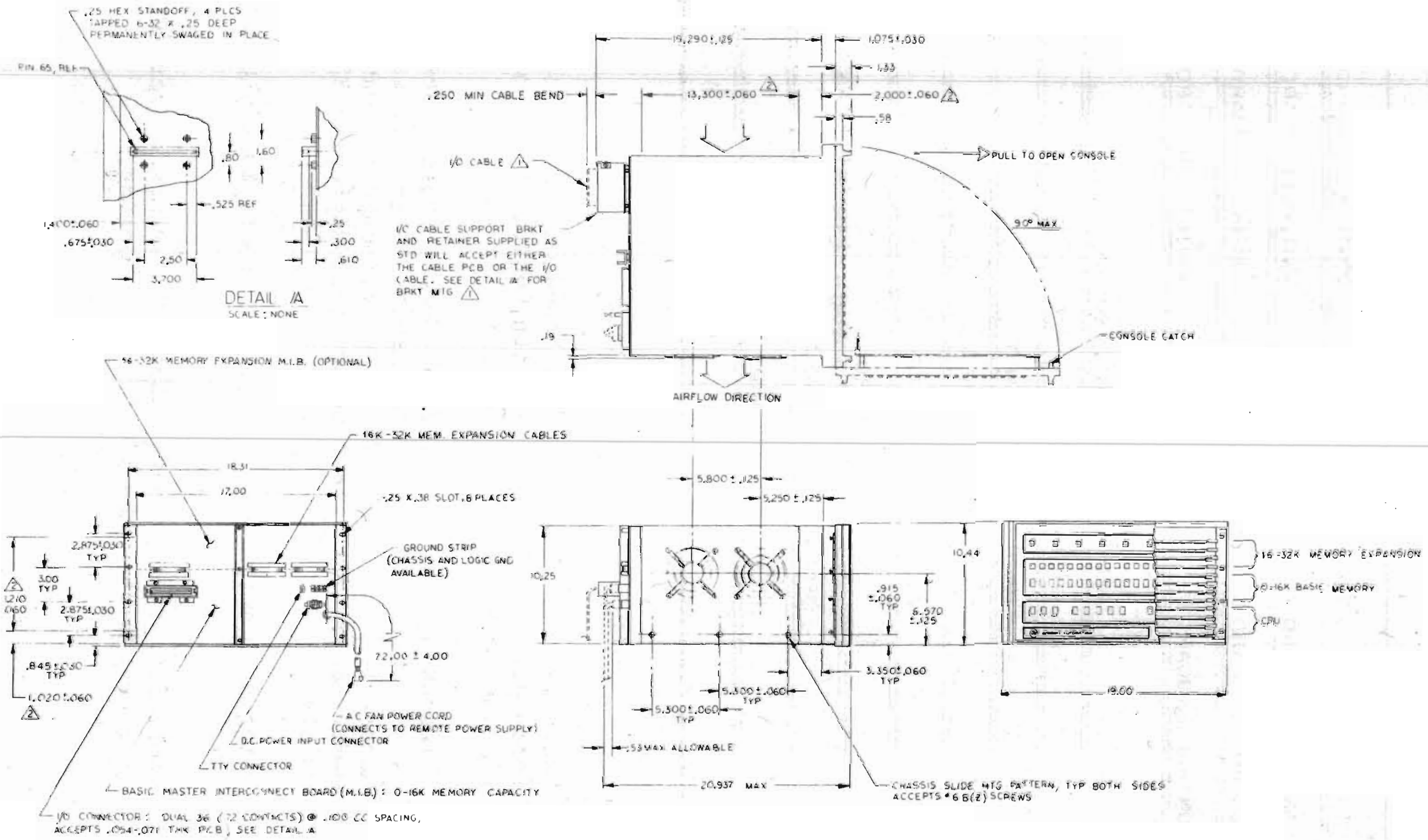
1. Controller board numbers are based on access priority. Peripheral unit associated with Controller Board #1 has highest access priority, etc.

2. Using 8K memory boards: First Memory Board provides 8K of memory. Adding Second Memory Board increases memory to 16K; adding Third Memory Board increases memory to 24K; all four memory boards provide 32K of memory.

3. Using 4K memory boards, all four memory boards provide 16K.

4. Controller slot 8 holds a Cable Interface Driver (CID) board in a system with the I/O Expansion option.

Figure 2-2. SPC-16 40/60/80 Internal Board Arrangement



2 DIMENSIONS INDICATE INLET AREA FOR AIR INTAKE

1 COMPUTER SUPPLIED STD WITH CABLE PCB ONLY. I/O CABLE IS SUPPLIED WITH MODEL 1615 I/O ENCLOSURE

Figure 2-3. Installation Drawing, SPC-16/45/65/85 With 0 - 32K Memory Capability
Model 1645 (or 65 or 85) -1100

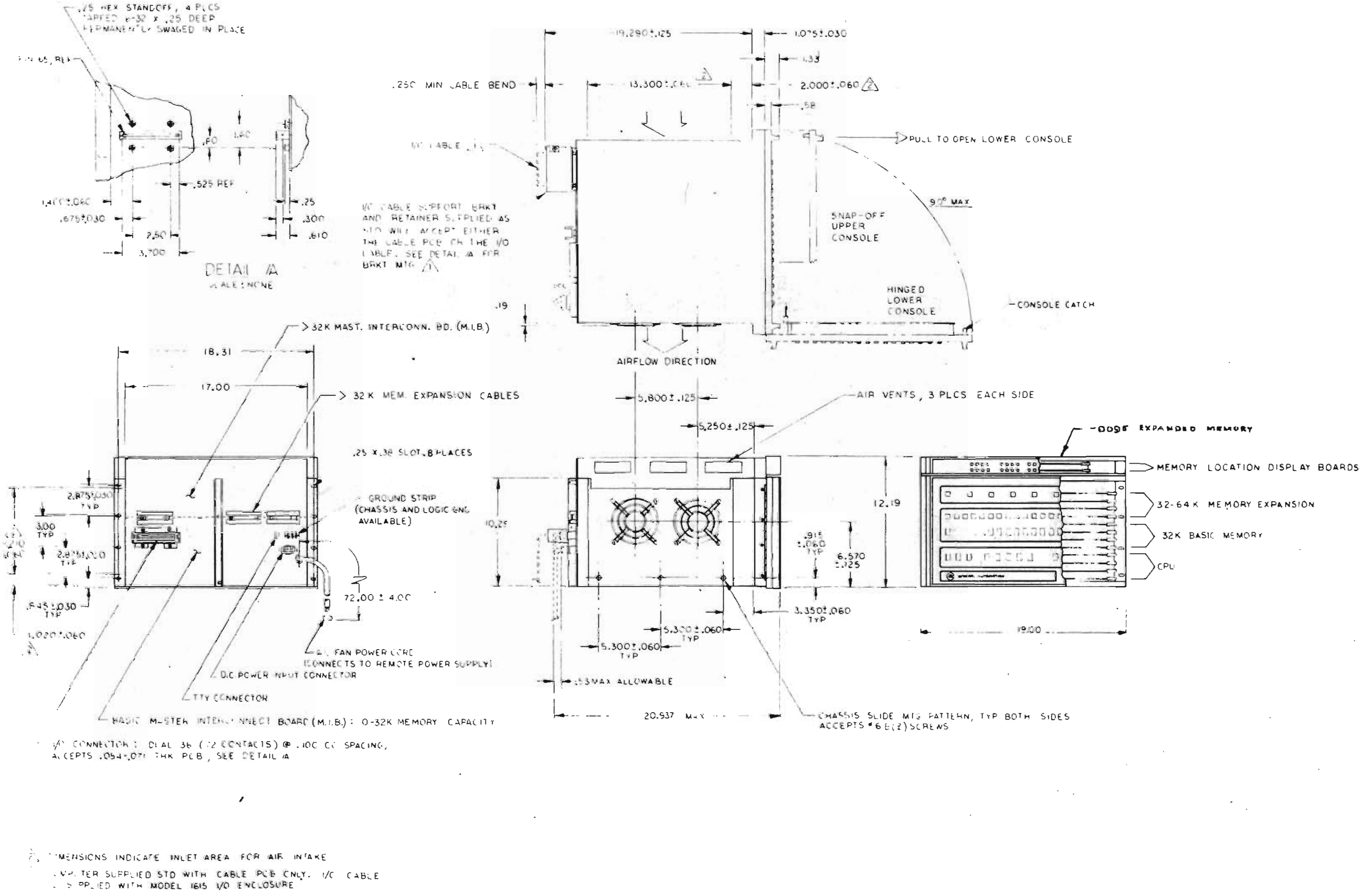
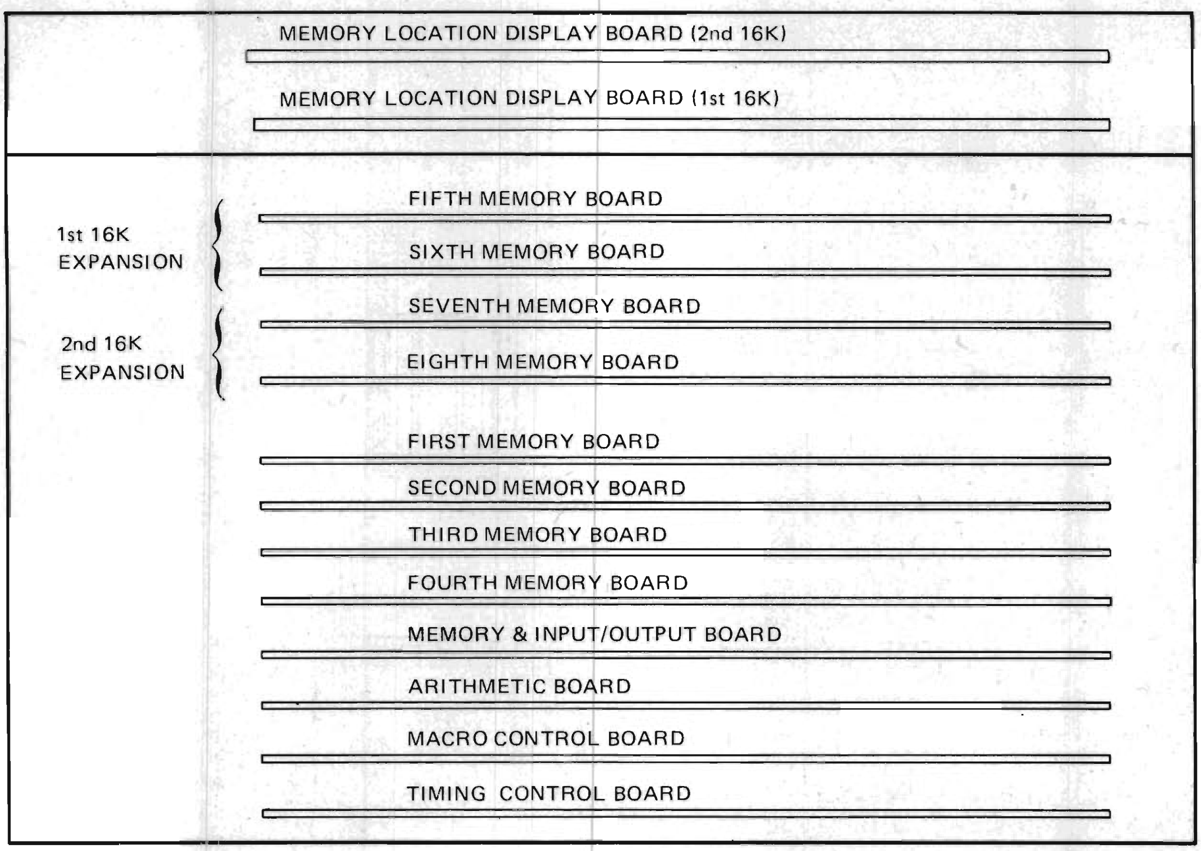

.25 HEX STANDOFF, 4 PLCS
TAPPED 6-32 X .25 DEEP
PERMANENTLY SWAGED IN PLACE
PIN 65, REF
1.400±.060
.675±.030
2.50
3.700
.525 REF
.60
1.60
.25
.300
.610
DETAIL A
SCALE: NONE
.250 MIN CABLE BEND
I/O CABLE
I/O CABLE SUPPORT BRKT AND RETAINER SUPPLIED AS STD WILL ACCEPT EITHER THE CABLE PCB OR THE I/O CABLE. SEE DETAIL A FOR BRKT MTG 1
19.290±.125
13.300±.060 2
1.075±.030
.33
2.000±.060 2
.58
PULL TO OPEN LOWER CONSOLE
SNAP-OFF UPPER CONSOLE
90° MAX
HINGED LOWER CONSOLE
CONSOLE CATCH
.19
AIRFLOW DIRECTION
32K MAST. INTERCONN. BD. (M.I.B.)
18.31
17.00
32 K MEM. EXPANSION CABLES
.25 X .38 SLOT, 8 PLACES
GROUND STRIP (CHASSIS AND LOGIC GND AVAILABLE)
2.875±.030 TYP
3.00 TYP
2.875±.030 TYP
.645±.030 TYP
1.020±.060 2
72.00 ± 4.00
AC FAN POWER CORD (CONNECTS TO REMOTE POWER SUPPLY)
D.C POWER INPUT CONNECTOR
TTY CONNECTOR
BASIC MASTER INTERCONNECT BOARD (M.I.B.): 0-32K MEMORY CAPACITY
I/O CONNECTOR: DUAL 36 (72 CONTACTS) @ .100 CC SPACING, ACCEPTS .054-.071 THK PCB, SEE DETAIL A
5.800±.125
5.250±.125
AIR VENTS, 3 PLCS EACH SIDE
10.25
12.19
.915 ±.060 TYP
6.570 ±.125
3.350±.060 TYP
5.300±.060 TYP
5.300±.060 TYP
.53 MAX ALLOWABLE
20.537 MAX
CHASSIS SLIDE MTG PATTERN, TYP BOTH SIDES ACCEPTS #6 E(2) SCREWS
-0095 EXPANDED MEMORY
MEMORY LOCATION DISPLAY BOARDS
32-64 K MEMORY EXPANSION
32K BASIC MEMORY
CPU
19.00
2 DIMENSIONS INDICATE INLET AREA FOR AIR INTAKE
1 ADAPTER SUPPLIED STD WITH CABLE PCB ONLY. I/O CABLE SUPPLIED WITH MODEL 1615 I/O ENCLOSURE

Figure 2-4. Installation Drawing, SPC-16/45/65/85 With Expanded 32 - 64K Memory Option Model 1645 (or 65 or 85) -1200 or -1300 with -0095 Memory Expansion Chassis Installed



NOTE

1. Using 8K memory boards: First Memory Board provides 8K of memory. Adding Second Memory Board increases memory to 16K, etc. All eight boards provide 64K.

2. Using 4K memory boards, all 8 memory boards provide 32K.

Figure 2-5. SPC-16 45/65/85 Internal Board Arrangement
(Expanded Memory Option Configuration Shown)

SECTION 3
FUNCTIONAL DESCRIPTION OF THE SPC-16 PROCESSOR

This section is intended to enhance the maintenance technician's general understanding of the internal organization of the SPC-16 computer.

Section 3.1 describes each major element of the Central Processing Unit. Sections 3.2 through 3.8 provide a functional description of each of the processor's boards.

3.1 ELEMENTS OF THE CENTRAL PROCESSING UNIT

Figure 3-1 illustrates the internal organization of the SPC-16 computer and identifies the principal CPU elements. Each of these elements is described in the ensuing paragraphs.

TRANSFER BUSES

Three buses provide the main communication paths within the computer:

- Augend bus (16-bits)
- Addend bus (16-bits)
- Data bus (16-bits)

These buses allow a flow of information within the computer that is both efficient and uncomplicated. The Augend and Addend buses are the paths by which all data enters the Arithmetic/Logic Unit. The Data bus is the path by which data is transferred from the Arithmetic/Logic Unit to other elements of the system.

ARITHMETIC/LOGIC UNIT

The Arithmetic/Logic Unit is the central element of the computer; it is used in executing every instruction. Through this unit pass the operands of all load, store, jump, data transfer, arithmetic, logical, shift and input/output operations.

There are three inputs to the Arithmetic/Logical Unit: the Augend bus, the Addend bus and the Link indicator. The Arithmetic/Logic Unit can perform the following operations:

- Transfer Augend bus
- Transfer Addend bus
- Add Augend to Addend
- Subtract Addend from Augend
- Add Link to Augend
- Add Augend, Addend and 1
- Complement Augend
- OR Augend and Addend
- XOR Augend and Addend
- AND Augend and Addend

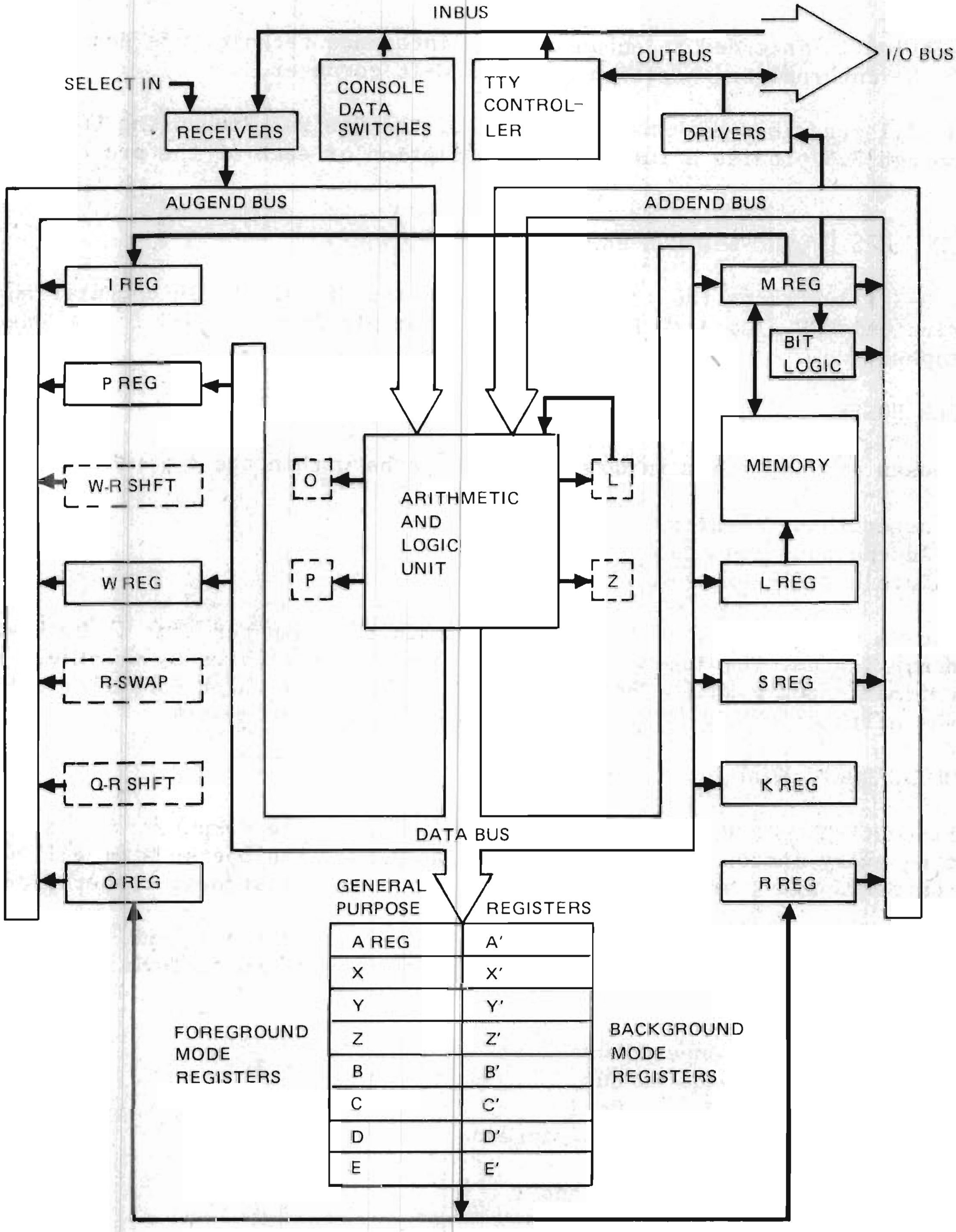


Figure 3-1. Internal Arrangement of the CPU

The 16-bit result of any operation leaves the Arithmetic/Logic Unit via the Data bus. The Link, Zero, Plus and Overflow indicators reflect certain properties of the results of some operations (see S-Register).

GENERAL PURPOSE REGISTERS

The following eight general purpose, 16-bit hardware registers are standard on all SPC-16 models:

Register	Description	Function
A	General Purpose	Accumulator
X	Index/General Purpose	Used for indexed addressing or as A-register
Y	Index/General Purpose	As X-register
Z	Index/General Purpose	As X-register
B	General Purpose	As A-register
C	General Purpose	As A-register
D	Base-Relative Addressing/General Purpose/Pre-Indexing	Contains the base address for base relative addressing as pre-index. Otherwise used as A-register.
E	Subroutine Linking/ General Purpose	During subroutine or interrupt execution, contains return address and Interrupt Status Enable Control. Otherwise used as A-register.

With the foreground/background option, eight additional general purpose registers may be used. Only one set of registers is active at a given time. Instructions are provided to activate either set (i.e., change to foreground or background mode). Any reference to a register, say the A-register, refers to the active one of the pair, either A if in foreground mode or A' if in background mode. However, the inactive registers retain their contents while they are inactive.

Subsequent register descriptions apply equally to foreground or background registers. All discussions of register operations imply the active set.

The General Purpose Registers are implemented in a fast access "scratch pad" memory. A four-bit code is used to select the register being referenced for reading or writing. The high order bit of the select code comes from the Foreground (F) indicator; the three low order bits are supplied by the CPU logic as required for the execution of a particular instruction.

F INDICATOR

1 → Foreground Registers
0 → Background Registers

CPU Logic

000	A-Register
001	X-Register
010	Y-Register
011	Z-Register
100	B-Register
101	C-Register
110	D-Register
111	E-Register

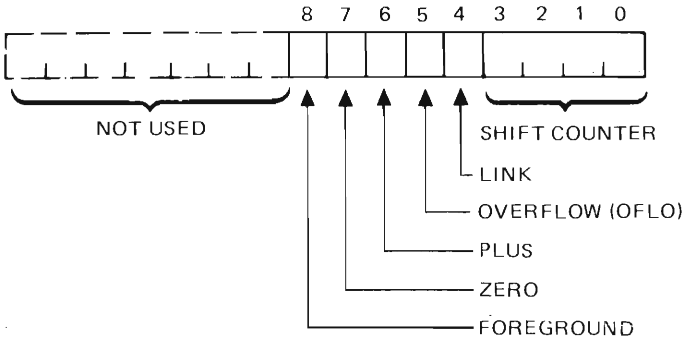
Information written into a selected register always comes off the Data Bus. Information read from the selected register goes to the Q-Register, the R-Register, or both.

Instructions are provided to perform the following operations on any of the eight general purpose registers.

- Load register from memory (word or byte).
- Store register in memory (word or byte).
- Arithmetic and logical operations between any two registers. ADD, SUBtract, OR, AND, eXclusive OR (XOR), transfer the contents of one to another.
- Arithmetic and logical operations between any register and memory (literal addressing).
- Compare any register with memory.
- Increment, decrement, complement or clear any register.
- Any register may be shifted in a variety of ways.
- Exchange the bytes in any register.
- Execute the contents of any register as an instruction.
- Input and Output of data to any peripheral may be performed with any register.

S-REGISTER

The S-Register, or STATUS Register, contains the Shift Counter (4-bits) and the Link, Overflow, Plus, Zero and Foreground/Background indicators.



An indicator is "set" if it contains a one, and is "reset" if it contains a zero.

The Shift Counter is used in the execution of the Shift instructions. A Shift instruction specifies a General Purpose register and the number of shifts to be performed. Each time the selected register is shifted, the Shift Counter is incremented and compared with the corresponding bits in the I-Register (which includes the number of shifts specified by the instruction) to determine whether or not another shift cycle is required. At the completion of the operation, the Shift Counter will contain the number of shifts minus one that were actually performed.

The Link Indicator is affected by two types of operations: arithmetic and shift.

1. Arithmetic. The Link is set (=1) if there is a carry, or reset (=0) if there is no carry, from bit 15 of a previous arithmetic operation. An instruction is provided to add the contents of the Link to any general purpose register. Thus, a carry out of bit 15 from one register containing the right half of a number may be added to another register containing the left half of the number, allowing the contents of the two registers to be treated as a single number.

2. Shift. During the execution of a shift instruction, the Link indicator contains the last value shifted out of bit 0 of the selected register.

Instructions are provided for setting, resetting and testing the Link indicator.

The Overflow Indicator indicates an arithmetic overflow for an add or subtract instruction. The Overflow indicator is set (=1) if the result of an addition or subtraction operation has a sign different from the original sign of the destination register. Otherwise the indicator is reset (=0).

Instructions are provided to test the Overflow indicator for 1 or 0, which also resets it.

The Plus indicator is set to indicate that the result of the last arithmetic, logical or transfer operation is positive (i.e., bit 15 is 0). If the result is negative, the Plus indicator is reset.

The Zero indicator is set if an arithmetic, logical or transfer operation results in zero (i.e., all bits of the result are 0). Otherwise, it is reset, indicating that the result is not zero. The Zero indicator is set to the complement of the state of a bit that is tested when the SBIT, RBIT and TBIT instructions are executed.

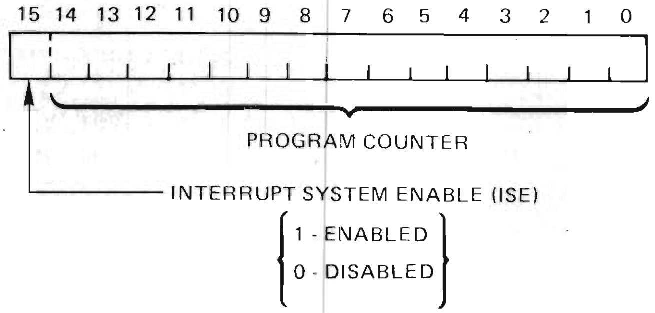
The Foreground indicator contains 1 if the Foreground Registers are active and 0 if the Background Registers are active. (If the Foreground/Background option is not installed, the indicator always contains zero.)

The S-Register can be loaded from the Data bus and gated onto the Addend bus. Instructions are provided to transfer the contents of the S-Register to any general purpose register and vice-versa.

P-REGISTER

Bits 0-14 of the P-Register are used as the program sequence register (called the "Program Counter").

The high order bit of the P-Register contains the Interrupt System Enable (ISE) indicator. (When loading the P-Register via the computer console, the ISE bit is not affected.)



The Program Counter normally contains the address of the next instruction to be executed. Its contents (bits 0-14) are transferred to the L-Register (Memory Location Register) following the execution of the current instruction, to read the next instruction from memory; then its contents are incremented (its contents are incremented again if the instruction is a double-word instruction). Instructions that modify instruction sequencing (e.g., JMP) do so by altering the contents of the P-Register. The P-Register can be changed by "jump", "conditional jump" (skip), "jump-to-subroutine" and "return" instructions.

When the Interrupt System Enable contains 1 following the execution of any instruction (except "JMP"), the processor will acknowledge an interrupt if one is pending. When this bit contains 0, no interrupt can be acknowledged (i.e., the interrupt system is disabled).

When an interrupt is acknowledged, a "jump-to-subroutine" (JSR) indirect instruction is executed, the contents of the P-Register (Program Counter = P_{0-14} and ISE = P_{15}) are saved in the E register; and the ISE bit in the P-Register is reset, inhibiting other interrupts while the current interrupt is being serviced.

The ISE indicator is controlled by "interrupt enable", "interrupt inhibit", "restore interrupt system enable", "jump-to-subroutine" and "return" instructions, i.e., data cannot be entered into P15 from the console.

L-REGISTER

The L-Register (15 bits) is the memory location register. It contains the absolute address of the memory location from which or to which an instruction, an address or data is to be read or stored. Although the L-Register is logically 15 bits long, it is physically only 12 bits long; the high-order 3-bits which are not stored, are decoded to select one of eight 4K portions of memory and the low-order 12-bits indicate the address within this selected portion.

The L-Register is always loaded from the Data Bus.

M-REGISTER

The M-Register (16-bits) is the memory buffer register. It receives all data read from and contains all data written into the computer's memory.

The M-Register can be loaded from the Data bus. It can be selected onto the Addend bus, into the I-Register or through the I/O Drivers for the Outbus. (The Outbus always reflects the data in the M-Register.)

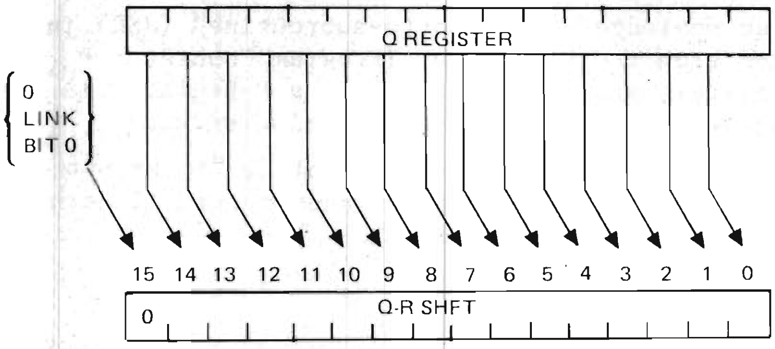
I-REGISTER

The I-Register (16-bits) always holds the instruction currently being executed (only the first word of a double-word instruction). After memory is cycled to fetch an instruction, the M-Register is transferred directly to the I-Register. The instruction decoding logic operates with the bit pattern in the I-Register to determine the sequence of events necessary to execute the instruction.

Q-REGISTER

The Q-Register serves as a buffer register between any general purpose register and the Augend bus.

A non-addressable register, labeled Q-R SHFT in Figure 3-1 is the Q-Register with bits shifted one position to the right.



Q-R SHFT may also be gated onto the Augend bus.

R-REGISTER

The R-Register (16-bits) serves as a buffer register between a general purpose register and the Addend bus.

A non-addressable register (16-bits), labeled R-SWAP in Figure 3-1, is the R-Register with the bytes reversed.

$$(R)_{0-7} \rightarrow (R\text{-}SWAP)_{8-15}$$

$$(R)_{8-15} \rightarrow (R\text{-}SWAP)_{0-7}$$

This non-addressable register may also be gated onto the Augend bus.

W-REGISTER

The W-Register holds the effective operand address for most address calculations. It is loaded from the Data bus and may be selected onto the Augend bus.

A non-addressable register, labeled W-R SHFT in Figure 3-1 is the W-Register with bits shifted one position to the right. W-R SHFT may also be gated onto the Augend bus (used for hardware multiply only).

K-REGISTER

The contents of the K-Register (16-bits) are continuously displayed on the computer console's Register Display Indicators. An instruction is provided to transfer the contents of any General Purpose Register to the K-Register and, thereby, display it on the console while the computer is executing a program.

The K-Register is loaded from the Data bus. The four low-order bits of the K-register are used to drive a four-bit Digital/Analog converter. This analog voltage is present on the I/O Bus as VU+.

3.2 CONSOLE BOARD

The Console Board contains the K Register and switch logic associated with the Display Console. The functional sections of the Console Board (Assembly Number 31D01446A) are shown in their approximate locations in Figure 3-2. The console switches and indicators are described in Appendix D.

The basic SPC-16 instruction set may be executed with this board removed.

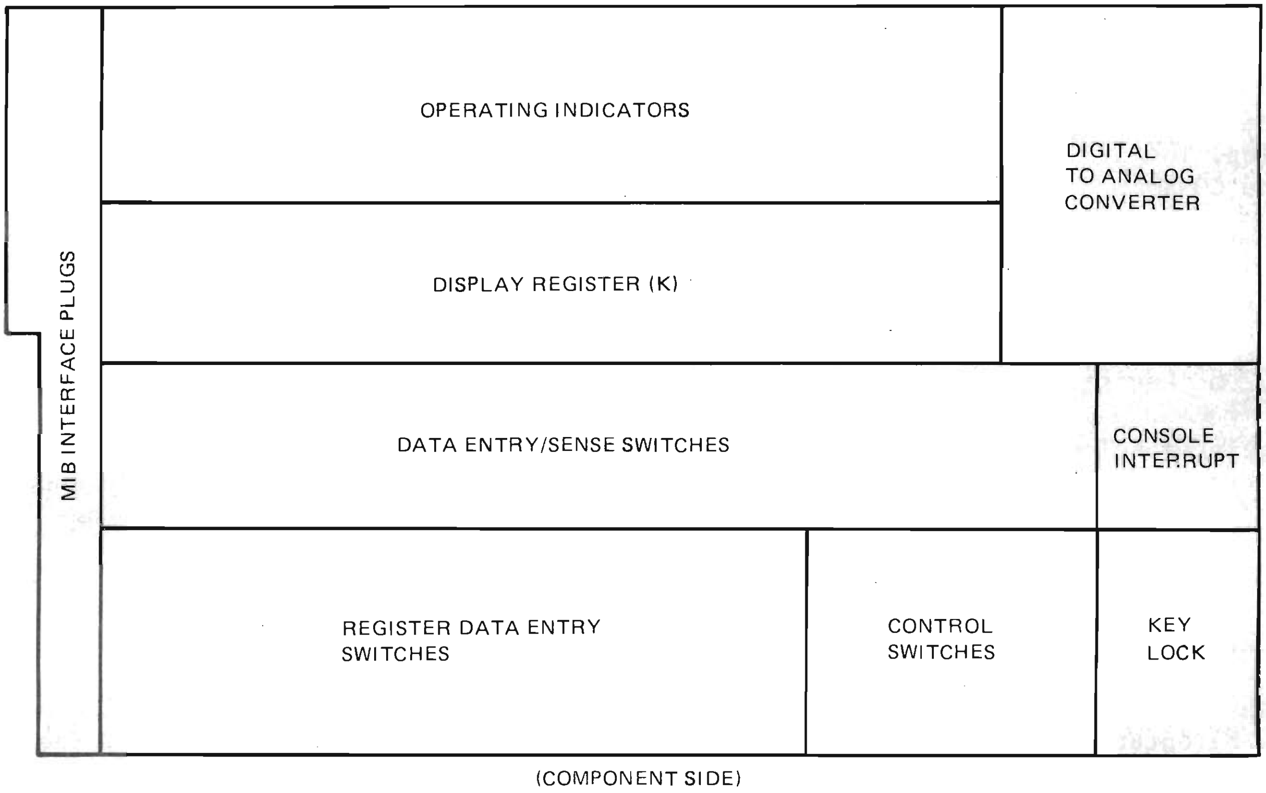


Figure 3-2. Console Board Functional Arrangement

3.2.1 Display Register (K)

The display register flip-flops receive inputs directly from the data bus. The outputs drive lamp drivers which control the console lamps.

3.2.2 Operating Indicators

The following operating indicators are located on the Console Board (left to right): Foreground Mode, ROM, DMA Acknowledge, Interrupt Acknowledge, Interrupt System Enable, Operations Monitor Alarm, Idle and Run.

3.2.3 Digital to Analog Converter (VU)

Consists of a D/A converter on the least significant four bits of the console display register yielding a maximum output voltage of 7.5 volts.

3.2.4 Control Switches

The STEP switch and CONSOLE INTERRUPT switch each operate through a debounce flip-flop. The SYSTEMS RESET switch and the ENTER switch are momentary contact switches. The CONSOLE ENABLE (keylock) switch is a 2-position switch. The RUN/IDLE switch and the two SAVE I switches are 2-position switches. The small SAVE I switch located behind the console cover is a maintenance SAVE I switch which operates independent of the RUN/IDLE switch for continuous execution of a single instruction in the run mode. The SAVE I switch on the console operates only in the idle mode.

3.2.5 Register Select Switches

Register select switches 1, 2 and 4, the STATUS, select P and select I switches are all 2-position switches. The switches are wired in a priority sequence such that I is the highest priority, W next, P next, etc. Register select switches 1, 2 and 4 select registers in the scratch pad chip.

3.2.6 Register Data Entry Switches

The 16 register display entry switches are all 2-position switches and allow data to be entered into the console display register and the selected register. They also may be used as sense switches and data input switches during program execution.

3.2.7 Console Interrupt

When the console interrupt switch is activated, the CINT flip-flop is set and an interrupt request (level 7) is presented to the processor.

3.2.8 Key Lock

When in the locked position (full CCW), the key lock disables all switches on the console except the 16 data bit sense switches.

3.2.9 MIB Interface Plugs

The MIB interface plugs allow the console board to be plugged into the Master Interconnect Board.

3.3 ARITHMETIC BOARD

The Arithmetic Board contains the Augend, Addend and Data Bus, the Arithmetic/Logic Unit and the General Purpose registers. The functional sections of the Arithmetic Board (Assembly number 31D01333A) are shown in their approximate locations in Figure 3-3.

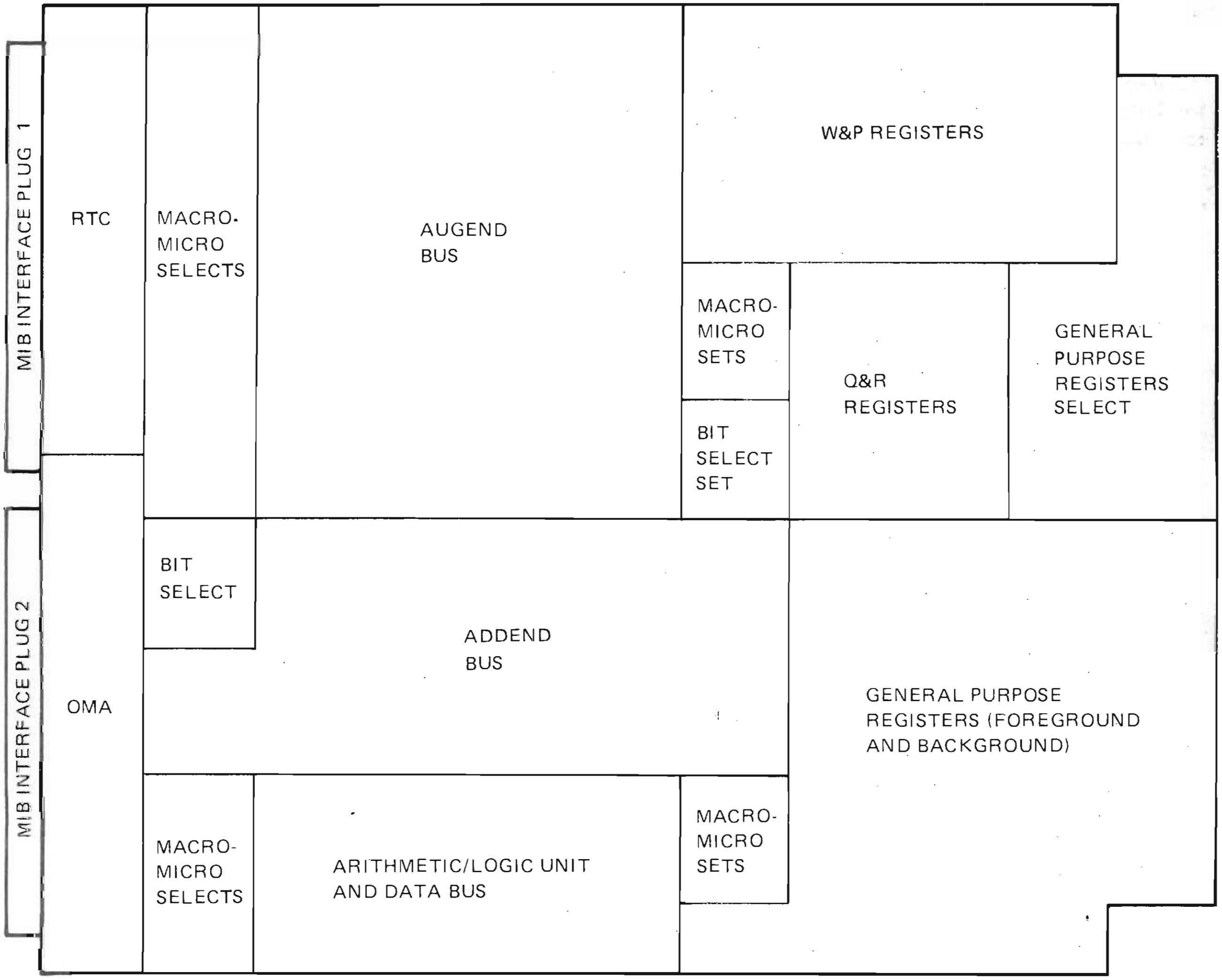


Figure 3-3. Arithmetic Board Functional Arrangement

3.3.1 Micro/Macro Selects and Sets

The selects and sets associated with micro control are ORed with the macro selects and sets associated with standard processor operation.

3.3.2 Augend Bus (AU)

Two expanders for each bit are used to accommodate the eight input selections to the augend bus.

3.3.3 Addend Bus (AD)

The addend bus has fewer inputs to each bit than the augend bus, so one expander accommodates the inputs to each of the 16 bits.

3.3.4 Arithmetic/Logic Unit and D Bus

The ALU is a 16-bit binary, full adder that performs all arithmetic and logical functions. Data to the ALU comes from the augend and addend bus and outputs from the ALU constitute the data (D) bus.

3.3.5 General Purpose Registers Select

The three register address bits (RAB0, RAB1 and RAB2) decode one of eight registers. The state of the foreground flip-flop determines which group of eight registers (either foreground or background) is selected.

3.3.6 General Purpose Registers

The 16 general purpose registers are A, A', X, X', Y, Y', Z, Z', B, B', C, C', D, D', E and E'. The primed registers indicate background registers and the unprimed registers indicate foreground registers.

3.3.7 P and W Registers

Bits 0 - 15 of the data bus are inputs to the W register. Bits 0 - 14 of the data bus are inputs to the P register. The P register retains the program counter; the W register is a general purpose working register used by most of the SPC-16 instructions. For memory reference instructions, the EA is placed in the W register.

3.3.8 Bit Select Logic

Bits 5, 6 and 7 of the I register, during bit instructions, decode one of 8 bits to be set, reset or tested. The left or right bytes (bits 0 - 7 or 8 - 15) are affected depending on whether the index register is odd or even.

3.3.9 Relative Time Clock (RTC)

The DOC1- clock is presented to a divide-by-four counter. The output of the divide-by-four counter is input to three decade counters. The output, RTCCK-, clocks the set logic of the RTC flip-flop, i.e., the RTC period is 1000 machine (memory) cycles. The RTC will continue to count even though the RTC flip-flop is set. The RTC flip-flop will be reset when the CPU acknowledges the RTC interrupt.

3.3.10 Operations Monitor Alarm (OMA)

RTCCK- is input to the OMA circuitry. If the OMA mask flip-flop has been set by a PMA instruction, an input to the two decade counters is initiated. The second decade counter output is input to the two final stages of OMA flip-flops. A PMA instruction must be executed to turn on the OMA. Normally, 200,000 machine cycles lapse after the last PMA command before the OMA times out resulting in a stall.

During a power fail, the OMA mask flip-flop is set and the RTC and OMA counters are initialized to time out in 100 machine cycles. One-hundred machine cycles are available after power fail until the CPU is forced into a stall.

3.3.11 Q and R Registers

Any of the general purpose registers can be inputs to either the Q or R register. The outputs of the R register go to either the augend or the addend bus. The outputs of the Q register go to the augend bus.

3.4 MASTER INTERCONNECT BOARD (MIB)

The MIB contains the connectors for all other boards in the mainframe as well as the external connectors to the teletype, power supply and extended I/O.

The SPC-16 40/60/80 series MIB (Assembly number 31D01640A) is a single board having the functional arrangement shown in Figure 3-4.

The SPC-16 45/65/85 uses two MIB's. The main MIB (Assembly number 31D01648A) is combined with a dummy panel (Assembly number 13C00189A01) if the memory expansion option is not employed and with a memory expansion MIB (Assembly number 31D01650A) if the memory expansion option is used. Figure 3-5 shows the functional arrangement of the main MIB and Memory Expansion MIB.

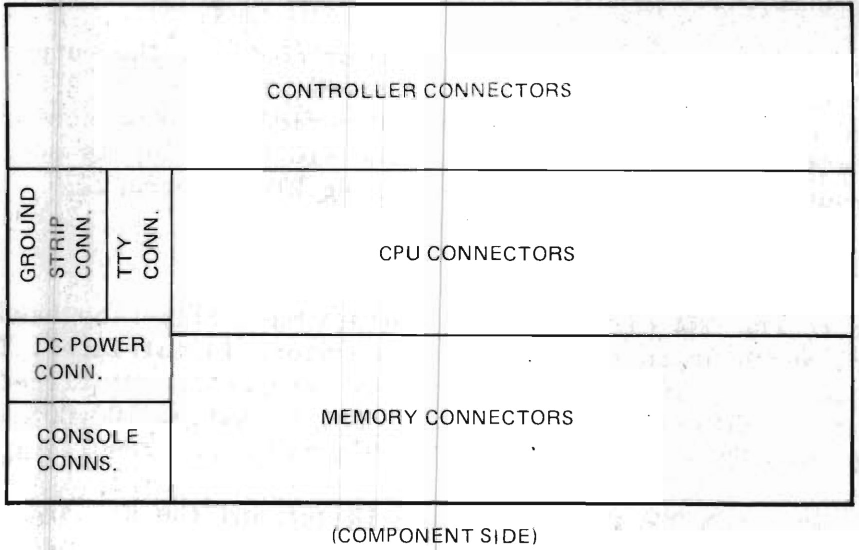
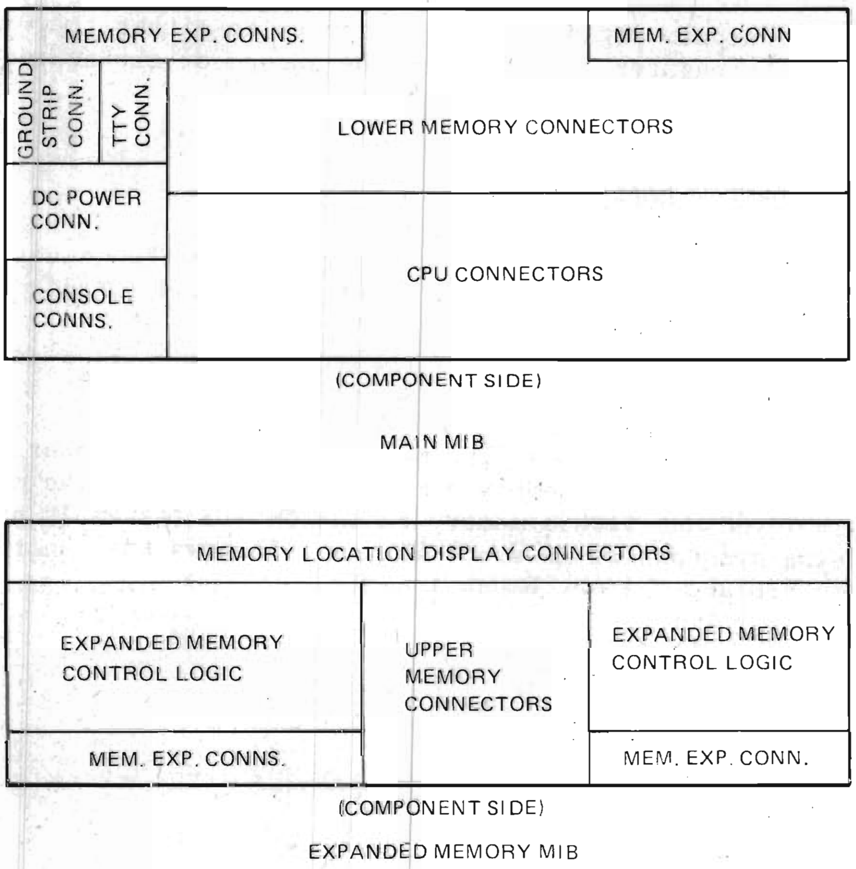


Figure 3-4. SPC-16 40/60/80 Master Interconnect Board Functional Arrangement



MAIN MIB

MEMORY LOCATION DISPLAY CONNECTORS
EXPANDED MEMORY CONTROL LOGIC
UPPER MEMORY CONNECTORS
EXPANDED MEMORY CONTROL LOGIC
MEM. EXP. CONNS.
MEM. EXP. CONN.
(COMPONENT SIDE)

EXPANDED MEMORY MIB

Figure 3-5. SPC-16 45/65/85 Master Interconnect Boards Functional Arrangement

3.4.1 SPC-16 40/60/80 MIB Connectors

The connectors and their respective jack designations on the SPC-16 40/60/80 MIB are as follows:

Board/Connector	Jack Designation(s)
Console	J1, J2
Memory	J3, J4, J5, J6
Timing	J7, J8
Macro	J9, J10
Arithmetic	J11, J12
MIO	J13, J14
Ground	J15
TTY	J16
DC Power	J17
Controllers	J18 - J33

3.4.2 SPC-16 45/65/85 Main MIB Connectors

The connectors and their respective jack designations on the SPC-16 45/65/85 Main MIB are as follows:

Board/Connector	Jack Designation(s)
Console	J1, J2
Memory	J3, J4, J5, J6
Timing	J7, J8
Macro	J9, J10
Arithmetic	J11, J12
MIO	J13, J14
Ground	J15
TTY	J16
DC Power	J17
Memory Expansion	J18 - J21

3.4.3 SPC-16 45/65/85 Expanded Memory MIB Connectors

The connectors and their respective jack designations on the SPC-16 45/65/85 Expanded Memory MIB are as follows:

Board/Connector	Jack Designations
Memory	J1, J2, J3, J4
Memory Expansion	J5, J6, J7
Memory Location Display	J8, J9, J10, J11

3.5 MACRO CONTROL BOARD (MAC)

The Macro Control Board contains the Instruction (I) register and associated instruction decode network, the ALU function bus, the ZPOL indicator logic and the shift counter. The functional sections of the Macro Control Board (Assembly number 31D01331A) are shown in their approximate locations in Figure 3-6.

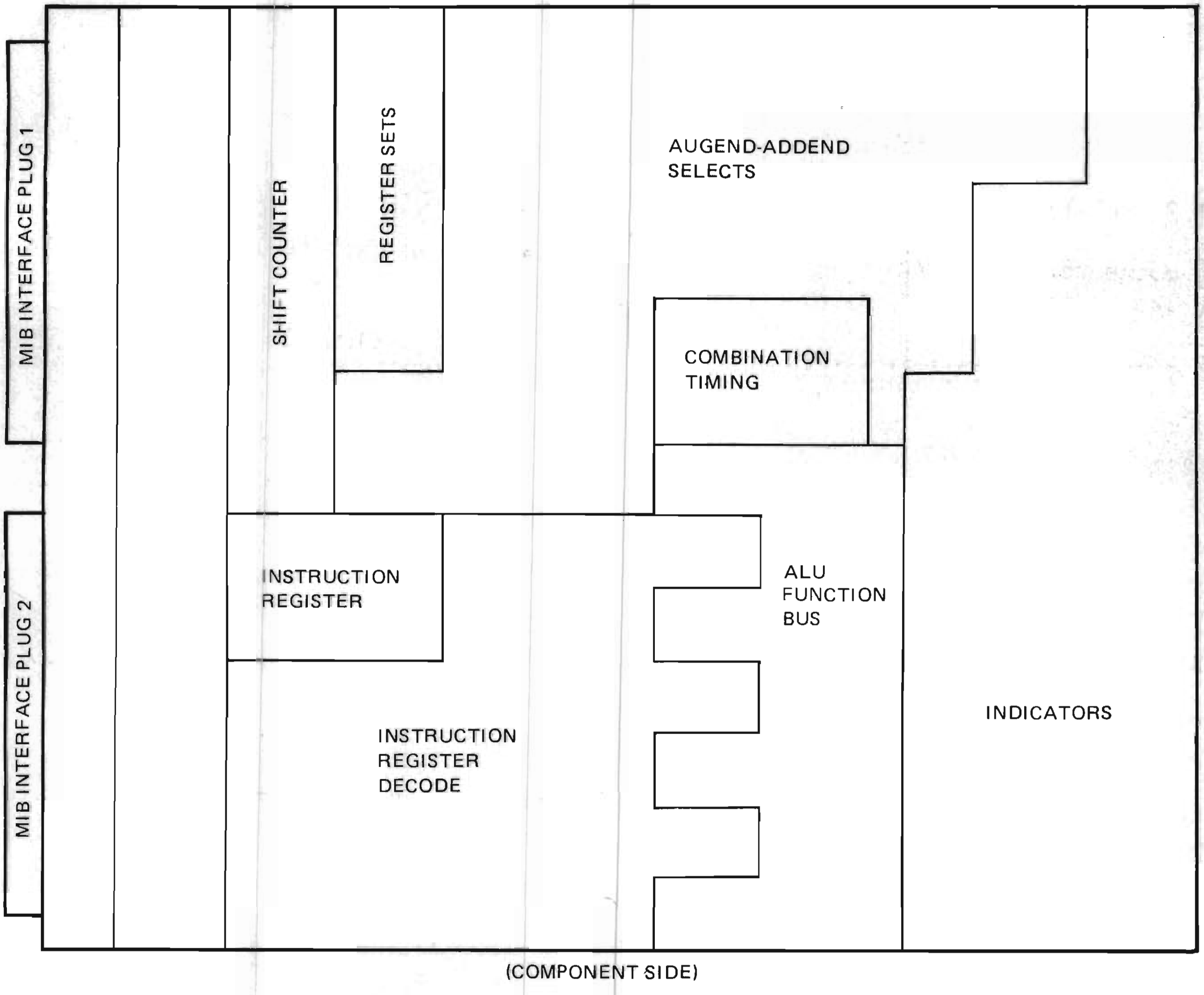


Figure 3-6. Macro Control Board Functional Arrangement

3.5.1 Instruction (I) Register

The 16-bit instruction register inputs come directly from the memory data register. The I register maintains the instruction throughout its execution; it is set during the instruction fetch cycle and remains set until the subsequent instruction fetch.

3.5.2 Instruction Register Decode

Bits 12, 13, 14 and 15 decode the op code of the instruction. Bits 0 - 11 of the instruction select different functions, depending on the type of instruction. Refer to Figure 6-2 for a summary of SPC-16 instruction set decodes.

3.5.3 Indicators

The indicators located on the Macro Control Board are listed as follows:

a. Zero (ZERO)
b. Plus (PLUS)
c. Overflow (OFLO)
d. Link (LINK)
e. Interrupt System Enable (ISE)
f. Foreground (FRGND)

3.5.4 Augend-Addend Select Functions

The augend-addend select functions (shown in Figure 4-1) are generated on the Macro Control Board. They are SMMS6, SM04, SM58, SM9, SMX8, SMX9, SBITR, SBITL, SZPOL, SS, SRLL, SRRR, SHFW, SIO, SQLL, SQRR, SRRL, SRLR. SW, SP, SI and SHFQ are generated on the Timing Control Board. These signals are defined in Section 4.

3.5.5 Register Set Functions

The I and K register set functions (STI and STK) are found on the Macro Control Board. STQ, STR, STML, STMR, STL, STI, STW and STP are generated on the Timing Control Board. These signals are defined in Section 4.

3.5.6 Shift Counter

The four-bit shift counter contains the shift count during an MPY, DVD or SHIFT instruction.

3.5.7 Combination Timing

The combination timing signals that are generated on the Macro Control Board are: N891+, N8N12+, N893+, N89+, N8D03+1, N8D02+1, N852+, N853+ and N813+. These signals are defined in Appendix A.

3.5.8 MIB Interface Plugs

The MIB interface plugs J9 and J10 allow the Macro Control Board to be plugged into the MIB.

3.6 TIMING CONTROL BOARD

The Timing Control Board contains the timing, sequence control and Augend-Addend Select logic for the processor. The functional sections of the Timing Control Board (Assembly number 31D01811A) are shown in their approximate locations in Figure 3-7.

The theory of processor timing is described in detail in Section 5.

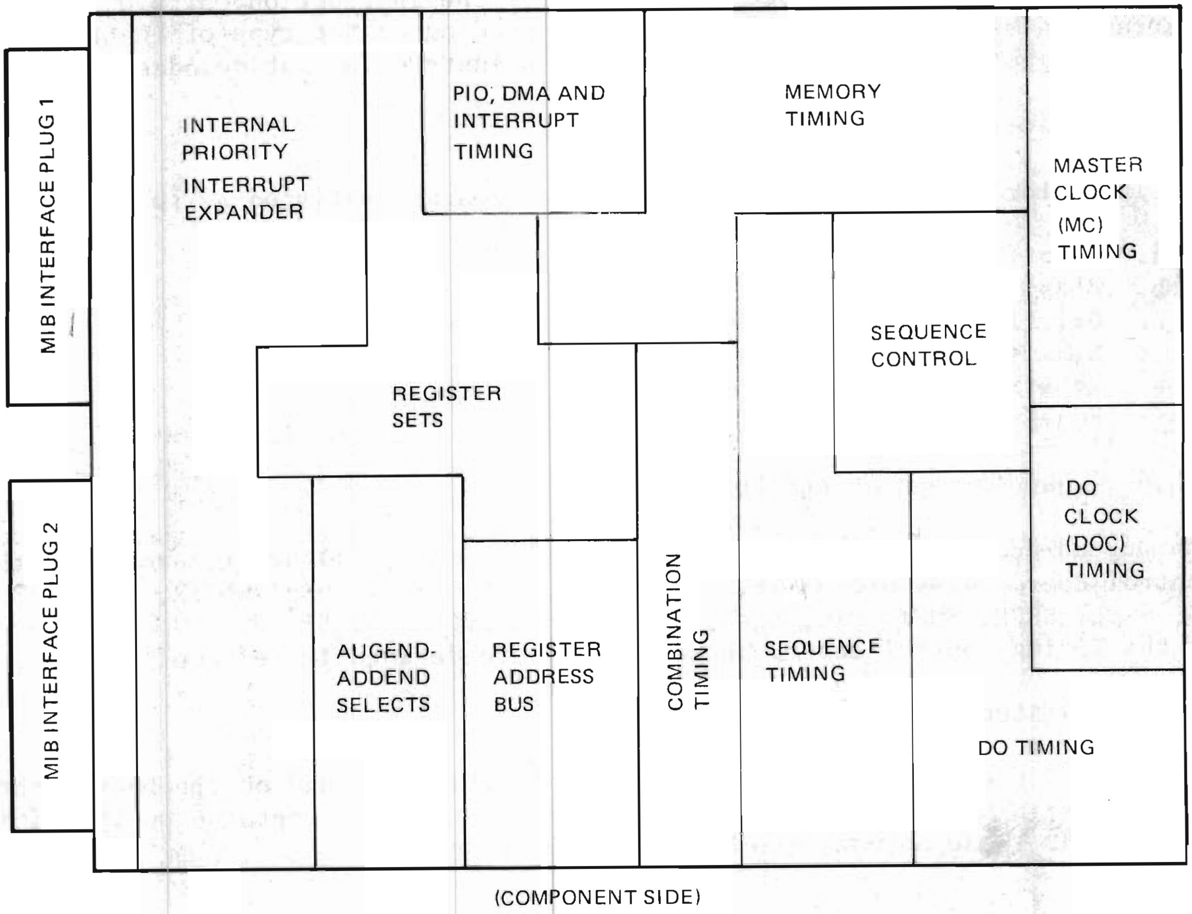


Figure 3-7. Timing Control Board Functional Arrangement

3.6.1 Master Clock

The crystal-controlled master clock pulses are 90, 60 and 50 ns, respectively, from leading edge to leading edge.

3.6.2 DO Clock (DOC) Timing

The master clocks control the DOC flip-flops. The master clock oscillator and the DOC's are located at the rear of the board. Four DOC states are counted within each DO state (see Section 5.1.2).

3.6.3 DO Timing

The DO times (DO1, DO2, DO3, DO4) and the associated enable signals are distributed throughout the processor. Four DO states normally comprise a sequence state. Exceptions are defined in the instruction sequencing section (Section 6). The DO flip-flops control ALU cycles, the sequence flip-flops control memory cycles. The DO flip-flops normally cycle sequentially, i.e., DO1-, DO2-, DO3-, DO4. However, during fast ROM processing or shift instructions, or MPY, or DVD, the DO logic can provide a DO2-, DO3-, DO2-, DO3, etc., sequence.

3.6.4 Sequence Control

The sequence advance clocks (SACKA, SACKB) advance the sequencer control flip-flops. The sequence varies as a function of the instruction being executed.

3.6.5 Sequence Timing

The nine sequence flip-flops (S1 through S9) control the instruction sequence timing in the SPC-16 processors. Refer to Section 5 for a detailed description of the sequence states.

3.6.6 Combination Timing

Some of the combination timing signals are generated on the Timing Control Board. The signal N812 means $\overline{S8}\cdot S1\cdot D2$. Refer to the glossary of terms, Appendix A, for a description of the combination timing signals.

3.6.7 Memory Timing

The memory start enable signal (MSE) is generated at either DO2 or DO4 time. The memory timing (MTIM) flip-flop, the read cycle (READ+) signal, the write cycle (WRIT-) signal, inhibit (INHT+) and memory start clock (MSTC) are all generated on this board. Also, the clear M register (CLRML, CLRMR) signals are generated here.

3.6.8 Programmed Input/Output (PIO), DMA, Interrupt Timing

The following I/O and DMA timing signals are generated on the Timing Control Board: interrupt acknowledge (IACK), data channel acknowledge (DACK), acknowledge time (ACKT), extended processor option flip-flop (XPOF), DO1 or DO3 (D103), D3 general-purpose clock, multiplex-address-control (MACF), data transfer pulse (DTP), function address pulse (FAP), select out (SOUT) and data channel data transfer pulse (DDTP). The DTP and DDTP pulses provides timing for data transfers. DTP is for Programmed I/O (PIO) single-word transfers. DDTP is for Data Channel I/O (DCIO), designated by the D prefix. The general-purpose clock is provided for external peripheral controllers and is not used internally. The SOUT signal will be true any time data-in is not selected and the processor is not in a read cycle.

The WWPCK signal is used on the Arithmetic Board as a clock to set W, P and to write into the general-purpose registers. The set I logic and some of the console switch termination circuits are on this board.

3.6.9 Register Address Bus

The register address bus (RAB0, RAB1, RAB2) signals decode the general-purpose register (either background or foreground) to be used in each state within the execution of an instruction.

3.6.10 Augend-Addend Selects

The following augend and addend select signals are generated on the Timing Control Board: select DMA (SDMA), select P (SP), select and increment P (SIP), select W (SW), select I (SI) and shift Q (SHFQ).

3.6.11 Register Sets

The following register set signals are generated on the Timing Control Board: set Q enable (STQE), set R enable (STRE), set M left (STML), set M right (STMR), set P enable (STPE) and set W enable (STWE). The above register set enable signals are combined with timing signals (WWPCK) to accomplish the ultimate set function.

3.6.12 Priority Interrupt Expander

The six holding flip-flops for the interrupt request signals are active at all times except at interrupt acknowledge (IACK-) time. During IACK no level changes are allowed while the Branch Vector address is decoded. The second group of six flip-flops are the mask register used for the disarming. The inputs to the second group of flip-flops come from the M register during a set mask instruction. Each flip-flop holds the arm/disarm condition for a corresponding individual interrupt.

The seven interrupt request lines (IRQ1 - 6 and PFIR) are encoded to form a priority address via the DMA lines. The relative time clock reset (RTCR) and the reset of the console interrupt (RCINT) are also located here.

DMA12, DMA13 and DMA14 allow the power fail and restart vectors to be moved to the 16K or 32K memory position.

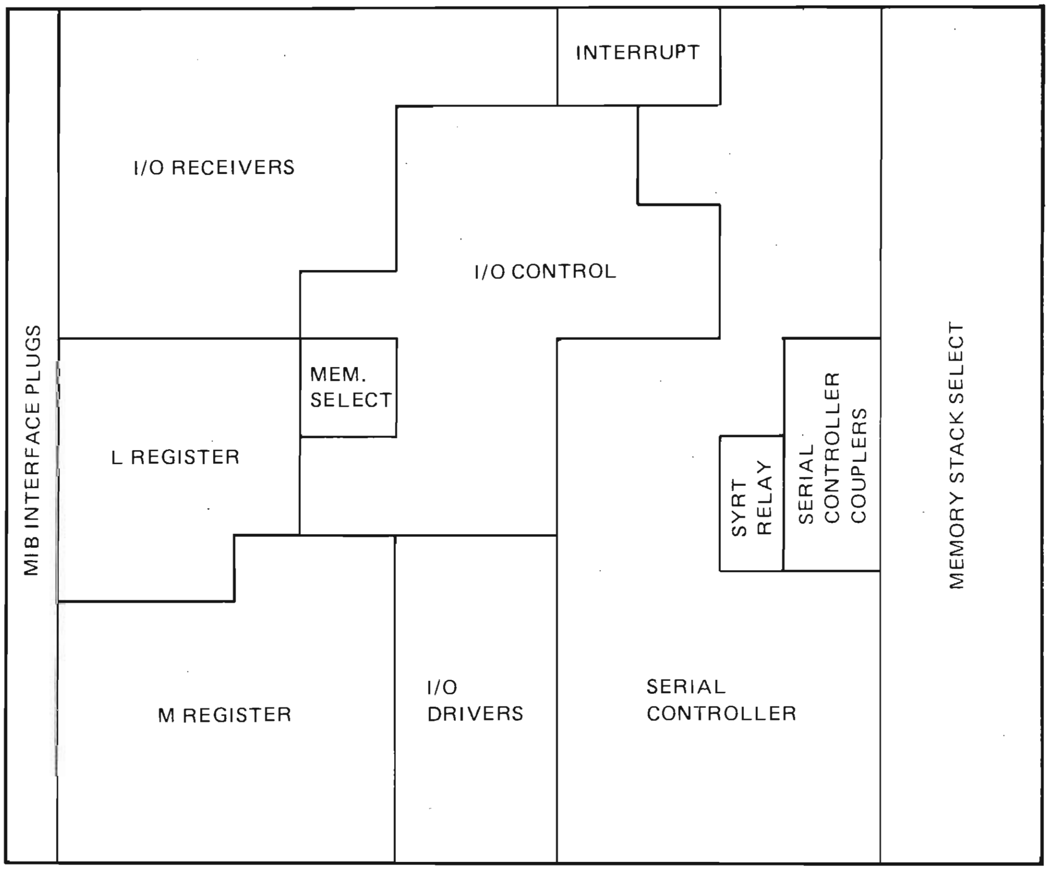
3.6.13 MIB Interface Plugs

The MIB interface plugs J7 and J8 permit the Timing Control Board to be plugged into the MIB.

3.7 MEMORY AND INPUT/OUTPUT BOARD (MIO)

The Memory and Input/Output Board contains the Interrupt and I/O Control logic, the Teletype Controller, the M and L registers and the Memory Select Logic.

The functional sections of the MIO Board (Assembly number 31D01628A) are shown in their approximate locations in Figure 3-8.



(COMPONENT SIDE)
Figure 3-8. Memory and Input/Output Board Functional Arrangement

3.7.1 Memory Data (M) Register

The 16-bit M register inputs come from the data bus and memory. The outputs of the M register go to the bit logic, the addend bus, the I/O drivers, the I register and memory.

3.7.2 Memory Address (L) Register

The 12-bit L register contains the 12 least-significant bits of the address of the data or instruction being accessed from memory. The inputs to the 12 flip-flops come directly from the data bus and they are clocked by set L (STL+).

3.7.3 I/O Drivers, Receivers and Termination

The I/O drivers place 16 data bits from the M register onto the Out-Bus (OTB00-15). The I/O receivers place 16 data bits from the In-Bus onto the augend bus (IN00-15). The I/O Bus lines are terminated at both the CPU and controller ends by 100 ohm pull-up resistors. Each I/O driver has a drive capability of approximately 250 milliamps.

3.7.4 Memory Stack Select

The most significant 3 bits of the memory address on the data bus (DB12-14) are used to select one of eight 4K memory portions. Once the selection has been made, the selected portion remains selected throughout the memory cycle independent of the status of the selection bits. Hence there is no requirement to hold these most significant 3 bits of the memory address in the L register.

3.7.5 System Reset and Memory Guard Relays

The system reset relay provides the signals SYRT- and SYRT-I through normally closed contacts to ground. When power reaches an unsafe condition (via PFD-), the relay is de-energized resulting in a SYRT condition. While power is in the safe range, the relay is energized, yielding a false SYRT signal. The memory guard relay provides -15V to the memory through normally open contacts. While power is safe, the contacts are closed and -15V is presented to memory. When power is not up, the relay drops out, opening the contacts and removing the -15V signal.

3.7.6 Serial (Teletype) Controller

The serial controller contains the serial controller clock, the clock divider, clock counter, busy flip-flop and transmit/receive complete flip-flop.

The four device control pulses required for teletype operation are: DCPA, DCPB, DCPC, and DCPD.

DCPA sets the controller to the transmit mode, DCPB sets it to receive mode, DCPC sets it to the echo mode. The difference between the receive and the echo mode is that in the echo mode the characters received are transmitted back to the teletype without being printed. The DCPD control pulse puts the serial teletype line in an open or spacing condition (if the transmit mode is already active).

The serial controller clock oscillator frequency is adjustable. It is adjusted at the factory, but may require adjustment later. Refer to Section 10.1.2 for adjustment procedure.

The clock counter counts the number of clock pulses. The data to or from the teletype is 11 bits. The first bit of the teletype format is a sync bit. On input, it synchronizes the serial controller with the teletype. On output, it indicates to the teletype that a data word is being transmitted. The next eight bits are the data word and the last two bits are mark bits (also classified as sync bits).

The clock pulses are counted in order to maintain synchronization with the teletype. Data from the processor is placed into the shift register to keep the timing sequence continuously going so that at the end of the two mark bits, a sync bit is automatically transmitted for the next character with no time lag between characters. When 10.5 clock pulses have been received, the transmit receive complete flip-flop and the busy flip-flop establish the TTY interrupt (TTYI-) and the teletype test line true condition.

This indicates to the processor that it is time to place a new data word in the shift register or to get the data that has been input.

3.7.7 Serial Controller Optical Couplers

The serial controller optical couplers (Z9HA, Z9HB) form the interface between the controller and the teletype. The couplers are used to electrically and physically isolate the two. The three lines to the teletype are: serial data in (SDI-), serial data out (SDO-) and common (SDC-S).

3.7.8 MIB Interface Plugs

The MIB interface plugs J13 and J14 permit the MIO Board to be plugged into the MIB.

3.8 MEMORY BOARD (MEM)

Three types of memory boards are currently being used:

- o A 4K low-profile planar array memory board (Assembly number 31D01635A).

- o An 8K low-profile planar array memory board (Assembly number 31D01594A).

- o A 16K planner array memory board (Assembly number 31D01783A).

The functional arrangements of components on each of these boards are shown in Figures 3-9, 3-10 and 3-11 respectively.

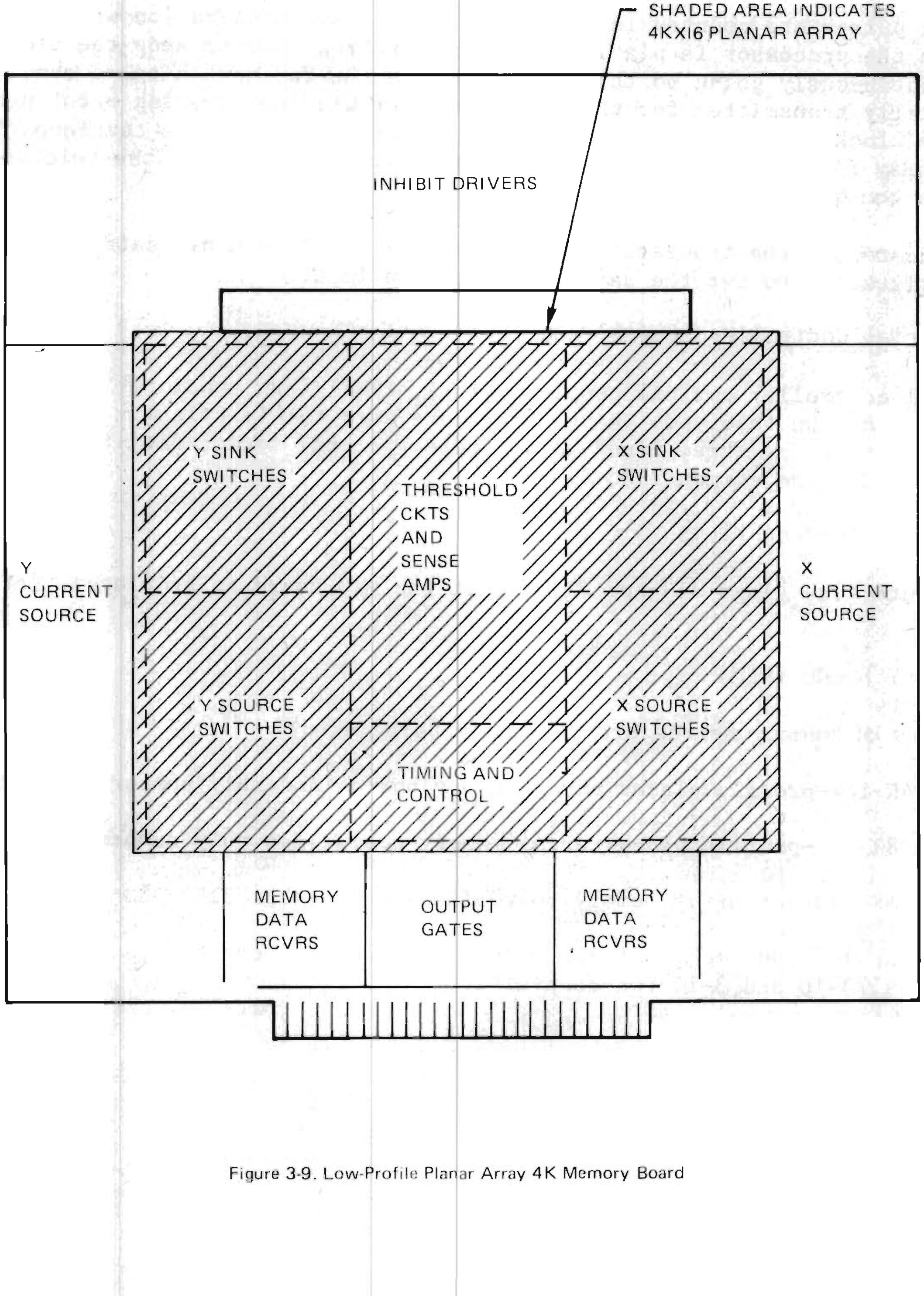


Figure 3-9. Low-Profile Planar Array 4K Memory Board

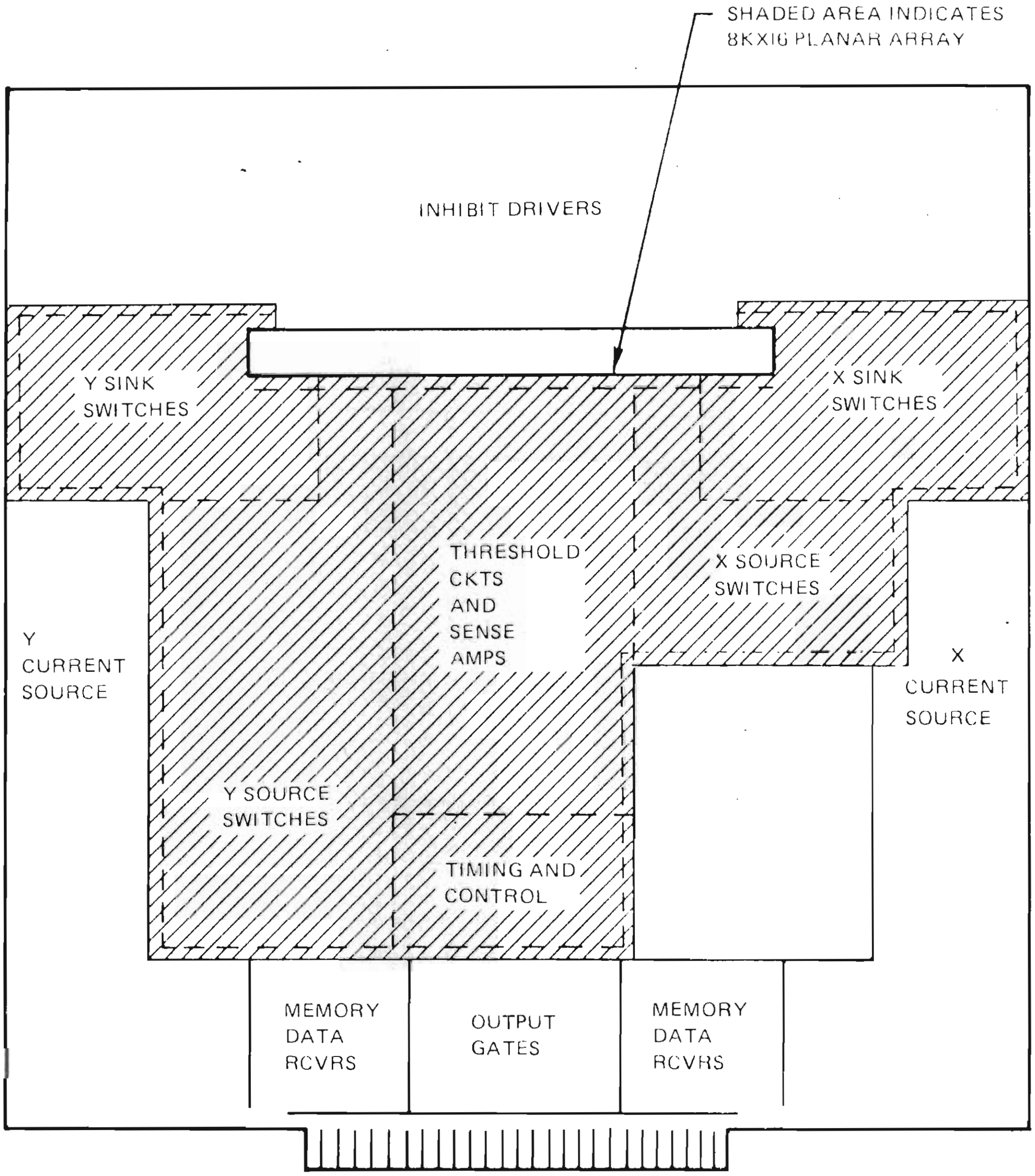


Figure 3-10. Low-Profile Planar Array 8K Memory Board

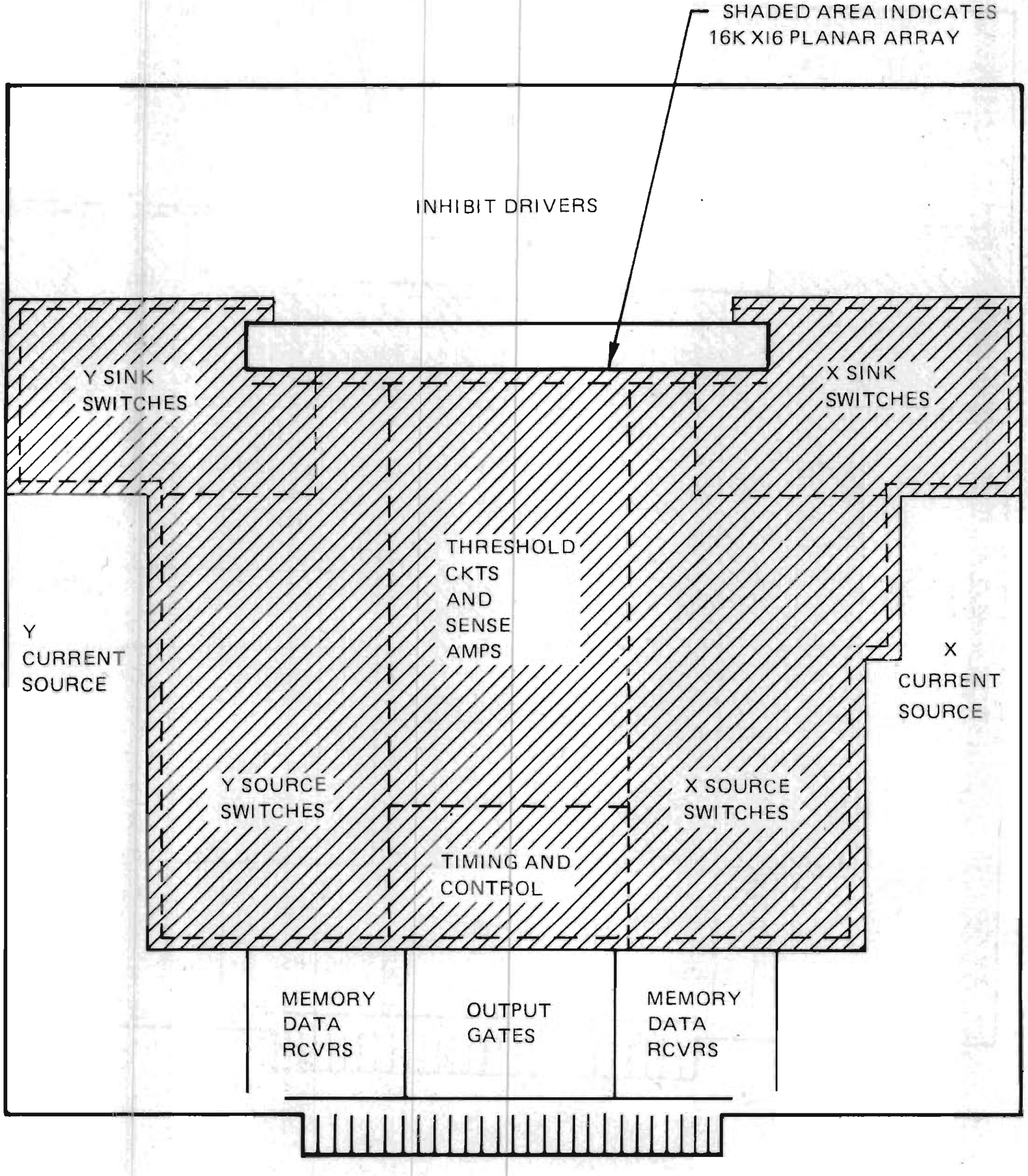


Figure 3-11. Planar Array 16K Memory Board

SECTION 4
DATA FLOW

4.1 DATA FLOW ORGANIZATION

The purpose of this section is to supply the user with data flow information. This manual will not go into detailed logic analysis and assumes the reader to be familiar with standard DTL and TTL integrated circuit characteristics.

Figure 4-1 is a block diagram of the processor showing the logical functions which are utilized during the instruction execution.

The SPC-16 processors employ a bus philosophy in the routing of data between registers. This produces an efficient and yet highly capable processor. The following paragraphs provide brief descriptions of the logical blocks shown in Figure 4-1.

4.1.1 Arithmetic/Logic Unit (ALU)

The Arithmetic/Logic Unit can act as a 16-bit binary adder or as a logical function generator, depending upon the level of its mode control input (FINH). With FINH low, the ALU performs arithmetic operations on the Augend Bus and the Addend Bus; with FINH high, the ALU performs logical operations on these busses. The particular operation performed is selected by the level of the four Function Select signals (FX0, FX1, FX2 and FX3) input.

Table 4-1 is the truth table for arithmetic operations; Table 4-2 is the truth table for logical operations.

The outputs of the ALU comprise the Data Bus. A carry out sets the Link indicator and a zero detect sets the Zero indicator. A Plus indicator and Overflow indicator are also available to monitor operations performed by the APU.

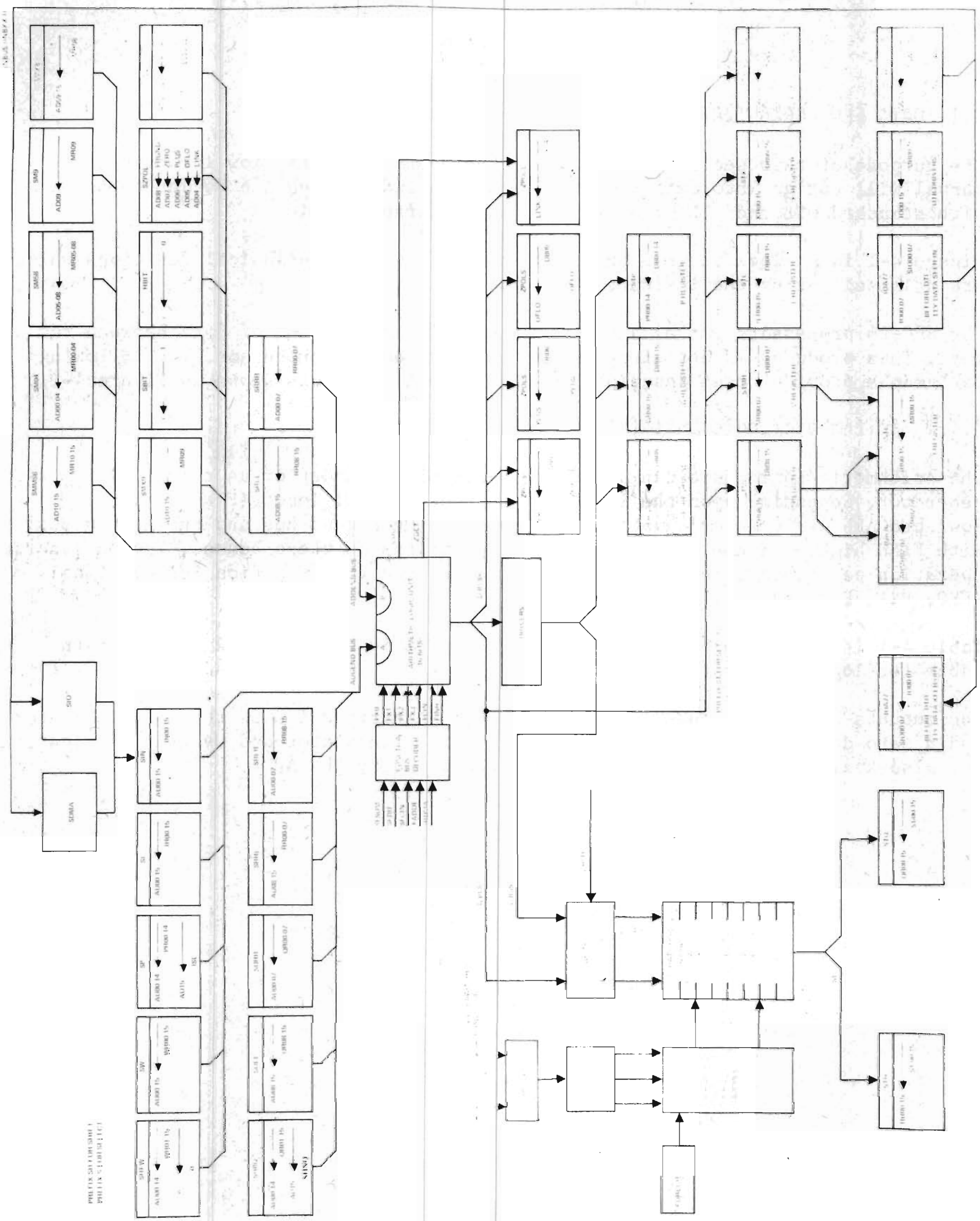


Figure 4-1. SPC-16 Block Diagram

Table 4-1. Table of Arithmetic Operations

Function Select				Negative Logic, *
FX3	FX2	FX1	FX0	Low Levels Active
L	L	L	L	F = A minus 1
L	L	L	H	F = AB minus 1
L	L	H	L	F = $A\overline{B}$ minus 1
L	L	H	H	F = minus 1 (2's complement)
L	H	L	L	F = A plus ($A+\overline{B}$)
L	H	L	H	F = AB plus (A+B)
L	H	H	L	F = A minus B minus 1
L	H	H	H	F = $A+\overline{B}$
H	L	L	L	F = A plus (A+B)
H	L	L	H	F = A plus B
H	L	H	L	F = $A\overline{B}$ plus (A+B)
H	L	H	H	F = A+B
H	H	L	L	F = A plus A
H	H	L	H	F = AB plus A
H	H	H	L	F = $A\overline{B}$ plus A
H	H	H	H	F = A

* With mode control (FINH) and FCIN low.

Table 4-2. Table of Logic Functions

Function Select				Negative Logic
FX3	FX2	FX1	FX0	Low Levels Active*
L	L	L	L	$F = \overline{A}$
L	L	L	H	$F = \overline{AB}$
L	L	H	L	$F = \overline{A}+B$
L	L	H	H	F = Logical 1
L	H	L	L	$F = \overline{A+B}$
L	H	L	H	$F = \overline{B}$
L	H	H	L	$F = \overline{A \oplus B}$
L	H	H	H	$F = A+\overline{B}$
H	L	L	L	$F = \overline{A}B$
H	L	L	H	$F = A \oplus B$
H	L	H	L	$F = B$
H	L	H	H	$F = A+B$
H	H	L	L	F = Logical 0
H	H	L	H	$F = A\overline{B}$
H	H	H	L	$F = AB$
H	H	H	H	$F = A$

* With mode control (FINH) high, FCIN irrelevant

4.1.2 Augend Bus

The Augend Bus is a series of 16 lines (AU00 - AU15) that terminate in the augend inputs to the ALU. The select and shift terms associated with the Augend Bus are described in the following paragraphs:

SHFW: Shifts the W register bits 1 - 15 onto Augend Bus bits 0 - 14. Zero is shifted into bit 15 of the Augend Bus. Used in the MPY algorithm.

SW: The Select W circuitry gates the contents of the W register onto the Augend Bus directly.

SP: The Select P circuitry gates bits 0 - 14 of the P register onto the Augend Bus. The Interrupt System Enable flip-flop is gated onto bit 15 of the Augend Bus.

SI: The Select I circuitry gates the contents of the I register onto the Augend Bus directly.

SIN: The Select Data-In circuitry gates data from either the DMA address bus or the I/O bus onto the Augend Bus.

SHFQ: The Shift Q register logic gates bits 1 - 15 of the Q register to bits 0 - 14 of the Augend Bus. SHNQ is shifted to bit 15 of the Augend Bus.

SQLL: The Select Q register Left to Left logic gates bits 8 - 15 of the Q register to bits 8 - 15 of the Augend Bus.

SQRR: The Select Q register Right to Right logic gates bits 0 - 7 of the Q register to bits 0 - 7 of the Augend Bus.

SRRL: The Select R register Right to Left logic gates bits 0 - 7 of the R register to bits 8 - 15 of the Augend Bus.

SRLR: The Select R register Left to Right logic gates bits 8 - 15 of the R register to bits 0 - 7 of the Augend Bus.

4.1.3 Addend Bus

The Addend Bus is a series of 16 lines (AD00 - AD15) that terminate in the addend inputs to the ALU. The Addend Bus - selects route data to the addend portion of the ALU from the M register. Bits 8 and 9 of the M register are extended to full 16-bit size for displacement computation of effective operands. Provision is made for a one or a zero to be forced into the selected bit of the M register for a SBIT and RBIT instruction, respectively. The status of the Zero, Plus, Overflow, Link and Foreground indicators and the contents of the Shift Counter are also gated to the Addend Bus. R register left-to-left and right-to-right transfers are provided for BYTE instructions.

The select terms associated with the Addend Bus are described in the following paragraphs:

SMMS6: This select gates bits 10 - 15 of the M register to bits 10 - 15 of the Addend Bus.

SM04: The Select Memory Data Register 0 - 4 logic gates bits 0 - 4 of the M register to bits 0 - 4 of the Addend Bus.

SM58: The Select Memory Data Register 5 - 8 logic gates bits 5 - 8 of the M register to bits 5 - 8 of the Addend Bus.

SM9: The Select Memory Data Register 9 logic gates bit 9 of the M register to bit 9 of the Addend Bus.

SMX8: The Select Memory Data Register and Extend bit 8 logic gates bit 8 of the M register to bits 9 - 15 of the Addend Bus. This is referred to as extended displacement.

SMX9: The Select Memory Data Register and Extend bit 9 logic gates bit 9 of the M register to bits 10 - 15 of the Addend Bus. This is referred to as extended displacement.

SBIT: The Set Bit logic forces a logical one into the bit selected on the Addend Bus.

RBIT: The Reset Bit logic forces a logical zero into the bit selected on the Addend Bus.

SZPOL: The Select Indicators logic gates the following indicators to the specified Addend Bus bit:

Indicator	Addend Bus bit
Foreground	AD08
Zero	AD07
Plus	AD06
Overflow	AD05
Link	AD04

SS: The Select Shift Counter logic gates bits 0 - 3 of the Shift Counter to bits 0 - 3 of the Addend Bus.

SRLL: The Select R register Left-to-Left logic gates bits 8 - 15 of the R register to bits 8 - 15 of the Addend Bus.

SRRR: The Select R register Right-to-Right logic gates bits 0 - 7 of the R register to bits 0 - 7 of the Addend Bus.

4.1.4 Data Bus

The Data Bus is a series of 16 lines (DB00 - DB15) on the output of the Arithmetic/ Logic Unit (ALU). The Data Bus reflects the result of the operations performed on the data present on the Addend and Augend buses. The Data Bus minus lines (DB04-, DB05-, DB06- and DB07-) are inputs to the Link, Overflow, Plus and Zero indicators, respectively. Also, DBxx- lines are inputs to the write logic of the scratch pad, the Memory Data (M) register, the L register, console display (K) register and the Shift Counter. DBxx+ lines are inputs to the write logic of the scratch pad, the W register, the P register and the Foreground/Background indicator.

The following paragraphs describe the selects associated with the Data Bus.

ZPOLS: This select gates the Data Bus bits listed below to the specified indicators.

Indicator	Data Bus Bit
Foreground	DB08
Zero	DB07
Plus	DB06
Overflow	DB05
Link	DB04

Also the ZPOLS logic gates bits 0 - 3 of the Data Bus to bits 0 - 3 of the Shift Counter.

STW: The Set W logic gates bits 0 - 15 of the Data Bus to bits 0 - 15 of the W register.

STP: The Set P logic gates bits 0 - 15 of the Data Bus to bits 0 - 15 of the P register.

STML: The Set Memory Data register Left logic gates bits 8 - 15 of the Data Bus to bits 8 - 15 of the M register.

STMR: The Set Memory Data register Right logic gates bits 0 - 7 of the Data Bus to bits 0 - 7 of the M register.

STL: The Set L register logic gates bits 0 - 11 of the Data Bus to bits 0 - 11 of the L register.

STK: The Set Console Display register logic gates bits 0 - 15 of the Data Bus to bits 0 - 15 of the K register.

STI: The Set Instruction register logic gates bits 0 - 15 of the M register to bits 0 - 15 of the I register.

Certain logic functions are selectively activated during the execution of an instruction depending on the decoding of that particular instruction.

The four least-significant bits of the I register (IR00 - IR03) are decoded into 6 function bits, which control ALU operation. The decoding is shown in Table 4-3.

Table 4-3. I Register Decode

I Register				Function Bits							
IR03	IR02	IR01	IR00	FX3	FX2	FX1	FX0	FINH	FCIN	ALU Function	When Used
0	0	0	0	0	0	0	0	1	0	$\overline{A}$	CMPL
0	0	0	1	1	0	0	1	0	Link	A plus B	RLK
0	0	1	0	0	1	1	0	0	0	A minus B minus 1	DECR
0	0	1	1	1	1	1	1	0	0	$\overline{A}$	DMA XFER
0	1	0	0	0	0	0	0	1	1	$\overline{A}$	Not Used
0	1	0	1	1	0	1	0	1	1	B	LDV, RTR
0	1	1	0	0	1	1	0	0	1	A minus B	SUB
0	1	1	1	1	1	1	0	1	1	A and B	AND
1	0	0	0	1	0	0	1	1	0	A X or B	XOR
1	0	0	1	1	0	0	1	0	0	A plus B	ADD
1	0	1	0	1	0	0	1	0	0	A plus B	Not Used
1	0	1	1	1	0	0	1	0	0	A plus B	ADDS
1	1	0	0	1	0	0	1	1	1	A X or B	Not Used
1	1	0	1	1	0	1	1	1	1	A or B	OR
1	1	1	0	1	0	0	1	0	1	A plus B plus 1	INCR
1	1	1	1	1	0	1	1	1	1	A or B	Not Used

The Augend Bus and Addend Bus terminate as augend inputs and addend inputs to the Arithmetic Logic Unit. The function to be performed by the ALU is determined by the function select (FX0, FX1, FX2, FX3, FCIN and FINH) lines decoded by the function bus decoder.

4.1.5 I/O Bus

The I/O BUS consists of 16 data out lines (OTB00-I→OTB15-I) and 16 data in lines (INB00-I→INB15-I). The outbus lines are driven directly from the M-Register.

SCSW: The Select Console Switches logic gates console switches 0 - 15 to bits 0 - 15 of the IN Bus.

IOA77: The I/O Address 77 logic gates data in bits 0 - 7 of the OT Bus to bits 0 - 7 of the TTY controller at Data Transfer Out (DTO) time. At Data Transfer In (DTI) time, bits 0 - 7 of the TTY controller are gated to bits 0 - 7 of the IN Bus.

IOA76: The I/O Address 76 logic gates data contained in bits 2 - 7 of the M register to the priority interrupt expander arm register.

4.1.6 General Purpose Registers

The 8 foreground general-purpose registers (A, X, Y, Z, B, C, D, E) and the 8 background general-purpose registers (A', X', Y', Z', B', C', D', E') make a total of 16 registers available for general purpose use.

The group selection (foreground or background) is dependent upon the state of the foreground/background mode control flip-flop (FRGND). If the foreground/background flip-flop is set, the foreground registers are selected; if the flip-flop is reset the background registers are selected.

The 16 general purpose registers and their associated addressing and selection logic are implemented in matrix fashion with sixteen 16-bit scratch pad memory integrated circuit chips. This arrangement is one of many features which enable the SPC-16 to have broad functional capabilities in minimum space.

Functionally, these registers respond to instructions in the same manner as any other hardware register. The difference lies in the physical and electrical implementation methods. Figure 4-2 illustrates the organization of a typical bit of these registers.

Instructions which reference a general purpose register carry a corresponding 3-bit address within their formats. In addition, if indexing is required, a different two bits of the instruction register select one of three (X, Y, Z) operational registers.

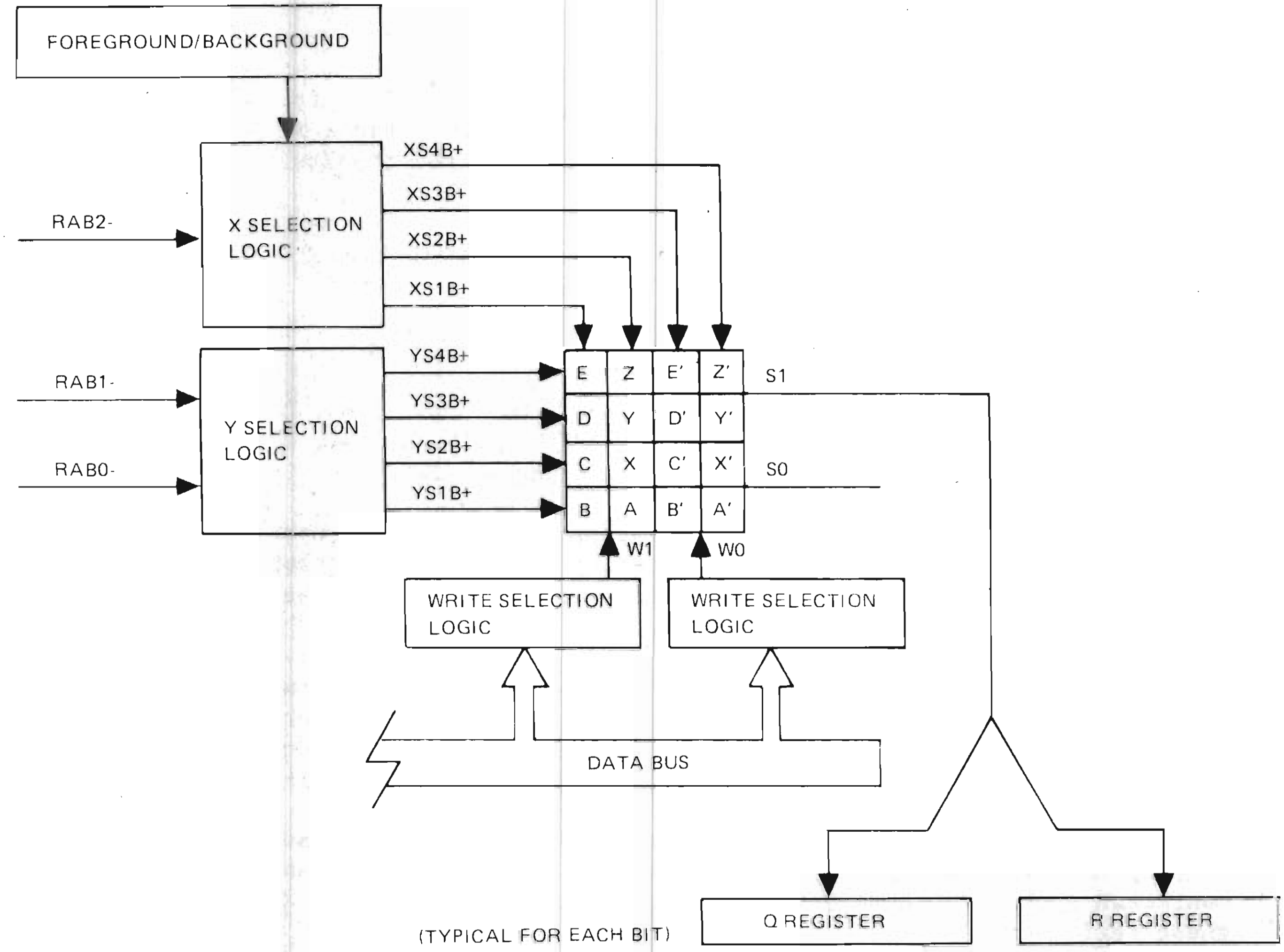


Figure 4-2. Operational Register Organization

As the execution of the instruction proceeds, the 3-bit address will be gated onto the 3-bit Register Address Bus (RAB0, RAB1, RAB2). The decode of the three register address bus lines and the state of the foreground mode control flip-flop decode one of a group (foreground/background) of 16 registers.

The X and Y lines are connected to the 16 bi-stable elements on each of the 16 scratch pad memory integrated circuit chips. If the instruction being executed requires a read from the specified register, activation of the selected X and Y lines selects the contents of one bit of each of the 16 scratch pad chips, which is then gated into the associated bit position of either the Q or R register.

If the instruction being executed requires a load into a register, the operation proceeds as before, except that the write selection logic will gate the contents of the Data Bus to appropriate positions in the matrix.

SECTION 5
TIMING

5.1 BASIC PROCESSOR TIMING

The execution of an instruction is really the execution of several small steps by the hardware in proper sequence at the proper time. In other words, to execute any instructions a 'micro program' must be performed by the hardware. This program must have various paths to be followed depending on the type of instruction, the addressing option selected by the programmer, the status of the SPC-16 interrupt system, and the kind of I/O in progress. Each hardware step of this 'microprogram' can be labeled in the following general form:

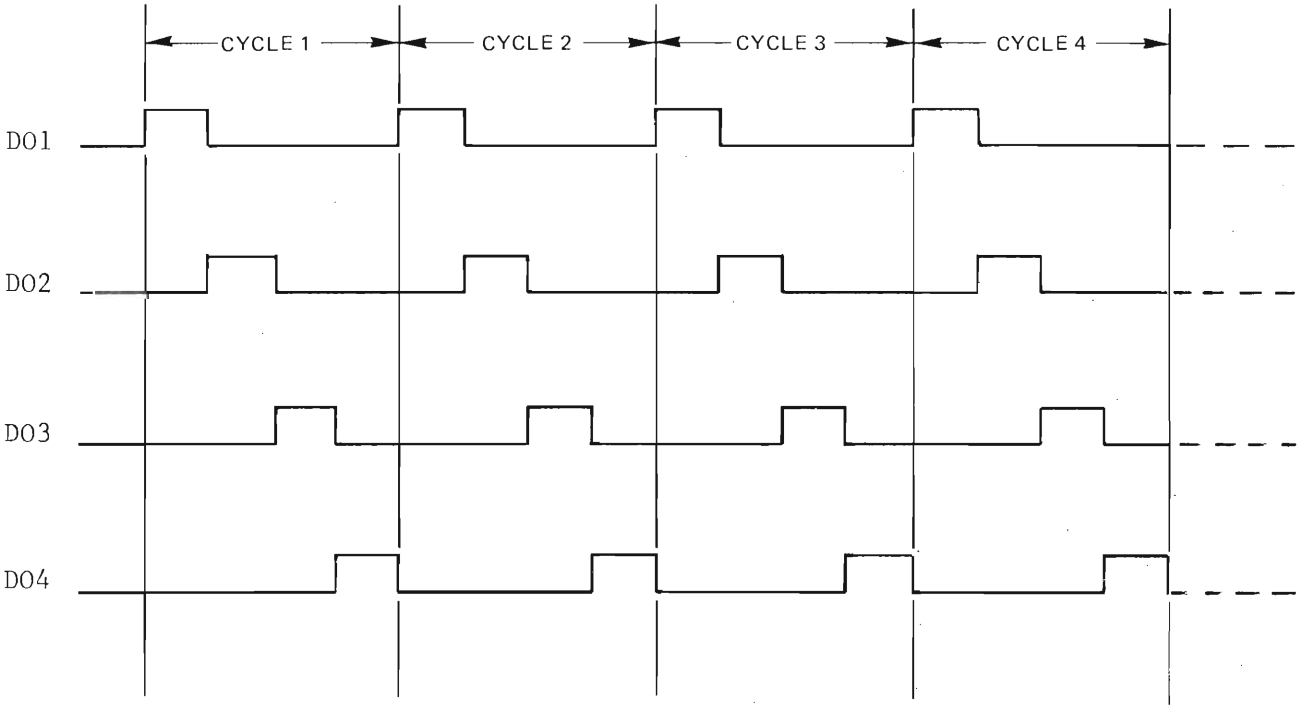
Sa DOb

where a = 1,2,3,4,5,6,7,8 or 9

and b = 1,2,3 or 4

S stands for Sequence State of the SPC-16. There are 9 possible sequence states, S1 through S9. Each sequence state usually takes 1 memory cycle.

DO stands for 'DO Time'. To properly time the execution of each sequence state, each computer memory cycle has been broken into quarters and these pulses are called DO1, DO2, DO3 and DO4. A rough timing diagram of DO times could be drawn as follows:



In addition, each DO time is divided into four DO clocks (DOC1, DOC2, DOC3 and DOC4).

The overall basic timing of the SPC-16 processor is shown in Figure 5-1.

5.1.1 Master Clock

The master clock oscillator is located on the Timing Control board. The worst case timing diagram in Figure 5-2 shows all timing signals referenced to MC+. The times listed are the worst case latest times that the signals could occur relative to MC+. The minimum times are not shown but are on the order of 2-5 ns. For ease of reading, only the 960 nanosecond cycle is shown.

5.1.2 DO Clocks

The DO clocks (DOC1, DOC2, DOC3 and DOC4) are nominally 60 nanoseconds wide, with each DOC clock triggered by the leading edge of MC+.

The DO times (DO1, DO2, DO3 and DO4) are also presented in Figure 5-2. A flowchart of the DO timing is shown in Figure 5-3.

The DO times progress through DO1, DO2 and DO3. If there is a request at DO3 for a DMA cycle, the sequencer goes to S8DO4. If there is no request for a DMA cycle, then any one of the conditions shown in Figure 5-3 may cause the DO timer to go back to DO2.

a. Shift right instruction and end shift right count not complete.

b. Extended processor option flip-flop set.

c. Multiply instruction and sequence state 9.

d. Non-memory reference instruction, instruction fetched from ROM, no power fail interrupt PFIR, RUN mode operational and no interrupt request.

If none of these conditions are present, then the DO4 time is entered.

5.1.3 Sequence States

In defining the Sequence States, the designers of the SPC-16 asked themselves this question: "What set of steps must the computer perform in order to execute every instruction in the repertoire under all possible conditions?". Then to each of these steps they associated a combination of flip-flops, some set and some reset. This combination they call a 'state' of the machine. Since the 'states' of the machine must be held for the proper duration of time and must be changed to the next proper 'state' at the proper instant of time (as triggered by DO and DOC pulses) the 'states' are called 'sequence states'. The Sequence States S1 through S9 will now be listed and the purpose of each one will be explained. (In most cases the computer will spend the four successive DO times, or 1 memory cycle, in each sequence state though there are exceptions to this.)

P2 72

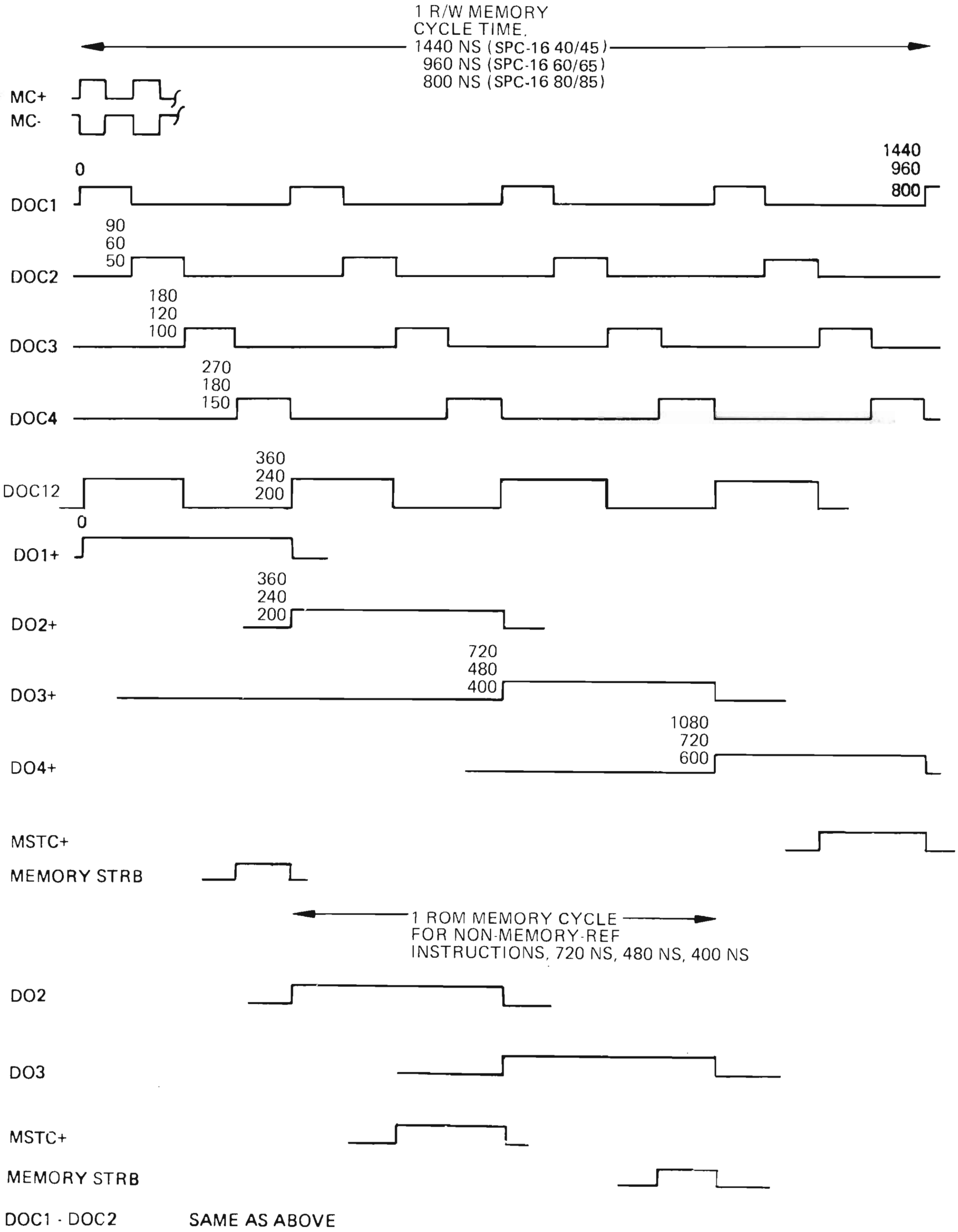


Figure 5-1. Basic Timing

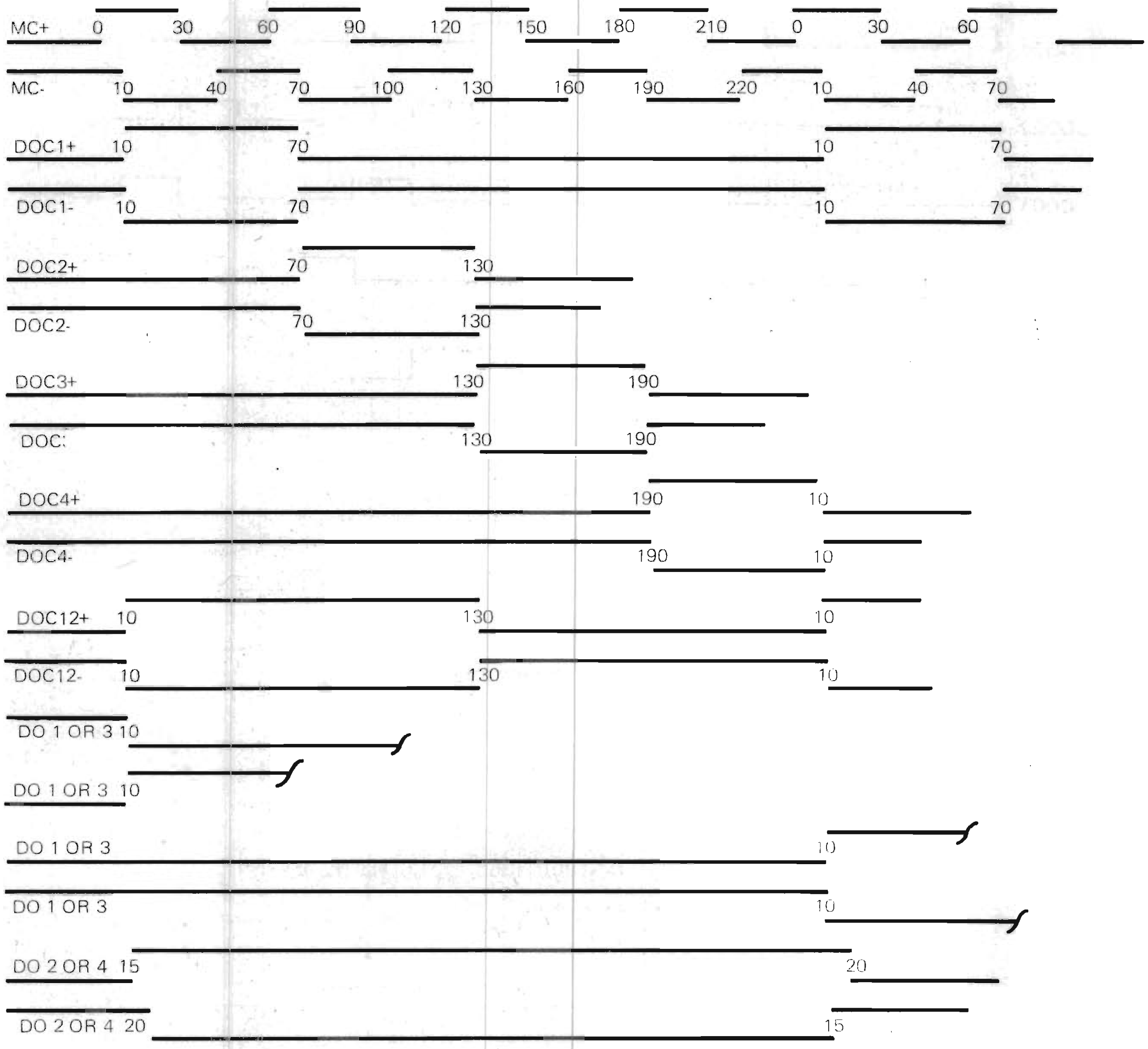


Figure 5-2. Basic Timing of Processor (DOC and DO Worst Case)

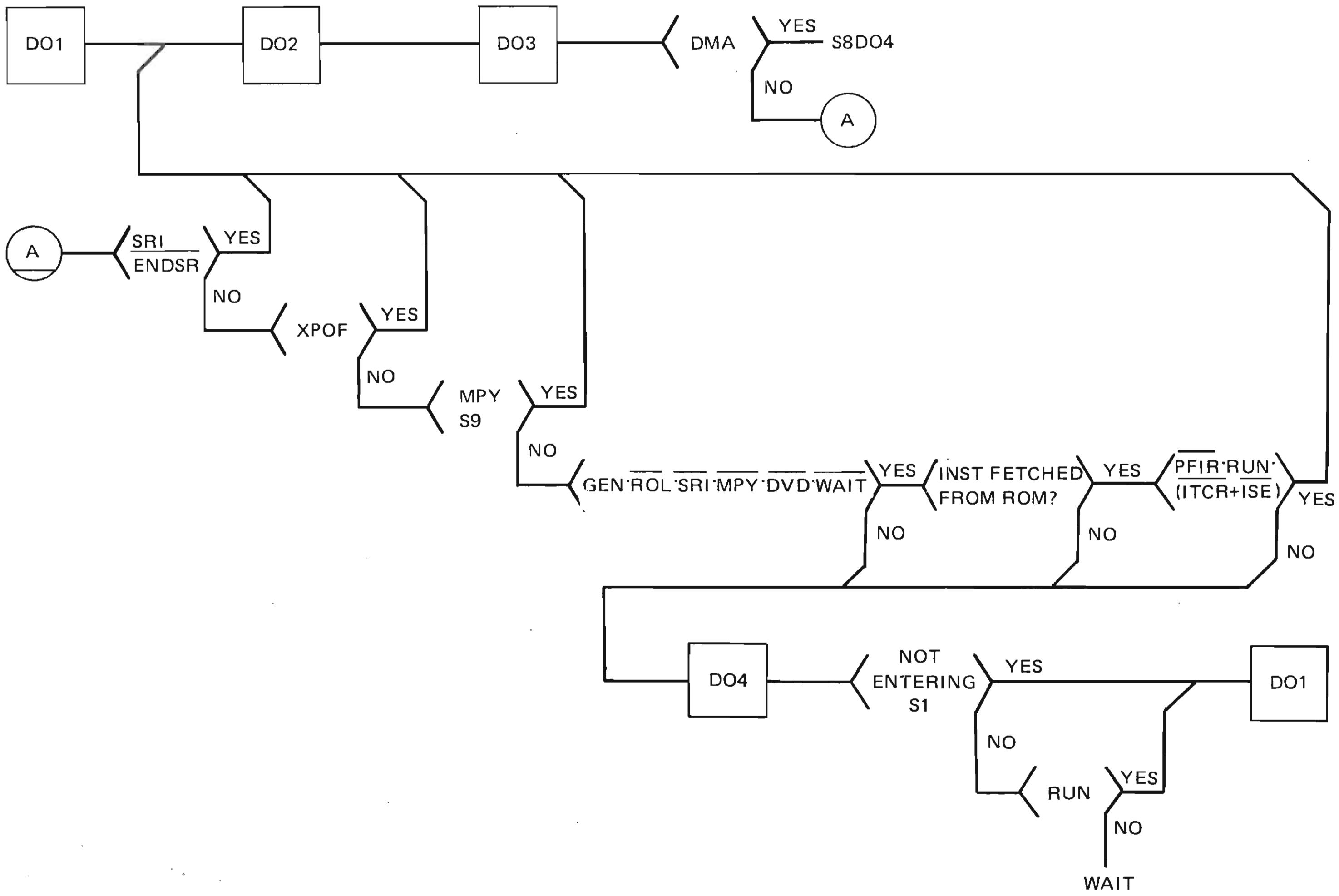


Figure 5-3. DO Timing Flowchart

S1: All instructions start at S1, DO1.

DO1: Fetch next instruction from memory. (Started by DO4 of last instruction.) Put into M reg. then into I reg. Memory access time is one DO time (i.e., 1/4 of a memory cycle).

DO2: P+1+ISE→P.
Decode I

DO3: Place displacement into W reg. (from M). (P+displacement if program relative.) Indexing is done at this point if the operand is not indirect.

DO4: If base relative, D+W→W.
One word instruction -
If direct go to S3 if not go to S5.
Two word instruction -
MRX - P→L go to S5.
Literal - P→L go to S5.

S2: Only used on two word MRX instructions.

DO1: Memory accesses second word of instruction.

DO2: P+1+ISE→P.

DO3: M+index (if any)→W

DO4: If base relative, D+W→W.
W→L.

S3: Indirect addressing state. (MR & MRX)

DO1 & DO2: Memory is accessed, producing indirect address.

DO3: Indirect address + index (if any)→ W

DO4: W→L

S4: Used only by LARS and SARS

DO1: Memory cycle initiated for loading or storing register.

DO2: Register↔M

DO3: W is incremented. Register count is incremented.

DO4: W→L, if register count ≠ 8 go back to DO1, otherwise go to S5.
(Status is handled in S5.)

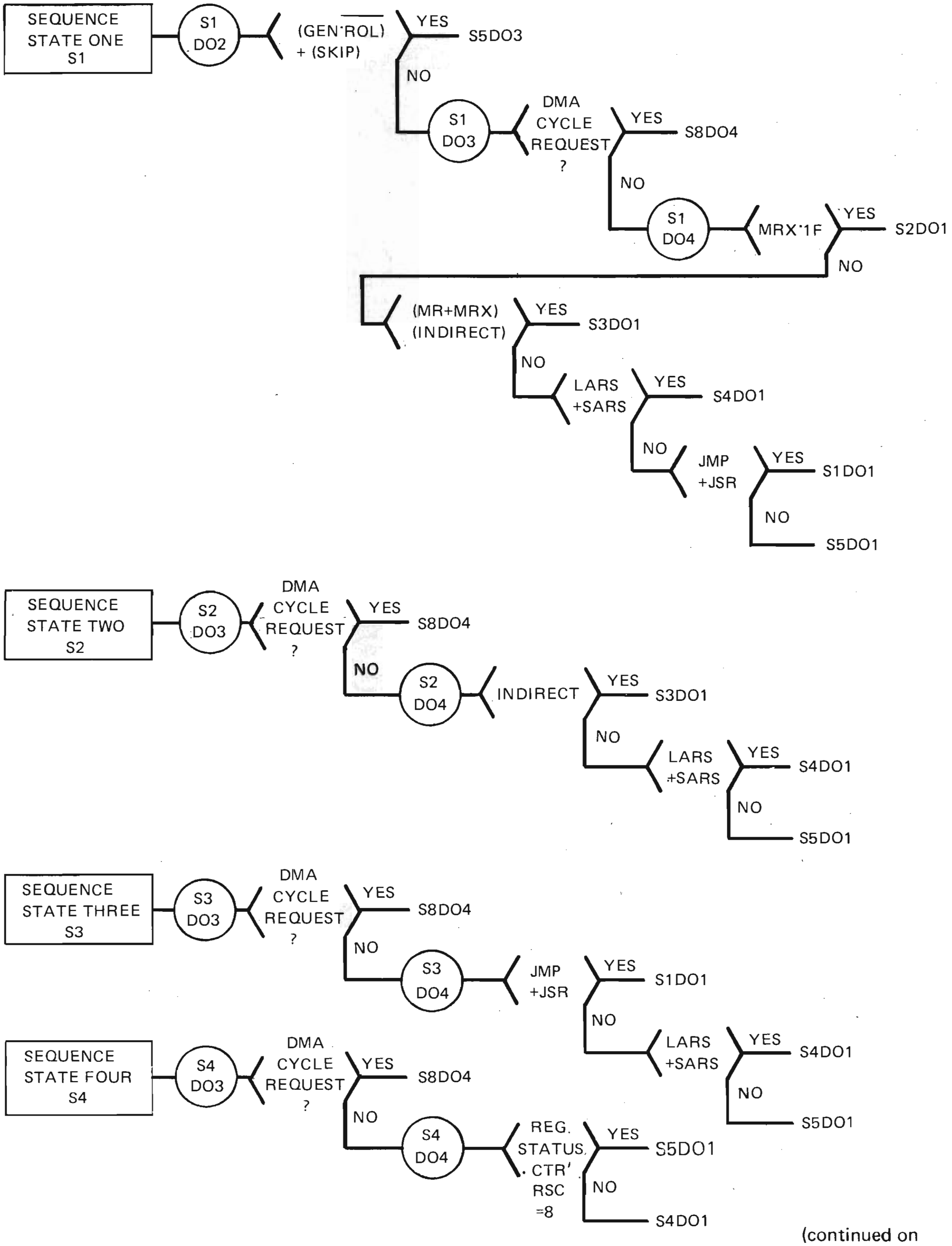
S5: Used by all instructions except JMP and JSR.

DO1: Nothing done.

DO2: Instruction execution is normally completed and indicators set (e.g. for LDR M→R, for SARS Status→M, etc.).

DO3: Normally, nothing is done.

DO4: If no interrupt, P→L, go to S1.
If interrupt, interrupt vector location→L go to S7.



(continued on
on next page)

Figure 5-4. Sequence State Timing Flowchart

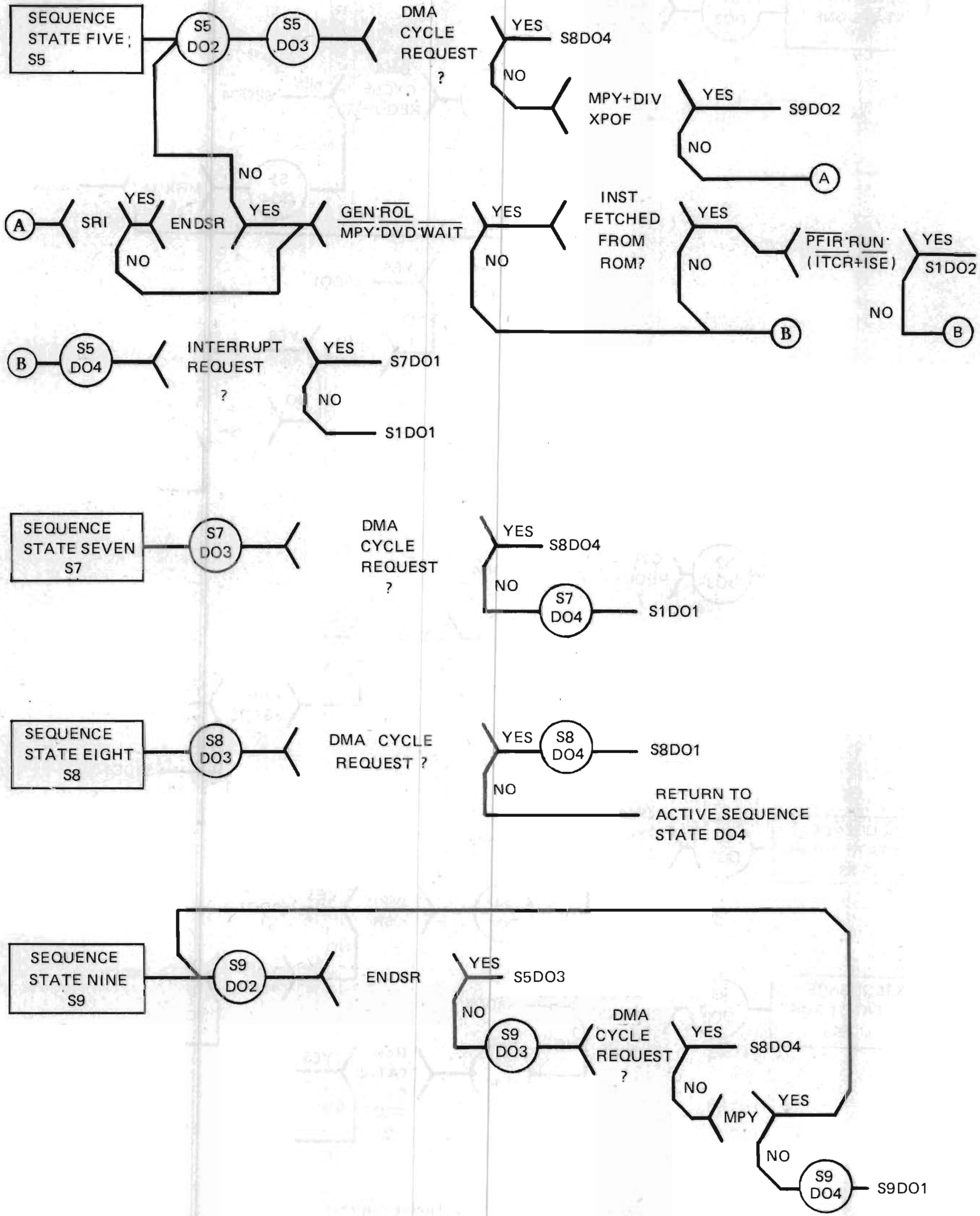


Figure 5-4. (cont'd.)

S6: For Micro-Instruction Logic.

A design feature for special instructions that a user may wish to add such as a Fast Fourier Transform. Otherwise S6 is never entered.

The customer should not attempt to make use of sequence state S6 without first consulting members of our engineering staff.

S7: Interrupt Sequence

DO1: Memory cycles accessing address at vector location.
DO2: P+ISE → E.
DO3: O → ISE
M → P
DO4: P → L

S8: DMA cycle steal, entered at DO4. S8 can occur after a DO3 in any state including S8. (This is why S8 terminates at DO3.)

DO4: DMA address → L
DO2: Memory access request.
DO2: M ↔ I/O
DO3: Return to active state.

S9: This state is used only with the hardware unsigned Multiply/Divide instructions. It allows the computer to perform the necessary number of add and shift operations. Algorithm for these instructions is given in Section 3.10 of SPC-16/40 Reference Manual.

Timing flowcharts for eight of the sequence states (S6 is omitted) are shown in Figure 5-4.

The sequence flip-flops are clocked by the sequence advance clock A (SACKA-) and sequence advance clock B (SACKB-). The SACKA- signal clocks flip-flops S3, S5, S6 and S9. Similarly, the SACKB- signal clocks flip-flops S2, S4 and S7. S1 state is controlled by both SACKA- and SACKB-. S8 state is clocked by DO3-. Figure 5-5 presents the sequence advance clock at DO4 worst case timing.

5.1.4 Combination Timing

The various combination timing signals are generally generated on the Timing Control board. A typical signal is defined below:

N812+ means Not S8 and S1 and DO2.

Another example is S1234+.

The definition of this signal is S1 or S2 or S3 or S4.

A similar signal such as S1233+ means S1DO3 or S2DO3 or S3DO3.

Refer to the signal index glossary in Appendix A for definitions of signal mnemonics.

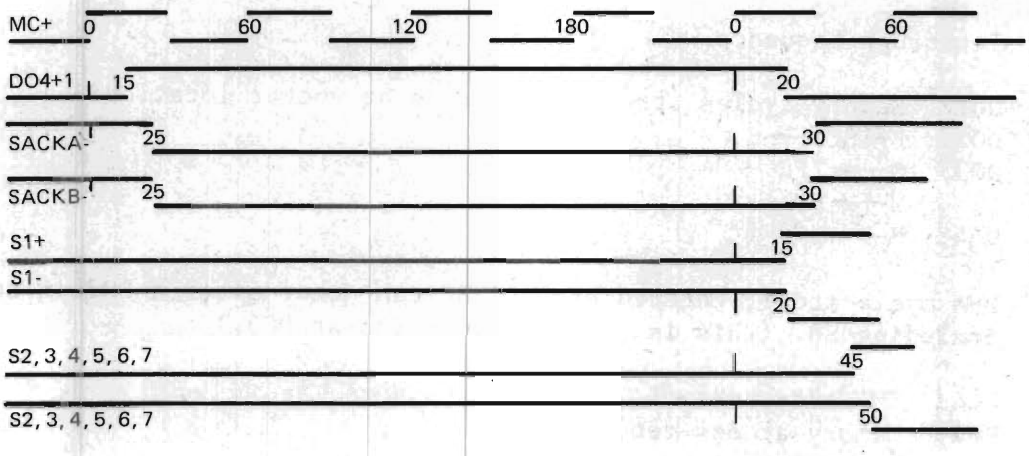


Figure 5-5. Sequence Advance Counter at DO4 (Worst Case Timing)

5.2 OPERATIONS TIMING

5.2.1 Memory Timing

A memory start clock signal is issued at the beginning of each instruction cycle.

If Read-Only-Memory is being accessed and the instruction is non-memory-reference, the next memory start clock will be issued at the end of DO2. If read/write memory (or ROM with memory reference) is being accessed, the next memory start clock will be issued at the end of DO4.

The timing associated with read/write memory is shown in Figure 5-6.

5.2.2 Systems Console

The timing of console operations such as STEP-RUN/IDLE and WAIT-STEP are shown in Figure 5-7.

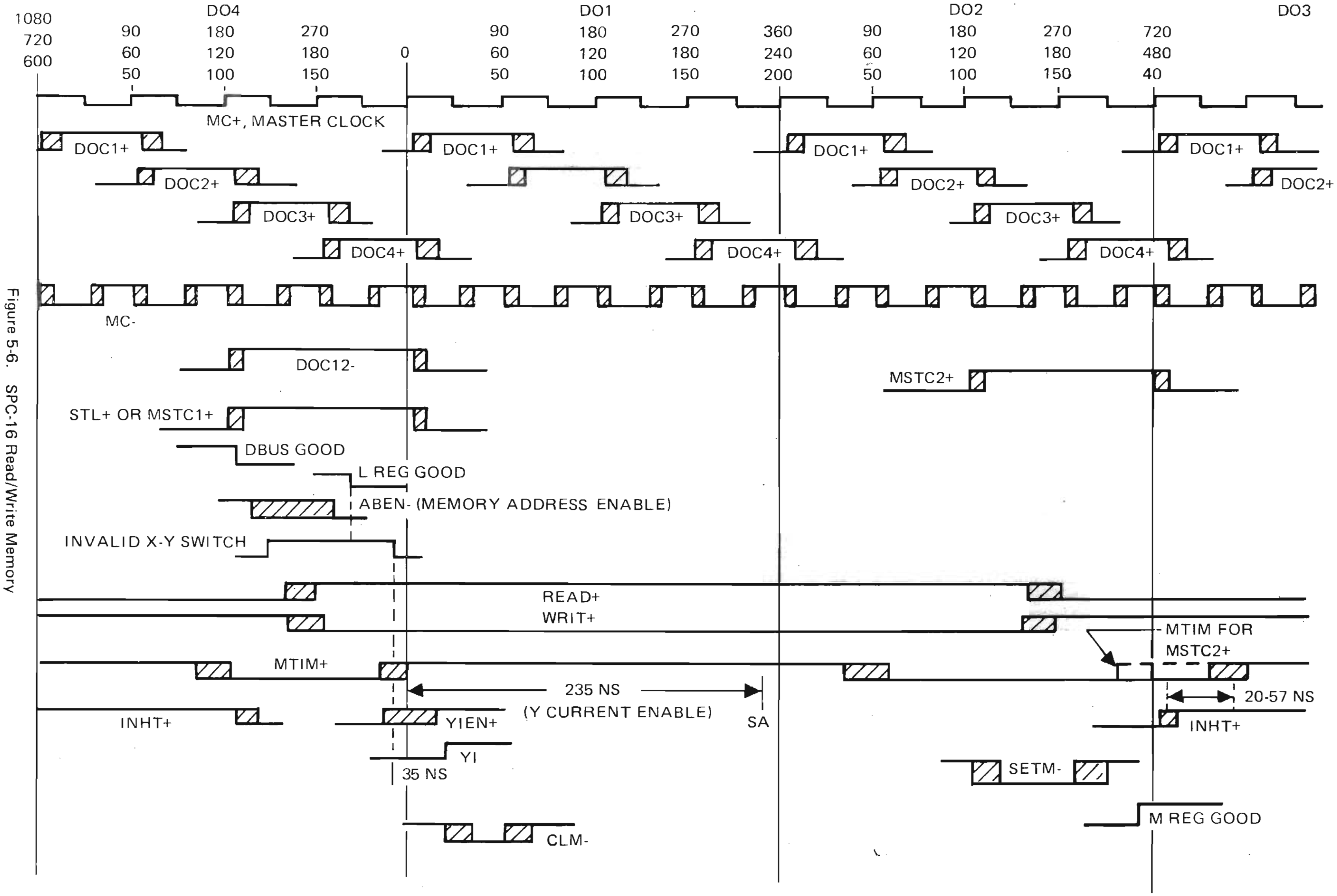


Figure 5-6. SPC-16 Read/Write Memory

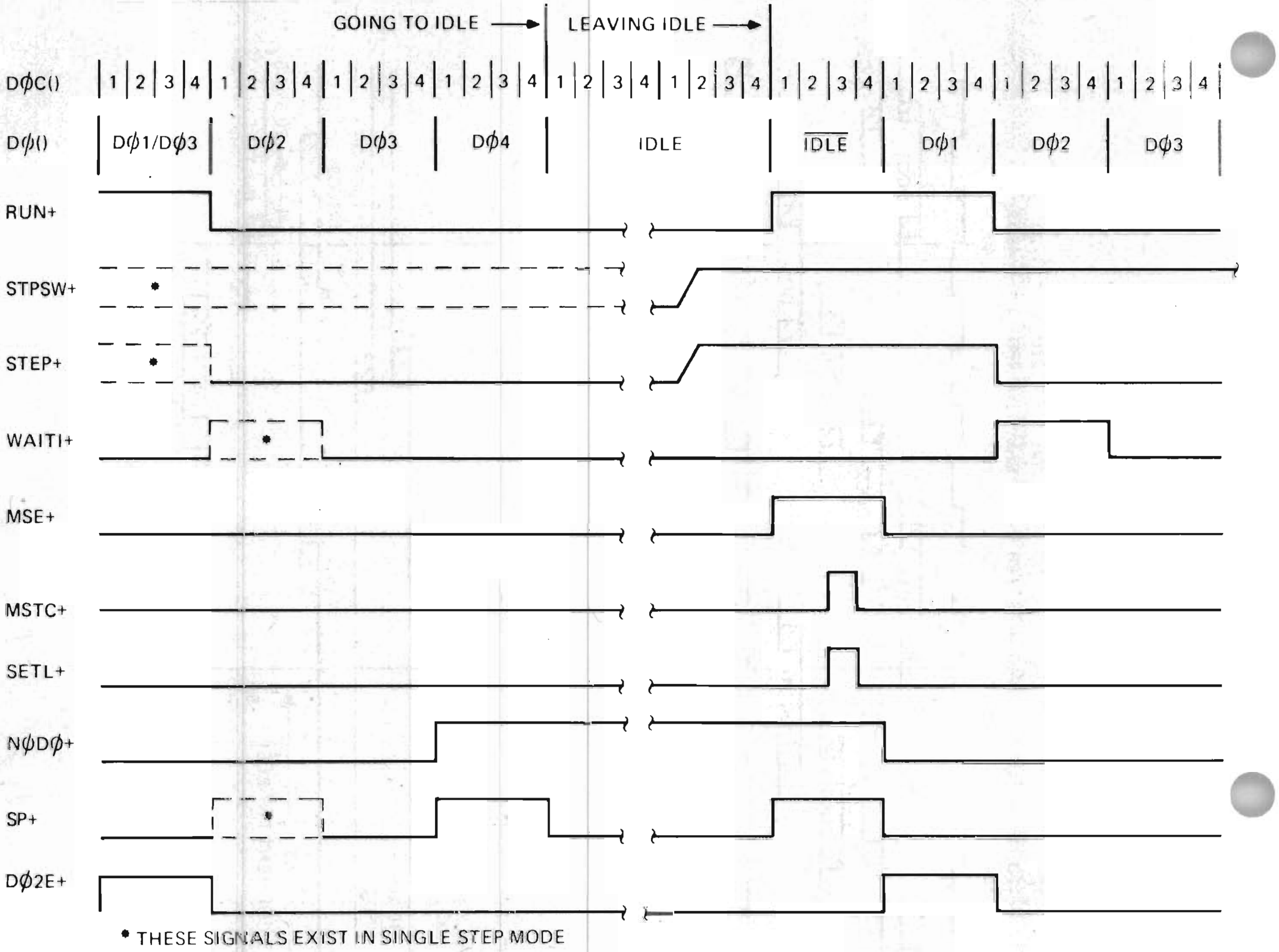


Figure 5-7. SPC-16 Step Timing - Idle Mode

5.2.3 Direct Memory Access/Arithmetic Logic Control I/O

The SPC-16 direct memory access port offers optional control lines that permit the I/O unit to perform arithmetic and logic operations between external data and the contents of memory. Each operation is performed in a single memory cycle time.

5.2.4 Serial I/O Interface

The serial I/O provides the interface between the serial controller and the teletype. The serial controller is located on the MIO board. The adjustment procedure for the serial controller 100-millisecond clock is described in the Teletype Test & Verify program manual (88A00185A).

5.2.5 Priority Interrupt

The SPC-16 priority interrupt system provides for 64 priority levels, six of which are internal. Each level has a dedicated memory location for the indirect jump vector.

The system as a whole is controlled by automatic and programmed control of the Interrupt System Enable (ISE) flip-flop. Also, each level may be individually masked for optimum control of the system. The interrupt system is described in Section 2 of the SPC-16/40+ System Reference manual 88A00243A.

5.2.6 Power Fail/Automatic Restart

Refer to Figure 5-8 for a diagram of the auto-restart timing. Power Fail/Auto Restart is explained in detail in Section 8.

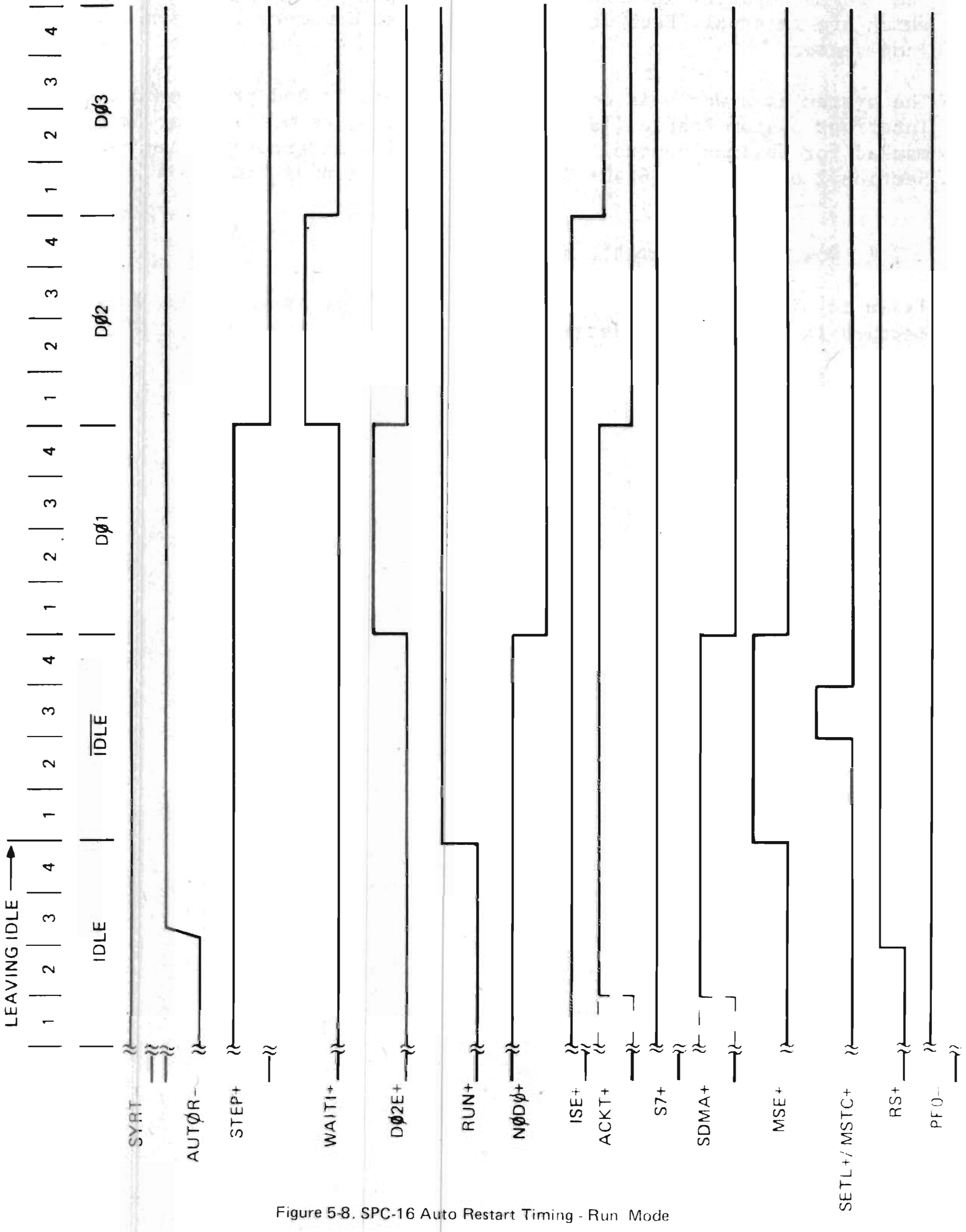


Figure 5-8. SPC-16 Auto Restart Timing - Run Mode

SECTION 6
INSTRUCTION SEQUENCING

6.1 BASIC DECODE

This section defines the operation codes and bit function assignments of the SPC-16 instruction set.

The memory (M) register is transferred to the I-register during the instruction fetch cycle. The I-register then drives the general decode logic shown in Figure 6-1. The main OP code (most significant 4 bits) is decoded into the main instruction classes. In addition, bits 8-11 are decoded into the sub-classes listed below:

a. GEN0 - Special GEN, includes WAIT, MPY and DIV.

b. ROLE - Register Operate Literal Enable.

c. SRLAE - Shift Right Logical and Count or Shift Right Arithmetic, Enable.

d. SRCCE - Shift Right Circular or Shift Right Circular Link, Enable.

e. CTRLE - Control Instruction Enable.

f. RCSE - Register Change Source Instruction Class Enable.

g. RCDE - Register Change Destination Instruction Class Enable.

h. RCOE - Register Change Operate Instruction Class Enable.

i. GEN - General Instruction Class.

j. XEC - Execute Instruction Decode.

k. RRRO - RO, RCO or ROL Instruction Group Decode.

l. SFIR - Select to ALU Function Bus from I-register.

The basic machine timing is shown in Figure 5-1 of the preceding section.

6.2 INSTRUCTION SEQUENCE CHARTS

Figure 6-2 is a summary of the SPC-16 instruction set.

The abbreviations used in the instruction summary are defined as follows:

DISP: Displacement; an absolute positive number defined by bits 0-9 for base relative memory reference instructions.

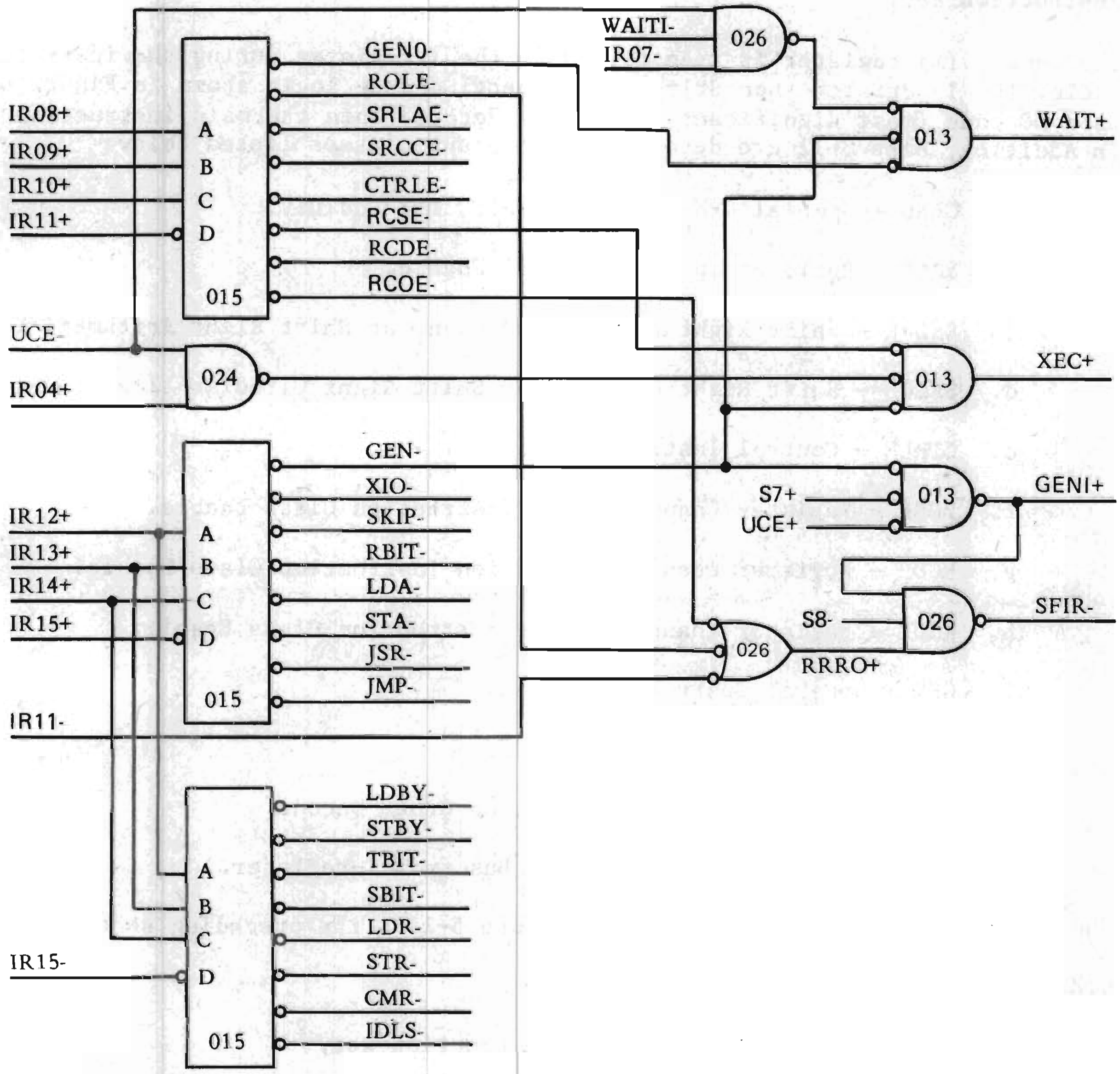
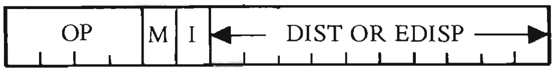


Figure 6-1. I Register General Decode

MEMORY REFERENCE

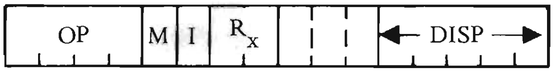


LDA	4
STA	5
JSR	6
JMP	7

DISP for M=1 (Base Relative)

EDISP for M=0 (Program Relative)

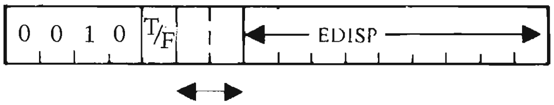
MEMORY REFERENCE WITH INDEX



LDR	C	REG
STR	D	REG
CMR	E	REG
LDBY	8	REG
STBY	9	REG
SBIT	B	Bit ID
RBIT	3	Bit ID
TBIT	A	Bit ID
LARS	F	1 0 0
SARS	F	1 1 0
INCM	F	0 0 0
DECM	F	0 1 0

M=1 (Base)
M=0 (Direct)

SKIP



0 0	OVFL
0 1	LINK
1 0	ZERO
1 1	PLUS

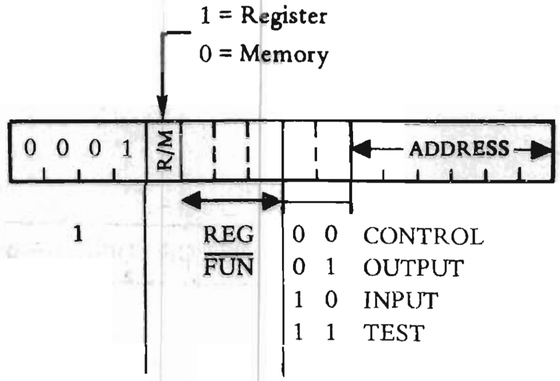
	SKIP FWD		SKIP REV	
SKOT	20	XX	21	XX
SKOF	28	XX	29	XX
SKS	2A	XX	2B	XX
SKR	22	XX	23	XX
SKZ	2C	XX	2D	XX
SKN	24	XX	25	XX
SKP	2E	XX	2F	XX
SKM	26	XX	27	XX

NOTE: SKIP REV XX is in 2's complement form.

Figure 6-2. SPC-16 Instruction Summary Hexadecimal Coding

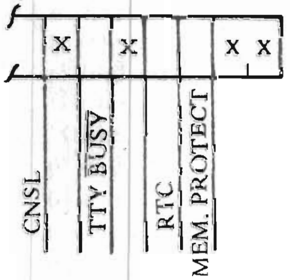
XIO

CTRL
DTOM/DTOR
DTIM/DTIR
TEST



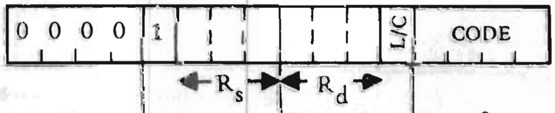
INTERNAL I/O

ENABLE INTERRUPTS	1	0/1	R	7	E
TTY OUTPUT	1	0/1	R	7	F
TTY INPUT	1	0/1	R	B	F
TTY TEST	1		0	F	F
TTY TRANSMIT	1		0	3	F
TTY RCV ONLY	1		2	3	F
TTY RCV/ECHO	1		4	3	F
TTY BREAK	1		6	3	F
RCSR/RCSM	1	0/1	R	B	E



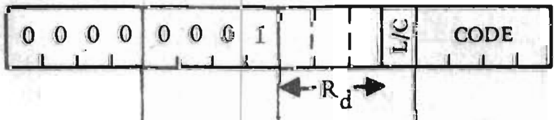
INTERRUPT MASK BITS

REGISTER OPERATE



0	$8+R_s$	$2R_d+1 = L$	9	ADD, ADDC
		$2R_d = C$	6	SUB, SUBC
			7	AND, ANDC
			8	XOR, XORC
			D	OR, ORC
			5	RTR

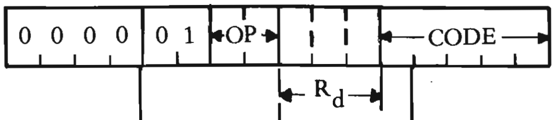
REGISTER OPERATE LITERAL



0	1	$2R_d+1 = L$	9	ADDV, ADDVC
		$2R_d = C$	6	SUBV, SUBVC
			7	ANDV, ANDVC
			8	XORV, XORVC
			D	ORV, ORVC
			5	LDV

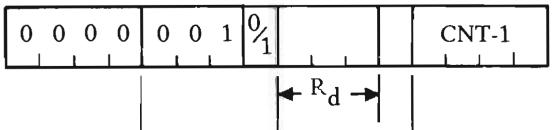
Figure 6-2 (continued)

REGISTER CHANGE



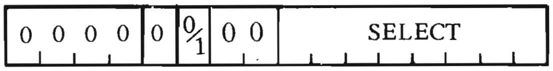
XEC	0	5	$2R_d+1$	0
RISE	0	5	$2R_d$	1
RTRN	0	5	$2R_d$	3
DSPL	0	5	$2R_d$	4
TRS	0	5	$2R_d$	8
ZERO	0	6	$2R_d$	0
ZRBY	0	6	$2R_d$	1
ZLBY	0	6	$2R_d$	2
EXBY	0	6	$2R_d$	4
TSR	0	6	$2R_d$	8
RCSW	0	6	$2R_d+1$	0
CMPL	0	7	$2R_d$	0
RLK	0	7	$2R_d$	1
DECR	0	7	$2R_d$	2
ADDS	0	7	$2R_d$	B
INCR	0	7	$2R_d$	E

SHIFTS



SRLC	0	2	$2R_d$	CNT-1
SRA	0	2	$2R_d+1$	CNT-1
SRC	0	3	$2R_d$	CNT-1
SRCL	0	3	$2R_d+1$	CNT-1

CONTROL



INE	0	4	0	3
INH	0	4	0	2
FMS	0	4	0	C
BMS	0	4	0	8
LKS	0	4	3	0
LKR	0	4	2	0
PMA	0	4	4	0
SYNC	0	4	8	0
WAIT	0	0	0	X
MPY	0	0	8	n
DIV	0	0	A	n

Figure 6-2 (continued)

EDISP: Extended Displacement; a signed number where the most significant bit of EDISP is extended up to bit 15. Used in program relative memory reference instructions and skip instructions.

R_x: Index register (X, Y, Z) specified by the binary decode of bits 8 and 9.

REG: General Purpose register specified by bits 8, 9 and 10 for XIO.

MODE: During an XIO instruction, the decode of bits 6 and 7 designate the I/O mode which is Control, Output, Input or Test type instruction.

M/R: Memory/Register select (for XIO instructions).

The symbols and terms used in the instruction sequence charts are defined in **Tables** 6-1 and 6-2, respectively.

Table 6-1. Symbol Definition

Symbol	Definition
←	is replaced by; that is, the term on the left side of the arrowhead is replaced by the term on the right side.
+	is added to
< >	Base Relative Function
( )	The contents of
/ /	Indexing function
,(comma)	and also
=	is equal to
≠	is not equal to
Q/2	Shift right one place

Table 6-2. Term Definition

Term	Definition
A	A-Register (Scratch Pad)
B	B-Register (Scratch Pad)
C	C-Register (Scratch Pad)
COUT	Carry Out
D	D-Register (Scratch Pad)
DB	Data Bus
DISP	Displacement
DREQ	Cycle-Steal Request
DTIM	Data Transfer Into Memory
DTIR	Data Transfer Into Register
DTOM	Data Transfer Out of Memory
DTOR	Data Transfer Out of Register
E	E-Register (Scratch Pad)
EA	Effective Address
EDISP	Extended Displacement
ENDSR	End Shift
FRGND	Foreground
I	Indirect Address
I/O	Input/Output Bus
IOT	I/O Test Flip-Flop
IREQ	Interrupt Request
ISE	Interrupt System Enable Bit
L	L-Register

Table 6-2. (continued)

Term	Definition
M	M-Register
P	P-Register
Q	Q-Register
R	R-Register
Rd	Destination Register (Scratch Pad)
Rs	Source Register (Scratch Pad)
rsc (subscript)	Register and Status Counter
RSC	Register and Status Counter
Rx	Index Register
S	Sequence State
SC	Shift Counter
SKOF	Skip on Overflow False
SKOT	Skip on Overflow True
SRI	Shift Instruction
TURN	Indicates fast (overlapped) ROM processing, i.e., the sequencer goes from S5·D03 to S1·D02.
W	W-Register
XPOF	Extended Processor Option Flip-Flop
ZP	Zero and Plus Indicator
ZPL	Zero, Plus and Link Indicators
ZPOL	Zero, Plus, Overflow and Link Indicator

In the sequence charts (see Figure 6-3 for example), the small boxes represent the scratch pad functions. The larger boxes generally represent ALU functions. The chart is divided so that the first small box occurs at DOC1, DOC2 of the DO state. There are nine sequence states (see Section 5.1.3) each representing one machine (memory) cycle: each sequence state is divided into four DO states (DO1, DO2, DO3, DO4); each DO state is divided into four DOC states (DOC1, DOC2, DOC3, DOC4). The DOC's are driven directly from the master clock. Normally, the set function for the first small box (looking from left to right) occurs at DOC2; in the large box, the set functions normally occur at the end of DOC3. In the second small box, the set function normally occurs at DOC4.

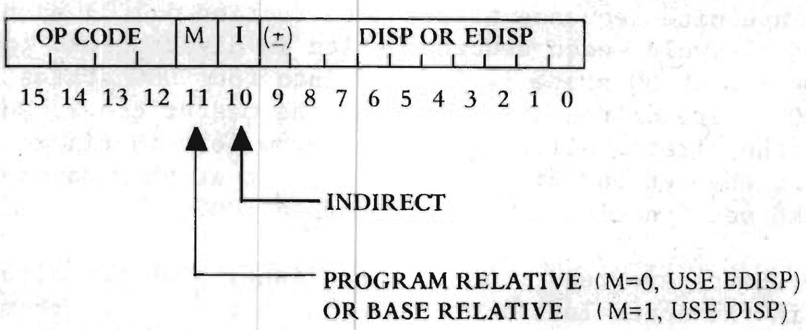
The sequence charts are read from left to right, starting with S1DO1 (row S1, column DO1) in the upper left corner of the chart. After progressing across sequence state 1 (S1) through S1DO4, sequence state 2 (S2) is read from left to right starting at S2DO1. The remainder of the chart is read in the same manner. Sequence states that are not applicable to the particular instruction being represented by the chart are disregarded. Thus, in the chart representing the LDA instruction (see Figure 6-3), sequence state 1 (S1) would be read from left to right, then sequence state 3 (S3) would be read, and finally, sequence state 5 (S5); the remaining sequence states would be disregarded because they are not used by the LDA instruction.

The instruction sequences shown in this section may be altered in the RUN mode if the dynamic SAVI switch (on the rear of the console board) is engaged. In troubleshooting a given instruction, therefore, it may be advisable to repeatedly execute a malfunctioning instruction by coding a test loop involving the instruction followed by a JMP $-1 instruction.

The instruction sequence charts do not show exit to S8 if there is a DMA request (which happens between DO3 and DO4). Nor is exit to an interrupt (at S5DO4) shown. Section 6.15 describes sequencing for these conditions. Just a normal sequence of events that occur during an instruction is shown.

Also not shown on the sequence diagrams, for purposes of clarity, are memory start clock, clear-M-register and strobed memory (sense-amp) output. A memory start clock occurs in <u>every</u> sequence state at approximately DO4·DOC3 unless explicitly stated otherwise. Every time a memory-start-clock occurs in DO4, a clear-M-register signal occurs in the following DO1 just prior to the forthcoming memory sense-amp output (Double-Rail RESET-SET on the M-register). One must keep these memory-related signals in mind when reading the sequence charts.

6.3 MEMORY REFERENCE INSTRUCTIONS (MR)



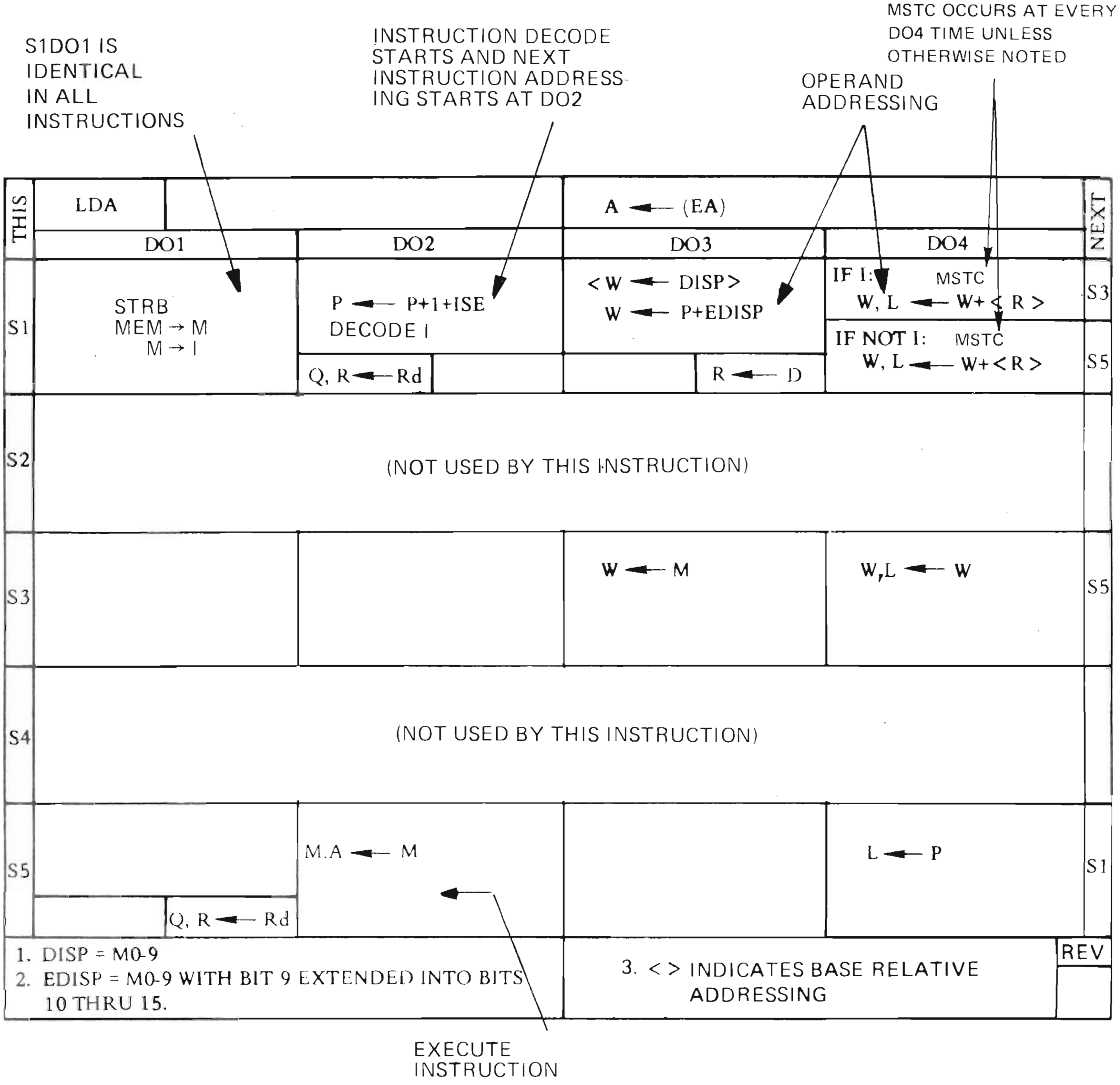
The op codes of the instructions in the memory reference group are shown below. Each instruction in this group is described in detail on the following pages.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	M	I	–	–	–	–	–	–	–	–	–	–
LDA	0	1	0	0												
STA	0	1	0	1												
JSR	0	1	1	0												
JMP	0	1	1	1												

DISP OR EDISP

6.3.1 Load Register A (LDA)

The contents of the location as specified by the EA replaces the contents of the A-register. Refer to Figure 6-3 for the instruction sequence chart associated with the LDA instruction. The LDA instruction is discussed in detail in the following paragraphs.



S1DO1 IS IDENTICAL IN ALL INSTRUCTIONS

INSTRUCTION DECODE STARTS AND NEXT INSTRUCTION ADDRESSING STARTS AT DO2

OPERAND ADDRESSING

MSTC OCCURS AT EVERY DO4 TIME UNLESS OTHERWISE NOTED

THIS	LDA		A ← (EA)		NEXT
	DO1	DO2	DO3	DO4	
S1	STRB MEM → M M → I	P ← P+1+ISE DECODE I Q, R ← Rd	<W ← DISP> W ← P+EDISP R ← D	IF I: MSTC W, L ← W+<R>	S3
				IF NOT I: MSTC W, L ← W+<R>	S5
S2	(NOT USED BY THIS INSTRUCTION)				
S3			W ← M	W,L ← W	S5
S4	(NOT USED BY THIS INSTRUCTION)				
S5	Q, R ← Rd	M.A ← M		L ← P	S1

1. DISP = M0-9
2. EDISP = M0-9 WITH BIT 9 EXTENDED INTO BITS 10 THRU 15.
3. < > INDICATES BASE RELATIVE ADDRESSING

REV

EXECUTE INSTRUCTION

Figure 6-3. LDA Instruction Sequence Chart

Sequence State One: In sequence state 1 (S1) the following events occur:

S1DO1: The contents of memory is loaded into the M-register near the end of DO1.

S1DO2: Q, R ← Rd, P ← P+1+ISE. The destination register (Rd) decoded by the bits 5, 6 and 7 of the M-register is read out. It won't be used in this case because we are only concerned with the A-register. The Rd register contents are transferred to the Q-register and the R-register.

Set Q (STQ) and set R (STR) occur at DOC2. The P counter is incremented through the ALU and loaded into P along with ISE.

S1DO3: Base Relative < W ←DISP > If bit 11 is a one, base relative addressing is specified and the addend bus contains the displacement (bits 0-9) that is loaded into the W-register.

$$\underbrace{\overbrace{\text{SM9, SM58, SM04}}^{\text{Addend (Displacement)}} \\ W \leftarrow \text{DISP}}$$

A set W (STW) signal occurs at the end of DOC3 and places the displacement into the W-register.

Program Relative: W ← P+EDISP. If bit 11 is a zero, program relative addressing is specified and the augend bus contains the contents of the P-register and the addend bus contains the extended displacement.

$$\overbrace{\text{SLP}}^{\text{Augend}} + \overbrace{\text{SMX9, SM9, SM58, SM04}}^{\text{Addend (Extended Displacement)}}$$

A set W (STW) signal occurs at end of DOC3 and this places P+EDISP into the W-register.

Base Relative or Program Relative: At DOC3/4 the contents of the D-register replace the contents of the R-register.

S1DO4: Base Relative: W, L ← W+(R). The contents of the W-register are selected onto the augend bus (SW) and the R-register is selected onto the addend bus. The sum (W+R) is placed in the W-register (STW) and the L-register (STL) and a memory start clock (MSTC) is initiated. This gives the first effective working address. The next sequence state is S3 if the instruction is indirect, otherwise the next sequence state is S5.

Program Relative: W, L ← W. The contents of the W-register are selected onto the augend bus, sent through the ALU and replace the contents of the W-register and L-register. This gives the effective working address. The next sequence state is S3 if the instruction is indirect, otherwise the next sequence state is S5.

Sequence State Three: In sequence state 3 (S3), the following events occur:

S3D01: The contents of memory is loaded into M-register near the end of D01.

S3D02: Nothing happens.

S3D03: W ← M. The indirect address contained in the M-register replaces the contents of the W-register.

S3D04: W, L ← W. The contents of W are selected onto the augend bus (SW), sent through the adder and replace the contents of the W and the L registers. The indirect address is now contained in the L-register, the memory location register and a MSTC is initiated. The next sequence state is S5.

Sequence State 5 (S5): The following events occur in S5.

S5D01: Q, R ← Rd (always happens, even though it is not required by LDA). At DOC3/4, the contents of the destination register replace the contents the Q and R-registers.

S5D02: A ← M. The memory is now available and the contents of the M-register (data) are selected onto the addend bus by SMMS6, SM58, SM04 and loaded into the scratch pad memory by selecting register A.

S5D03: Nothing happens.

S5D04: L ← P. The contents of the P-register replace the contents of the L-register and a MSTC is initiated. The sequencer then goes back to S1.

6.3.2 Store Register A (STA)

The contents of the A-register replace the contents of the memory location specified by the effective address. Refer to Figure 6-4.

Sequence states S1 and S3 are identical to the LDA instruction (Figure 6-3).

AT S5D02 time the contents of the R-register replace the contents of the M-register. This is accomplished by selecting R (SRRR, SRLL) into the addend and transferring through the ALU to the DBUS. STM occurs at DOC3 of D02 to complete the transfer. This places the contents of the A-register into the M-register prior to writing it into memory at D03/4 time (write cycle time). S5D04 is identical to the LDA instruction.

THIS	STA		(EA) ← A		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE
		Q, R ← Rd
			< W ← DISP >
			W ← P+EDISP
			R ← D
				IF I:
				W, L ← W+ <R>
				IF NOT I:
				W, L ← W+ <R>
					S3
					S5

S2					
S3			W ← M	W, L ← W	S5
S4					
S5	Q, R ← A	M ← R		L ← P	S1

1. DISP = M0-9
2. EDISP = M0-9 WITH BIT 9 EXTENDED INTO BITS 10 THRU 15.

3.<>INDICATES BASE RELATIVE ADDRESSING

REV

Figure 6-4. STA Instruction Sequence Chart

6.3.3 Jump to Subroutine (JSR)

The contents of the program counter plus one replace the contents of the P and E-registers, bits 0-15. The effective address replaces the contents of the program counter and the interrupt system enable is turned off, inhibiting interrupt requests from being acknowledged. This instruction cannot be used in location X'7FFF'. Refer to Figure 6-5. The differences in this instruction with respect to LDA and STA are:

a. At the start of S1D03, a zero is placed in the ISE indicator.

b. In S1D04, if not indirect P, W, L ← W+(R), then the sequencer goes back to S1D01.

JSR — JSR IS NOT INTERRUPTABLE

THIS	DO1	DO2	DO3	DO4	NEXT
S1		E,P ← P+1+ISE	< W ← DISP > W ← P+EDISP ISE ← 0	IF I: W,L ← W+ <R>	S3
		Q,R ← Rd	R ← D	IF NOT I: P,W,L ← W+ <R>	S1
S2					
S3			W ← M	P,W,L ← W	S1
S4					
S5					

1. DISP = M0-9
2. EDISP = M0-9 WITH BIT 9 EXTENDED INTO BITS 10 THRU 15.

3.<>INDICATES BASE RELATIVE ADDRESSING

REV

Figure 6-5. JSR Instruction Sequence Chart

c. In S3DO4, if indirect, P, W, L ← W places the new program location in the memory location (L) register and in the P-register.

6.3.4 Jump Unconditional (JMP)

The effective address replaces the contents of the P-register. This instruction is identical to the JSR instruction except that in S1DO2, the P-register does not transfer to the E-register and the ISE indicator is not affected in S1DO3. Refer to Figure 6-6.

THIS	JMP	JMP is not interruptable			NEXT
	DO1	DO2	DO3	DO4	
S1		P ← P+1+ISE	<W ← DISP> W ← P+EDISP	if I: W,L ← W+ <R>	S3
		Q,R ← Rd	R ← D	if not I: P,W,L ← W+<R>	S1
S2					
S3			W ← M	P,W,L ← W	S1
S4					
S5					

1. DISP = M0-9
2. EDISP = M0-9 with bit 9 extended into bits 10 thru 15.

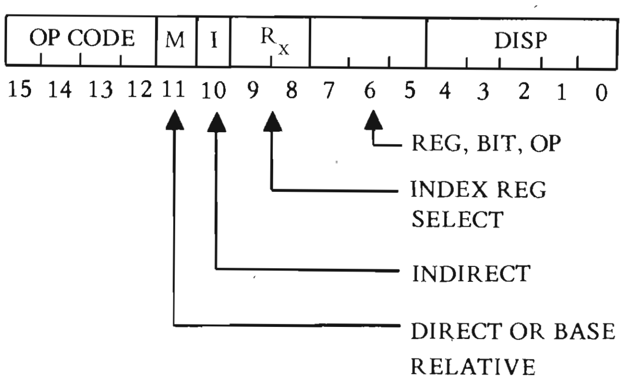
3.<>INDICATES BASE RELATIVE ADDRESSING

REV

Figure 6-6. JMP Instruction Sequence Chart

6.4 MEMORY REFERENCE WITH INDEXING INSTRUCTIONS (MRX)

The 12 instructions in this group are described in the order shown below.



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	M	I	X	X	—	—	—	—	—	—	—	—
LDR	1	1	0	0					Rd	Rd	Rd					
STR	1	1	0	1					Rd	Rd	Rd					
CMR	1	1	1	0					Rd	Rd	Rd					
LDBY	1	0	0	0					Rd	Rd	Rd					
STBY	1	0	0	1					Rd	Rd	Rd					
INCM	1	1	1	1					0	0	0					
DECM	1	1	1	1					0	1	0					
LARS	1	1	1	1					1	0	0					
SARS	1	1	1	1					1	1	0					
SBIT	1	0	1	1					B	I	T					
TBIT	1	0	1	0					B	I	T					
RBIT	0	0	1	1					B	I	T					
												← DISP →				

6.4.1 Load Register (LDR)

The contents of the location specified by the effective address replace the contents of the selected (destination) register. Refer to Figure 6-7.

THIS	LDR		Rd ← (EA)		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I:
			W ← DISP+/Q/
				IF DISP = 1F:
				W, L ← P
					S2

				IF I & DISP ≠1F W,L ← W+<R>	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠1F W,L ← W+<R>	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+/Q/
				IF I:
				W,L ← W+ <R>
					S3

				IF NOT I: W,L ← W+ <R>	S5
S3			W ← M+/Q/	W,L ← W	S5
S4					
S5	Q,R ← Rd	Rd ← M		L ← P	S1

1. DISP = M0-4
2.<>INDICATES BASE RELATIVE ADDRESSING
3. // INDICATES INDEXING

REV

Figure 6-7. LDR Instruction Sequence Chart

S1D02: Q, R ← Rd; P ← P+1+ISE, Q ← Rx. At DOC1/2 the destination register specified by bits 5, 6 and 7 replaces the contents of the Q and the R-registers. P+1+ISE is put into the P-register. At DOC3/4 the index (Rx) register specified by bits 8 and 9 of the instruction replaces the contents of the Q-register in anticipation of indexing.

S1D03: If bit 10 is a one, indirect addressing is specified, if it is a zero it is not indirect. At DOC3/4 the base relative register (D) is read into the R-register in anticipation of base relative modification during D04.

Indirect: W ← DISP. The displacement (M-register bits 0-4) is selected onto the addend bus, sent through the ALU and replaces bits 0-4 of the W-register. Zeroes are placed into bits 5-15 of the W-register.

Direct: W ← DISP+/Q/./Q/ indicates the instruction is indexed. The contents of the Q-register are selected onto the augend bus and the displacement (M-register bits 0-4) are selected onto the addend bus. The sum replaces the contents of the W-register.

S1D04: If DISP = 1F, W, L ← P and S2 is the next sequence state. If indirect and DISP ≠ 1F, W, L ← W+(R) and S3 is next. If not indirect and DISP ≠ 1F, W, L ← W+(R) and S5 is next. The base relative register is added to the W-register only if base relative addressing is specified.

S2D02: P ← P+1+ISE. P is incremented again (P+1) because the effective address was at the next memory location. The ISE indicator is also sent through the adder and goes to the ISE flip-flop.

S2D03: The instruction is again checked to see if it is direct or indirect.

Indirect: W ← M. The contents of the M-register replace the contents of the W-register.

Direct: The index register (in Q) is selected (indexing only) onto the augend bus and the M-register is selected onto the addend bus. The sum replaces the contents of the W-register.

S2D04: W, L ← W+(R). The contents of the W-register are added to the base relative register (if specified) and replace the contents of the W and the L-register. If indirect addressing is specified the next sequence state is S3. If it is a direct instruction, the next sequence state is S5.

S3D03: W ← M+/Q/. The index register (Q) is added (indexing only) to the M-register and the sum is placed in W.

S3D04: W, L ← W. The indirect address in W is then sent through the adder to the memory location (L) register and the W-register. The sequencer then goes to S5.

S5DO1: Q, R ← Rd. The decoded destination register (Rd) is sent to the Q and the R-registers even though this function is not required by the LDR instruction.

S5DO2: Rd ← M. The contents of the M-register replace the contents of the decoded destination register.

S5DO4: L ← P. The P counter replaces the contents of the L-register and the sequencer goes back to S1.

6.4.2 Store Register (STR)

The contents of the selected register replace the contents of the location specified by the effective address. The STR instruction is identical to the LDR instruction except that at S5DO2, the contents of the R-register replace the contents of the M-register. Refer to Figure 6-8.

THIS	STR		(EA) ← Rd		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I:
			W ← DISP+/Q/
				IF DISP = 1F:
				W,L ← P
					S2

				IF I & DISP ≠ 1F W,L ← W+ < R >	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠1F W,L ← W+ < R >	S5
S2		P ← P + 1 + ISE	IF I: W ← M	IF I: W,L ← W+ < R >	S3
			I NOT I: W ← M+/Q/	IF NOT I: W,L ← W+ < R >	S5
S3			W ← M+/Q/	W,L ← W	S5
S4					
S5		M ← R		L ← P	S1
	Q,R ← Rd				

1. DISP = M0-4
2. ‹› INDICATES BASE RELATIVE ADDRESSING
3. // INDICATES INDEXING

SHT	NXT	REV

Figure 6-8. STR Instruction Sequence Chart

6.4.3 Compare Memory with Register (CMR)

The contents of the location as specified by the effective address are subtracted from the contents of the selected register with the results of the subtraction being recorded only in the indicators; Zero, Plus, Overflow and Link. The CMR instruction differs from the STR instruction in DO2 and DO3 of S5. In S5DO2, nothing happens. In S5DO3, the Q-register contains the contents of the selected register and the M-register contains the contents of the specified memory location. The contents of the Q-register are selected onto the augend bus and the contents of the M-register are selected onto the addend bus. The difference (Q-M) is placed on the data (D) bus and the adder results appear in the Zero, Plus, Overflow and Link indicators. Refer to Figure 6-9.

THIS	CMR		DB ← Rd-(EA)		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I:
			W ← DISP+/Q/
				IF DISP = 1F:
				W,L ← P	

				IF I & DISP≠1F W,L ← W +<R>	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ <R>	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+/Q/
				IF I:
				W,L ← W+ <R>
					S3

				IF NOT I: W,L ← W+ <R>	S5
S3			W ← M+/Q/	W,L ← W	S5
S4					
S5	Q,R ← Rd		DB ← Q - M ZPOL ← ADDER RESULTS	L ← P	S1

1. DISP = M0-4
2. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
3. // INDICATES INDEXING

REV

Figure 6-9. CMR Instruction Sequence Chart

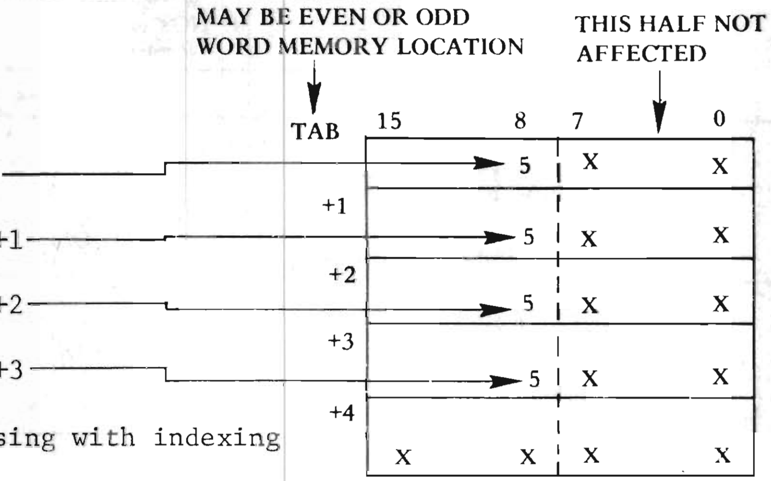
6.4.4 Bit/Byte Mode Addressing

Before describing the LDBY and STBY instruction, bit/byte mode addressing will be discussed.

Examples A and B illustrate the byte mode addressing both with and without indexing. In general, the effective address (EA) is composed of the displacement field, base relative and/or indirect address. The effective address is a word address pointing to a word location. The fact that this EA is odd or even has no bearing on the byte address mode. It is the odd/even contents of the specified index register which indicates the odd/even byte address. If no indexing is specified (that is, bits 8, 9 of the instruction are 00), the bit and byte instructions will always reference the left half of the word (bits 8-15). The right half of the word is unchanged. It is only when bits 8, 9 of the byte addressing instruction are 01, 10 or 11 that byte mode addressing is initiated. The word addressed is then EA+(X/2) and right or left half depending upon X being odd or even.

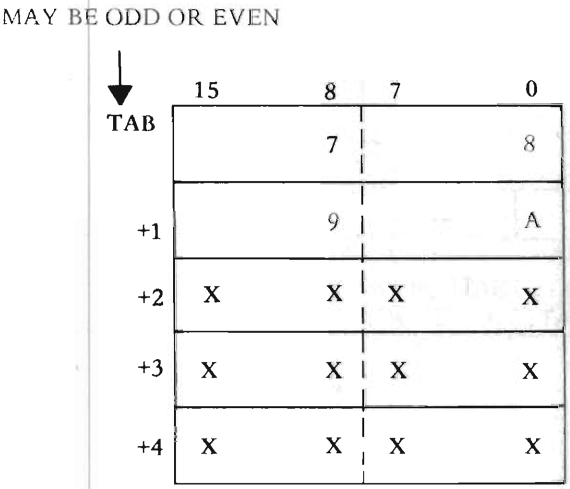
Example A. Byte addressing without indexing

```
    KON      EQU    5

             LDV    A,KON

             STBY   A,TAB

             STBY   A,TAB+1

             STBY   A,TAB+2

             STBY   A,TAB+3
```



Example B: Byte addressing with indexing

```
    KON      EQU    7

             LDV    A,KON

             LDV    X,0

             STBY   A,TAB,X

             INCR   X
             INCR   A
             STBY   A,TAB,X

             INCR   X
             INCR   A
             STBY   A,TAB,X

             INCR   X
             INCR   A
             STBY   A,TAB,X
```



6.4.5 Load Byte (LDBY)

If the contents of the selected X register is odd, the right byte (bits 0-7) of the memory location specified by the EA is placed in the right byte of the selected register. The left byte is not affected. If the contents of the selected X-register is even, the left byte (bits 8-15) of the memory location specified by the EA is placed in the right byte (bits 0-7) of the selected register. The left byte is not affected. Refer to Figure 6-10.

THIS	LDBY		Rd0-7 ← (EA)0-7 OR 8-15		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE
		Q,R ← Rd | Q ← Rx
			IF I:
			W ← DISP
			IF NOT I: $W \leftarrow DISP+/\frac{Q}{2}/$
			R ← D
				IF DISP = 1F:
				W,L ← P
				IF I & DISP ≠ 1F
				W,L ← W+ <R>
				IF NOT I & DISP ≠ 1F
				W,L ← W+ <R>
					S2
					S3
					S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			$W \leftarrow M+/\frac{Q}{2}/$
				IF I:
				W,L ← W+ <R>
				IF NOT I:
				W,L ← W+ <R>
					S3
					S5

S3			$W \leftarrow M+/\frac{Q}{2}/$	W,L ← W	S5
S4					
S5	Q,R ← Rd	Rd ← M	Rd8-15 ← Q8-15 IF INDEX IS EVEN: Rd0-7 ← R8-15 IF INDEX IS ODD: Rd0-7 ← R0-7 R ← Rd	L ← P	S1

1. DISP = M0-4
2. IF INDEXING IS NOT SELECTED, THE LEFT BYTE IS THE OPERAND.
3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-10. LDBY Instruction Sequence Chart

This instruction differs from the LDR instruction in the following sequence states:

S1D03: If a direct (bit 10 = 0) W ← DISP + /Q/2/. The DISP in bits 0-4 of the instruction is added to the index register (in Q) which has been shifted right one place (Q/2). The sum replaces the contents of W.

S2D03: Similarly, W ← M + /Q/2/.

S3D03: W ← M + /Q/2/.

S5D03: The destination register replaces the contents of the R-register. Bits 8-15 of the Q-register replace bits 8-15 of the destination register (i.e., bits 8-15 of the destination register are unchanged). If the contents of the index register are even, bits 8-15 of the R-register replace bits 0-7 of the destination register. If the contents of the index register are odd, bits 0-7 of the R-register replace bits 0-7 of the destination register.

6.4.6 Store Byte (STBY)

If the contents of the selected index register are odd, the right byte (bits 0-7) of the selected register replace the right byte of the memory location specified by the EA. The left byte is not affected. If the contents of the selected index register is even, the right byte (bits 0-7) of the selected register replace the left byte (bits 8-15) of the memory location specified by the EA. The instruction sequence chart associated with the STBY is shown in Figure 6-11.

STBY is identical to LDBY except in S5. Only the differences are described as follows:

S5D02: The left or right byte is affected depending on whether or not the contents of the selected index register is odd or even.

If the selected index register contents are even, the right byte (bits 0-7) of the R-register replace the left byte (bits 8-15) of the M-register. If the selected index register contents are odd, the right byte (bits 0-7) of the R-register replace the right byte of the M-register.

6.4.7 Set Bit (SBIT)

If the index is odd, place a 1 in the specified bit position (0-7) in the right byte of the location as specified by the effective address. If the index is even, place a 1 in the specified bit position (8-15) in the left byte of the location as specified by the effective address. Refer to Figure 6-12. Also set the zero indicator if the specified bit is zero before this instruction takes effect.

Only the difference between STBY and SBIT will be discussed.

THIS	STBY		(EA) 0-7 OR 8-15 ← Rd0-7		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I: W ← DISP+/$\frac{Q}{2}$/
				IF DISP = 1F:
				W,L ← P
					S2

				IF I & DISP ≠ 1F: W,L ← W+ ⟨ R ⟩	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F: W,L ← W+ ⟨ R ⟩	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+/$\frac{Q}{2}$/
				IF I:
				W,L ← W+ ⟨ R ⟩
					S3

				IF NOT I: W,L ← W+ ⟨ R ⟩	S5
S3			W ← M+/$\frac{Q}{2}$/	W,L ← W	S5
S4					

S5	Q, R ← Rd	IF INDEX IS EVEN:
		M8-15 ← R0-7
		IF INDEX IS ODD:
		M0-7 ← R0-7
				L ← P	S1

1. DISP = M0-4
2. IF INDEXING IS NOT SELECTED THE LEFT BYTE IS THE OPERAND.
3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-11. STBY Instruction Sequence Chart

THIS	SBIT		(EA) 0, ..., OR 15 ← 1		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I: W ← DISP + $/\frac{Q}{2}/$
				IF DISP = 1F:
				W,L ← P
					S2

				IF I & DISP ≠ 1F: W,L ← W+ <R>	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ <R>	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+$/\frac{Q}{2}/$
				IF I:
				W,L ← W+ <R>
					S3

				IF NOT I: W,L ← W+ <R>	S5
S3			W ← M+$/\frac{Q}{2}/$	W,L ← W	S5
S4					
S5	Q,R ← Rd	IF INDEX IS EVEN: M8, ..., OR 15 ← 1 ZERO ← M8, ..., OR 15 IF INDEX IS ODD: M0, . . . , OR 7 ← 1 ZERO ← M0, . . ., OR 7		L ← P	S1

1. DISP = M0
2. IF INDEXING IS NOT SELECTED
 ‘ THE LEFT BYTE IS THE OPERAND

3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-12. SBIT Instruction Sequence Chart

S5DO2: If index is even, a one is set into the specified bit of the left byte (bits 8-15) of the M-register. The selection of the bit is decoded by bits 5, 6 and 7 of the instruction.

If the index is odd a one is set into the specified bit of the right byte (bits 0-7) of the M-register.

The specified bit is forced to a one by virtue of the special addend bus logic. It then ORs with the other M-register bits when M is rewritten back into memory.

The selected bit is placed into the zero indicator prior to doing the set function. If indexing is not selected,the left byte is the operand.

6.4.8 Reset Bit (RBIT)

If the index is odd, place a zero in the specified bit position (0-7) in the right byte of the memory location as specified by the effective address. If the index is even, place a zero in the specified bit (8-15) in the left byte of the memory location as specified by the effective address. Refer to Figure 6-13.

THIS	RBIT		(EA) 0, ..., OR 15		NEXT
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I: W ← DISP
			IF NOT I: W ← DISP+/$\frac{Q}{2}$/
				IF DISP = 1F: W,L ← P	S2

				IF I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S5

S2		P ← P+1+ISE	IF I: W ← M
			IF NOT I: W ← M+/$\frac{Q}{2}$/
				IF I: W,L ← W+ ⟨ R ⟩	S3

				IF NOT I: W,L ← W+ ⟨ R ⟩	S5
S3			W ← M+/$\frac{Q}{2}$/	W,L ← W	S5
S4					
S5	Q,R ← Rd	IF INDEX IS EVEN: M8,. . . ., OR 15 ← 0 ZERO ← M8,. . . ., OR15 IF INDEX IS ODD: M0,. . . ., OR 7 ← 0 ZERO ← M0,. . . ., OR 7		L ← P	S1

1. DISP = M0-4
2. IF INDEXING IS NOT SELECTED, THE LEFT BYTE IS THE OPERAND

3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-13. RBIT Instruction Sequence Chart

The RBIT instruction is identical to the SBIT instruction except in S5D02.

S5D02: If the index is even, a zero is set into the selected bit of the left byte (bits 8-15) of the M-register. The selection of the bit is decoded by bits 5, 6 and 7 of the instruction.

If the index is odd a zero is set into the selected bit of the right byte (bits 0-7) of the M-register.

The M-register is selected onto the addend bus, sent through the ALU and the specified bit is set to zero. By virtue of the control in the addend logic, the specified bit is deselected as the M-register passes through the addend.

In each case, the state of the selected bit goes into the Zero indicator prior to setting the bit to zero.

6.4.9 Test Bit (TBIT)

If the index is odd, test the specified bit (0-7) of the right byte of the location as specified by the effective address. Set the Zero indicator accordingly. If the index is even, test the specified bit (8-15) of the left byte of the location as specified by the effective address. Set the Zero indicator accordingly. Refer to Figure 6-14.

This instruction is identical to the SBIT instruction except that no bits are affected in the specified memory location.

6.4.10 Increment Memory (INCM)

The contents of the memory location as specified by the effective address are incremented by one. Refer to Figure 6-15. S1, S2 and S3 are identical to the LDR instruction.

The difference in S5 is:

S5D02: The M-register is selected onto the addend bus, incremented and replaces the contents of M. The increment occurs as the result of forcing the carry FCIN into the ALU. At the same time ZPL is set. This places the ALU results into the Zero, Plus and Link indicators.

6.4.11 Decrement Memory (DECM)

The contents of the location as specified by the effective address are decremented by one. Refer to Figure 6-16. Sequence states S1, S2 and S3 are identical to the LDR instruction.

THIS	TBIT		ZERO ← (EA)0,. . . ., OR 15		NEXT
	DO1	DO2	DO3	DO4	
S1		P ← P+1+ISE	IF I: W ← DISP	IF DISP = 1F W,L ← P	S2
			IF NOT I: $W \leftarrow DISP+/\frac{Q}{2}/$	IF I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			$W \leftarrow M+/\frac{Q}{2}/$
				IF I: W,L ← W+ ⟨ R ⟩	S3

				IF NOT I: W,L ← W+ ⟨ R ⟩	S5
S3			$W \leftarrow M+/\frac{Q}{2}/$	W,L ← W	S5
S4					

S5	Q, R ← Rd	IF INDEX IS EVEN:
		ZERO ← M8,. . . ., OR 15
		IF INDEX IS ODD:
		ZERO ← M0, ..., OR 7
				L ← P	S1

1. DISP = M0-4
2. IF INDEXING IS NOT SELECTED, THE LEFT BYTE IS THE OPERAND.
3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-14. TBIT Instruction Sequence Chart

THIS	INCM		(EA)←(EA)+1		NEXT
	DO1	DO2	DO3	DO4	

S1		P← P+1+ISE	IF I:
			W←DISP
			IF NOT I:
			W← DISP+/Q/
				IF DISP = 1F
				W,L← P
					S2

				IF I & DISP ≠1F W,L← W+ ⟨ R ⟩	S3
		Q,R← Rd Q← Rx	R← D	IF NOT I & DISP ≠ 1F W,L← W+ ⟨ R ⟩	S5

S2		P← P+1+ISE	IF I:
			W←M
			IF NOT I:
			W← M+/Q/
				IF I:
				W,L← W+ ⟨ R ⟩
					S3

				IF NOT I: W,L← W+ ⟨ R ⟩	S5
S3			W← M+/Q/	W,L← W	S5
S4					
S5	Q,R← Rd	M← M+1 ZPL← ADDER RESULTS		L← P	S1

1.DISP = M0-4
2. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
3. // INDICATES INDEXING

REV

Figure 6-15. INCM Instruction Sequence Chart

THIS	DECM		(EA) ← (EA) - 1		
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W — DISP
			IF NOT I:
			W ← DISP+/Q/
				IF DISP = 1F:
				W,L ← P
					S2

				IF I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ ⟨ R ⟩	S5

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+/Q/
				IF I:
				W,L ← W+ ⟨ R ⟩
					S3

				IF NOT I: W,L ← W+ ⟨ R ⟩	S5
S3			W ← M+/Q/	W,L ← W	S5
S4					
S5	Q,R ← Rd	M ← M-1 ZPL ← ADDER RESULTS		L ← P	S1

1. DISP = M0-4
2. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
3. // INDICATES INDEXING

SHT	NXT		REV

Figure 6-16. DECM Instruction Sequence Chart

The difference in S5 is:

S5D02: The M-register is selected onto the addend bus. The SIN select is turned on with neither SDMA nor SIO on which forces all 1's on the augend bus (2's complement -1). The ALU results then replace the contents of M. At the same time ZPL is set. This places the ALU results into the Zero, Plus and Link indicators.

.4.12 Load All Registers and Status (LARS)

$A_{0-15} \leftarrow (EA)_{0-15}$
$X_{0-15} \leftarrow (EA+1)_{0-15}$
$Y_{0-15} \leftarrow (EA+2)_{0-15}$
$Z_{0-15} \leftarrow (EA+3)_{0-15}$
$B_{0-15} \leftarrow (EA+4)_{0-15}$
$C_{0-15} \leftarrow (EA+5)_{0-15}$
$D_{0-15} \leftarrow (EA+6)_{0-15}$
$E_{0-15} \leftarrow (EA+7)_{0-15}$

SHIFT COUNTER ← $(EA+8)_{0-3}$
LINK ← $(EA+8)_{4}$
OFLO ← $(EA+8)_{5}$
PLUS ← $(EA+8)_{6}$
ZERO ← $(EA+8)_{7}$
FOREGROUND ← $(EA+8)_{8}$

The LARS instruction is identical to the LDR in sequence states S1, S2 and S3. Sequence states S4 and S5 are described in the following paragraphs. Refer to Figure 6-17.

S4D02: $R_{rsc} \leftarrow M$. When S4D02 is first entered, the number in the register status counter (RSC) is zero, being cleared during S1. The contents of the M-register replace the contents of the register specified by the rsc subscript. In this case, since this number is zero, the A-register is selected.

S4D03: W ← W+1, RSC ← RSC+1. The W-register is incremented and placed back in W. This is done to keep track of the effective address, because we are going to use 9 contiguous memory addresses. The register and status counter (RSC) is also incremented.

THIS	LARS				
	DO1	DO2	DO3	DO4	

S1		P ← P+1+ISE	IF I:
			W ← DISP
			IF NOT I:
			W ← DISP+/Q/
				IF DISP = 1F:
				W,L ← P
					S2

				IF I & DISP ≠ 1F W,L ← W+ <R>	S3
		Q,R ← Rd Q ← Rx	R ← D	IF NOT I & DISP ≠ 1F W,L ← W+ <R>	S4

S2		P ← P+1+ISE	IF I:
			W ← M
			IF NOT I:
			W ← M+/Q/
				IF I:
				W,L ← W+ <R>
					S3

				IF NOT I: W,L ← W+ <R>	S4
S3			W ← M+/Q/	W,L ← W	S4
S4		R_{rsc} ← M	W ← W+1 RSC ← RSC+1	IF RSC ≠ 8 W,L ← W	S4
				IF RSC = 8 W,L ← W	S5
S5		FRGND, ZPOL & SC ← M		L ← P	S1
	Q,R ← Rd				

1. DISP = M0-4
2. RSC IS CLEARED DURING S1.
3. ⟨⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-17. LARS Instruction Sequence Chart

S4D04: RSC ≠ 8 W, L ← W, and S4 is next. RSC = 8 W, L ← W and S5 is next. If the RSC is not equal to 8, the contents of the W-register replace the contents of the W and the L-register. This gives us our next memory location and the sequencer goes back to S4D01. The first time through this loop, RSC=0. At the end of D04, a memory start clock is issued, and the next time through the loop, at D02, we have the new data and RSC=1. The contents of the M-register replace the X-register. The loop continues until RSC=7. This time (when RSC=7) the E-register is loaded. At this point we have loaded registers 0-7. At D03, the RSC is incremented and it now contains the number 8. The W-register has also been incremented and specifies the next memory location.

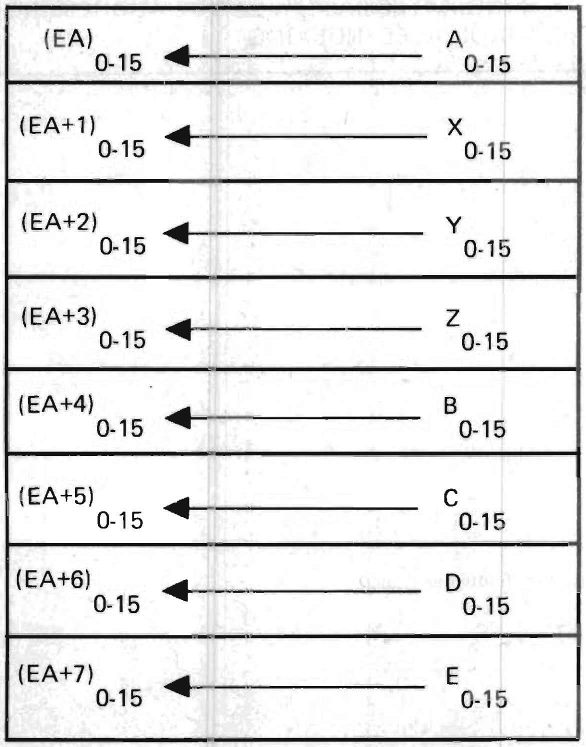
In D04, the test (RSC=8) is made and it is equal to 8. The next memory location transfers to the W and the L-register and the sequencer goes to S5D01.

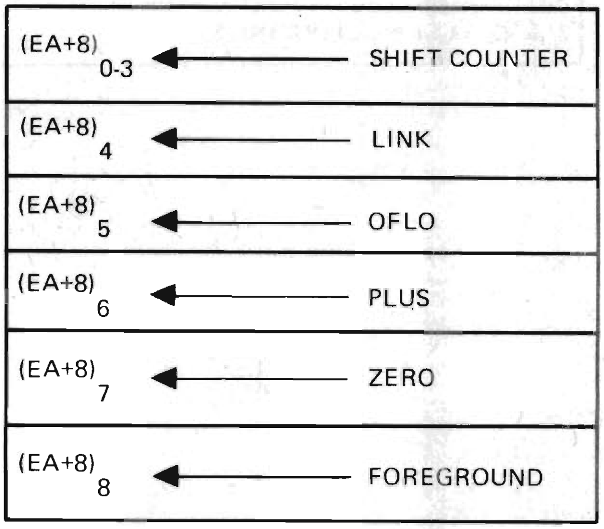
S5D02: FRGND, ZPOL and SC ← M. The contents of the M-register transfers through the adder and onto the data bus and the following bits go into the FRGND, ZPOL indicators and the shift counter.

Data Bus Bit	Indicator or Shift Counter
DB00-03	SC00-03
DB04	LINK
DB05	OFLO
DB06	PLUS
DB07	ZERO
DB08	FRGND

6.4.13 Store All Registers and Status (SARS)

The shift counter, the indicators and registers A,X,Y,Z,B,C, D and E replace the contents of the nine consecutive locations as specified by the EA through the EA+8.





The SARS instruction is identical to the LARS instruction in S1, S2 and S3. The differences in S4 and S5 are described in the following paragraphs. Refer to Figure 6-18.

THIS	DO1	DO2	DO3	DO4	NEXT
SARS					

S1		P←P+1+ISE	IF I:
			W←DISP
			IF NOT I:
			W←DISP+/Q/
				IF DISP = 1F:
				W,L←P
					S2

				IF I & DISP ≠ 1F W,L←W+ < R>	S3
		Q,R←Rd Q←Rx	R←D	IF NOT I & DISP ≠ 1F W,L←W+ <R>	S4

S2		P←P+1+ISE	IF I:
			W←M
			IF NOT I:
			W←M+/Q/
				IF I:
				W,L←W+ <R>
					S3

				IF NOT I: W,L←W+ < R>	S4
S3			W←M+/Q/	W,L←W	S4
S4		M←R	W←W+1 RSC←RSC+1	IF RSC ≠ 8 W,L←W	S4
	Q,R←R_{rsc}			IF RSC = 8: W,L←W	S5
S5		M← FRGND, ZPOL & SC		L←P	S1
	Q,R← Rd				

1. DISP = M0-4
2. RSC IS CLEARED DURING S1.
3. ⟨ ⟩ INDICATES BASE RELATIVE ADDRESSING
4. // INDICATES INDEXING

REV

Figure 6-18. SARS Instruction Sequence Chart

S4D01: Q, R ← R_{rsc}. The contents of the register specified by the RSC subscript replaces the contents of the Q and the R-registers. The first time through the S4 1-op RSC=0 so the A-register is specified.

S4D02: M ← R. The contents of the R-register (in this case, the A-register is in R) replace the contents of the M-register. The contents of the A-register are now ready to be stored in memory at the memory location specified by the effective address.

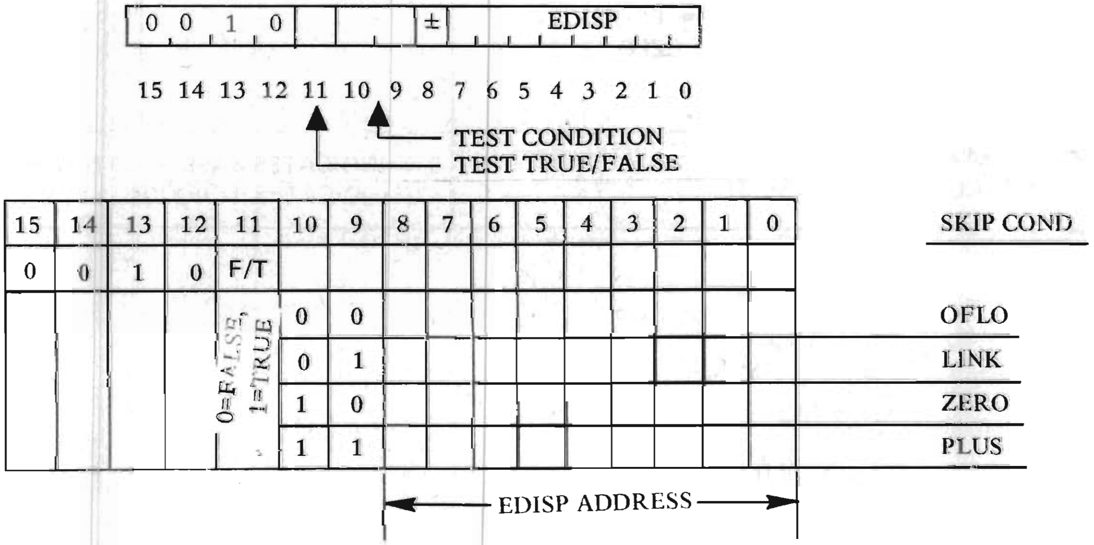
S4D03: W ← W+1; RSC ← RSC+1. The effective address in W is incremented and placed back in W. The register and status counter (RSC) is also incremented.

S4D04: RSC ≠ 8 W, L ← W and S4 is next; RSC = 8 W, L ← W and S5 is next. If the RSC is not equal to 8, the contents of the W-register replace the contents of the W and L-register. This gives our next memory location and the sequencer goes back to S4D01. The first time through the loop, RSC=0. At the end of D04, memory start clock is issued. The next time through the loop, RSC=1 and the contents of the X-register replace the contents of the M-register. This loop continues until RSC=7 and the E-register is loaded into M. At D03, the RSC is incremented and it now contains the number 8. The W-register has been incremented and specifies the next memory location and the sequencer goes to S5.

S5D02: M ← FRGND, ZPOL and SC. The FRGND, ZERO, PLUS, OFLO, LINK and the shift counter are loaded into the M-register.

S5D04: L ← P. The contents of P replace the contents of the L-register and the sequencer goes back to S1.

6.5 SKIP INSTRUCTIONS



If the condition being tested is met, the effective address is used; if the condition is not met, the next instruction in sequence is executed. Refer to Figure 6-19.

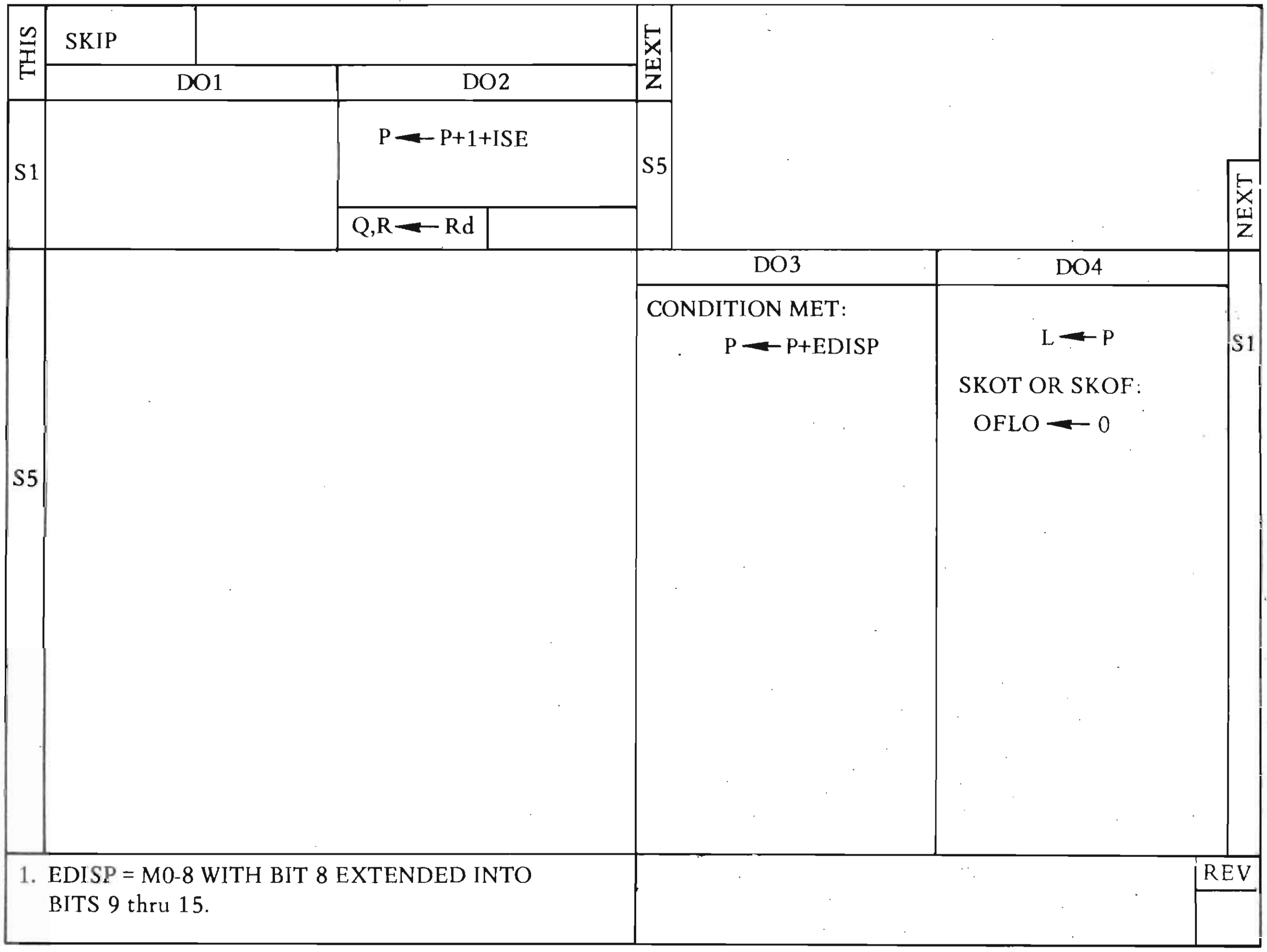


Figure 6-19. SKIP Instruction Sequence Chart

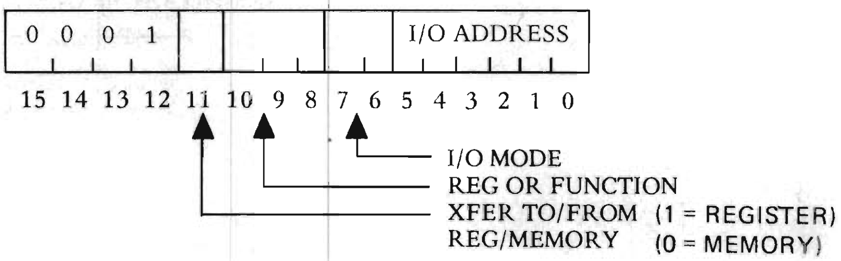
S1D02: Q, R ← Rd, P ← P+1+ISE. The P counter is incremented and placed back into P. The destination register replaces the contents of the Q and the R-register. The sequencer then goes to S5D03.

S5D03: P ← P+EDISP. If the condition is met, P+EDISP replaces the contents of P. EDISP=M-register bits 0-8 with bit 8 extended into bits 9-15. If the condition is not met, the next effective address is P+1.

S5D04: L ← P, OFLO ← 0 (SKOF or SKOT). If the instruction is a SKOF or SKOT, the OFLO indicator will be reset after the condition is tested. The contents of the P-register replace the contents of the L-register and the sequencer goes back to S1.

6.6 INPUT/OUTPUT INSTRUCTIONS (XIO)

6.6.1 XIO Control and Test Instructions (CTRL, TEST)



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	1	-	-	-	-	-	-	-	-	-	-	.	-	
				0	←FUN→			0	0							CONTROL
				1/0	←R→			0	1							OUTPUT
				1/0	←R→			1	0							INPUT
				0	←FUN→			1	1							TEST

← I/O ADDRESS →

The XIO command is used for control and test functions as well as data transfers between the processor and I/O devices. Refer to Figure 6-20.

THIS	XIO	CTRL, TEST			NEXT
	DO1	DO2	DO3	DO4	
S1		P ← P+1+ISE Q,R ← Rd R ← Rs		W,L ← W	S5
S2					
S3					
S4					

S5	Q,R ← Rd	IF TEST:
		IOT ← I/O CABLE TEST LINE
		IF TEST & IOT:
		P ← P+1+ISE
		IF TEST & NOT IOT:
		P ← P
				L ← P	S1

					REV

Figure 6-20. XIO Control and Test Sequence Chart

S1D02: P ← P+1+ISE, Q, R ← Rd, R ← Rs. The P counter is incremented and placed back into P. The destination register (Rd) replaces the contents of the Q and the R-register and the source register (Rs) replaces the contents of the R-register. The R ← Rs is not needed for the CTRL and TEST instruction, but is needed for the data transfer instruction. The CTRL and TEST instructions normally specify memory. (If register is specified, i.e., IR11=1, then the specified register will be changed.) Hence, memory start clock is issued in D04 in a useless memory access, since the memory data is not used by the instruction.

S5D02: The following events occur in S5D02. If TEST: take the test line from the I/O device and place the contents in the I/O test flip-flop. The test line is examined and clocked into I/O test flip-flop at lead edge of S5D02.

If TEST and IOT: (I/O test flip-flop set): The P-register is incremented a second time; it was incremented once in S1 (for next instruction) and again in S5. This is so that the next instruction in sequence will be skipped and the instruction in P+2 will be executed.

If TEST and Not IOT: (I/O test flip-flop set): The P-register is not incremented; that is, the contents of P replace the contents of P. The next instruction in sequence (P+1) is executed.

S5D04: L ← P. The contents of P replace the contents of the L-register and the sequencer goes to S1.

6.6.2 XIO Data Transfers

The XIO data transfer instruction is identical to the XIO Test and Control in S1D02. The difference thereafter are described in the following paragraphs. Refer to Figure 6-21.

S1D03: W ← R. If bit 11 is a zero, it is a memory I/O type data transfer and the effective address in R is transferred to W. If bit 11 is a one, W will be used in S5D02 for DTOR.

S1D04: W, L ← W if DTIM, DTOM, or RCSM. If it is a Data Transfer Into Memory (DTIM), Data Transfer Out of Memory (DTOM), or Read Console Switches Into Memory (RCSM), the effective address in W is transferred to the W- and to the L-register. The sequencer then goes to S5.

If it is a Data Transfer Into Register (DTIR), Data Transfer Out of Register (DTOR), or Read Console Switches Into Register (RCSR), a W, L ← W still takes place but no memory start clock is issued. In order to maintain proper I/O timing another sequence cycle-time is required, so the sequencer goes to S5.

S5D02: If it is a data transfer out of register, the W-register is selected onto the augend bus, sent through the ALU and replaces the contents of the M-register. Then the I/O is selected so the data in the M-register has been selected onto the I/O bus.

THIS	XIO	DATA TRANSFERS			NEXT
	DO1	DO2	DO3	DO4	
S1		P ← P+1+ISE Q,R ← Rd R ← Rs	W ← R	IF DTIM,DTOM, OR RCSM W,L←W IF DTIR,DTOR,OR RSCR W,L←W NO MEMORY START CLOCK & NO CLEAR M	S5
S2					
S3					
S4					
S5	Q,R ← Rd	DTOR: I/O←M←W DTIR: Rs←I/O RCSR: Rs←I/O †† DTOM: I/O←M DTIM: M←I/O RCSM: M←I/O ††		L ← P	S1
		SHT	NXT		REV

†† Source data for RCSR, RCSM is from console switches.

Figure 6-21. XIO Data Transfer Sequence Chart

If it is a Data Transfer Into Register or a Read Console Switches Into Register, the I/O bus is selected onto the augend bus, and replaces the contents of the source (Rs) register decoded by bits 8, 9, and 10 of the XIO instruction.

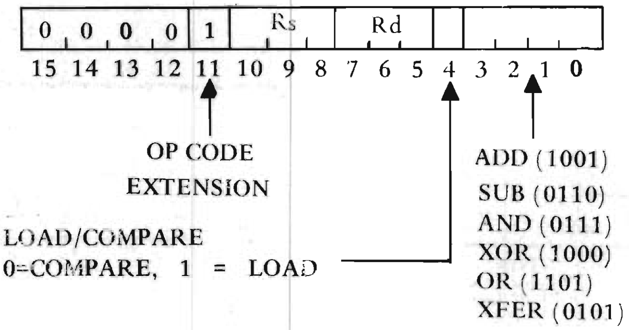
If it is a Data Transfer Out of Memory, the data in the M-register is selected (SOUT) onto the I/O bus.

If it is a Data Transfer Into Memory or a Read Console Switches Into Memory, the I/O bus is selected onto the augend bus and replaces the contents of the M-register.

S5D04: L ← P. The updated P-register replaces the contents of the L-register and the sequencer goes to S1.

6.7 REGISTER OPERATE AND REGISTER OPERATE COMPARE INSTRUCTIONS

The instructions in this group are of the format shown below. The decodes of the instruction are also shown.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	0	0	1	←Rs→			←Rd→			1					OPERATE	COMPARE
											0 = COMPARE, 1 = LOAD	0	1	0	1	RTR	
												1	0	0	1	ADD	ADDC
												0	1	1	0	SUB	SUBC
												1	1	0	1	OR	ORC
												1	0	0	0	XOR	XORC
												0	1	1	1	AND	ANDC

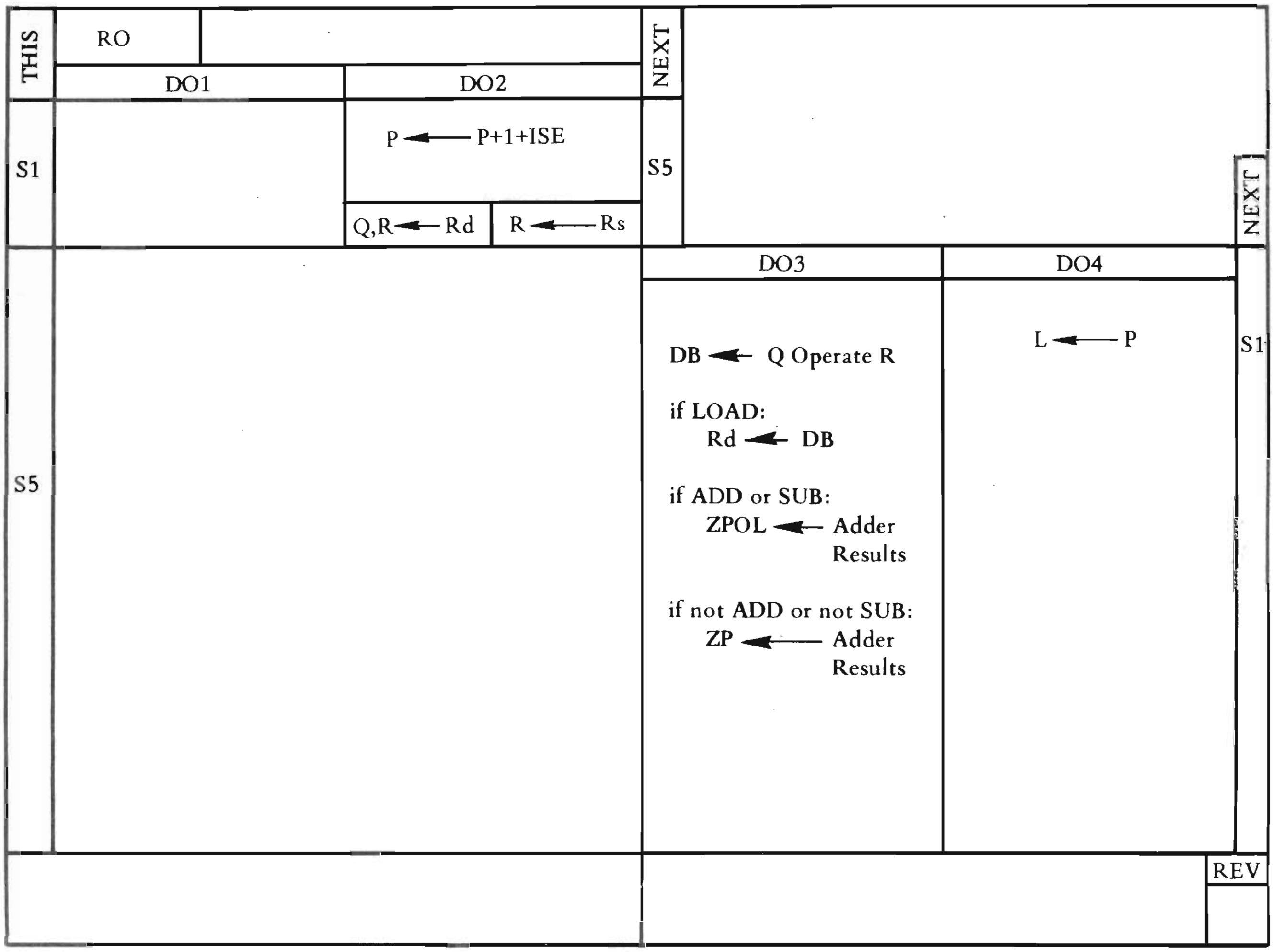
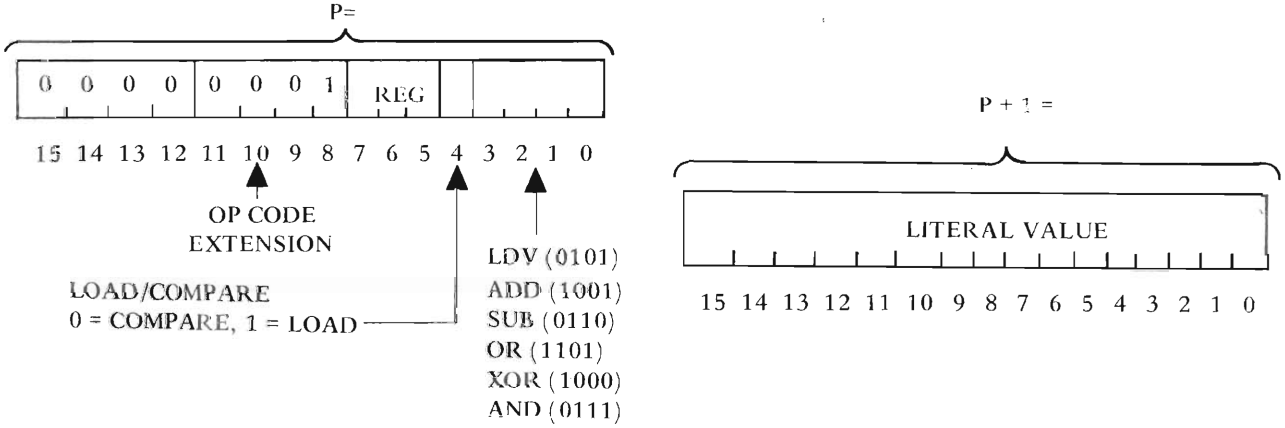


Figure 6-22. Register Operate Instruction Sequence Chart



Bits 5, 6 and 7 designate the destination register and bits 8, 9 and 10 designate the source register. The register bit code and the designated register are presented below.

Bit Code	Register	Bit Code	Register
000	A	100	B
001	X	101	C
010	Y	110	D
011	Z	111	E

If bit four is a one, a load is specified. The operation requested is performed and the result replaces the contents of the destination register (Rd) and the conditions of the result are placed into the indicators (Zero, Plus, Overflow, Link).

If compare is specified, the operation requested is performed and the conditions of the result are placed into the indicators. The destination register is unchanged. Refer to Figure 6-22.

S1D02: Q, R ← Rd, P ← P+1+ISE, R ← Rs. The destination register decoded by bits 5, 6 and 7 replaces the contents of the Q and the R-registers. The P counter is incremented and placed back into P. The source register decoded by bits 8, 9 and 10 replace the contents of the R-register. The sequencer then goes to S5D03.

S5D03: Different events occur depending on the state of bit 4. Assume Bit (4) is off, therefore a compare is specified. The Q-register contains the destination register and the R-register contains the source register. The ALU does a Q operate R just to the data bus. Just a compare of data (ADDC, SUBC, ORC, XORC, ANDC) is done. Only the ZPOL (all four indicators) are affected.

Assume the load bit (4) is on, therefore a load is specified,the contents of the data bus will replace the contents of the destination register.

S5D04: L ← P: The updated P counter replaces the contents of the P and the L-registers and the sequencer goes back to S1.

6.8 REGISTER OPERATE LITERAL AND REGISTER COMPARE LITERAL INSTRUCTIONS

The instructions in this group are of the following format. The decodes of all the instructions in this group are also shown.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	0	0	0	0	0	1	← Rd →			1					OPERATE	COMPARE
											0 = COMPARE, 1 = LOAD	0	1	0	1	LDV	
												1	0	0	1	ADDV	ADDVC
												0	1	1	0	SUBV	SUBVC
												1	1	0	1	ORV	ORVC
												1	0	0	0	XORV	XORVC
												0	1	1	1	ANDV	ANDVC

The destination register is specified by bits 5, 6 and 7 as shown below.

Bit Code	Register	Bit Code	Register
000	A	100	B
001	X	101	C
010	Y	110	D
011	Z	111	E

If LOAD is specified, the operation requested is performed and the result replaces the contents of the destination register (Rd) and the conditions of the result are placed into the indicators (Zero, Plus, Overflow, Link).

If COMPARE is specified, the operation requested is performed and the conditions of the results are placed into the indicators. The destination register (Rd) remains unchanged. Refer to Figure 6-23.

S1D02: Q,R ← Rd, P ← P+1+ISE. The destination register decoded by bits 5, 6 and 7 of the instruction replace the contents of the Q and the R-registers. Then, the P counter is incremented and is placed back in P.

S1D03: R ← D. This happens but does not affect the execution of the instruction.

S1D04: L ← P. The updated P counter replaces the contents of the L-register, the sequencer then goes to S5.

S5D02: P ← P+1+ISE. The P counter must be incremented a second time (first increment was in S1D02) because the literal value is the next location after the instruction.

THIS	ROL				NEXT
	DO1	DO2	DO3	DO4	
S1		P ← P+1+ISE Q,R ← Rd	R ← D	L ← P	S5
S5	Q,R —— Rd	P ← P+1+ISE	DB ← Q Operate M IF LOAD: Rd ← DB IF ADD OR SUB: ZPOL ← ADDER RESULTS IF NOT ADD OR NOT SUB: ZP ← ADDER RESULTS	L ← P	S1
					REV

Figure 6-23. Register Operate Literal Instruction Sequence Chart

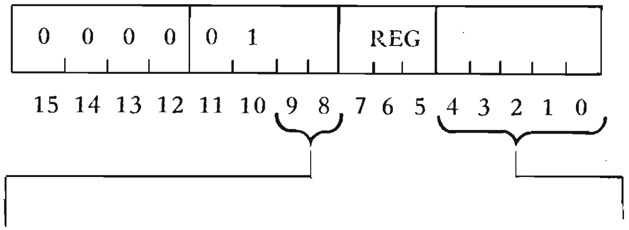
S5D03: Different events occur depending on the state of bit 4. If the load bit (4) is off, a compare is specified. The Q-register contains the destination register and the M-register contains the data. Q operate M is done by the adder with results appearing just on the data bus. A compare of data only (ADDC, SUBC, ORC, XORC, ANDC) is done. All four of the indicators (Z,P,O,L) are affected if the operation is an add or subtract. However, only the Z and P indicators are affected if a logical operation is performed.

Assume the load bit is on, therefore a load is specified and the contents of the data bus replace the contents of the destination register.

S5D04: L ← P. The P counter has been incremented twice, once in S1D02 and the second time in S5D02. The contents of the P counter replace the contents of the L counter and the sequencer goes to S1.

6.9 REGISTER CHANGE INSTRUCTIONS

There are 16 instructions in the Register Change group and each of these instructions is of the following format.



BITS	INSTRUCTION	BITS
9 8		43210
0 1	RESTORE INTERRUPT SYSTEM ENABLE	00001
0 1	SUBROUTINE RETURN	00011
0 1	DISPLAY REGISTER	00100
0 1	TRANSFER REGISTER TO STATUS	01000
0 1	EXECUTE	10000
1 0	ZERO	00000
1 0	ZERO RIGHT BYTE	00001
1 0	ZERO LEFT BYTE	00010
1 0	EXCHANGE BYTES	00100
1 0	TRANSFER STATUS TO REGISTER	01000
1 0	READ CONSOLE SWITCHES	10000
1 1	COMPLEMENT	00000
1 1	REGISTER ADD LINK	00001
1 1	DECREMENT	00010
1 1	ADD SHIFT COUNTER TO REGISTER	01011
1 1	INCREMENT	01110

The destination register is designated by bits 5, 6 and 7 as shown below.

Bit Code	Register	Bit Code	Register
7 6 5		7 6 5	
0 0 0	A	1 0 0	B
0 0 1	X	1 0 1	C
0 1 0	Y	1 1 0	D
0 1 1	Z	1 1 1	F

The operation requested is performed with the selected register and the result replaces the contents of that register. The conditions of that result are placed in the indicators (Z,P,O,L) associated with the operation.

The following instructions are described by the instruction sequence chart in Figure 6-24:

Register Add Link (RLK)
Add Shift Counter to Register (ADDS)
Increment Register (INCR)
Decrement Register (DECR)
Register Complement (CMPL)

S1D02: Q, R ← Rd, P ← P+1+ISE, R ← Rs. This state is identical to the register operate group of instructions. The destination register replaces the contents of the Q and the R-register, the P counter is incremented and placed back into P and the source register replaces the contents of the R-register. The sequencer then goes to S5.

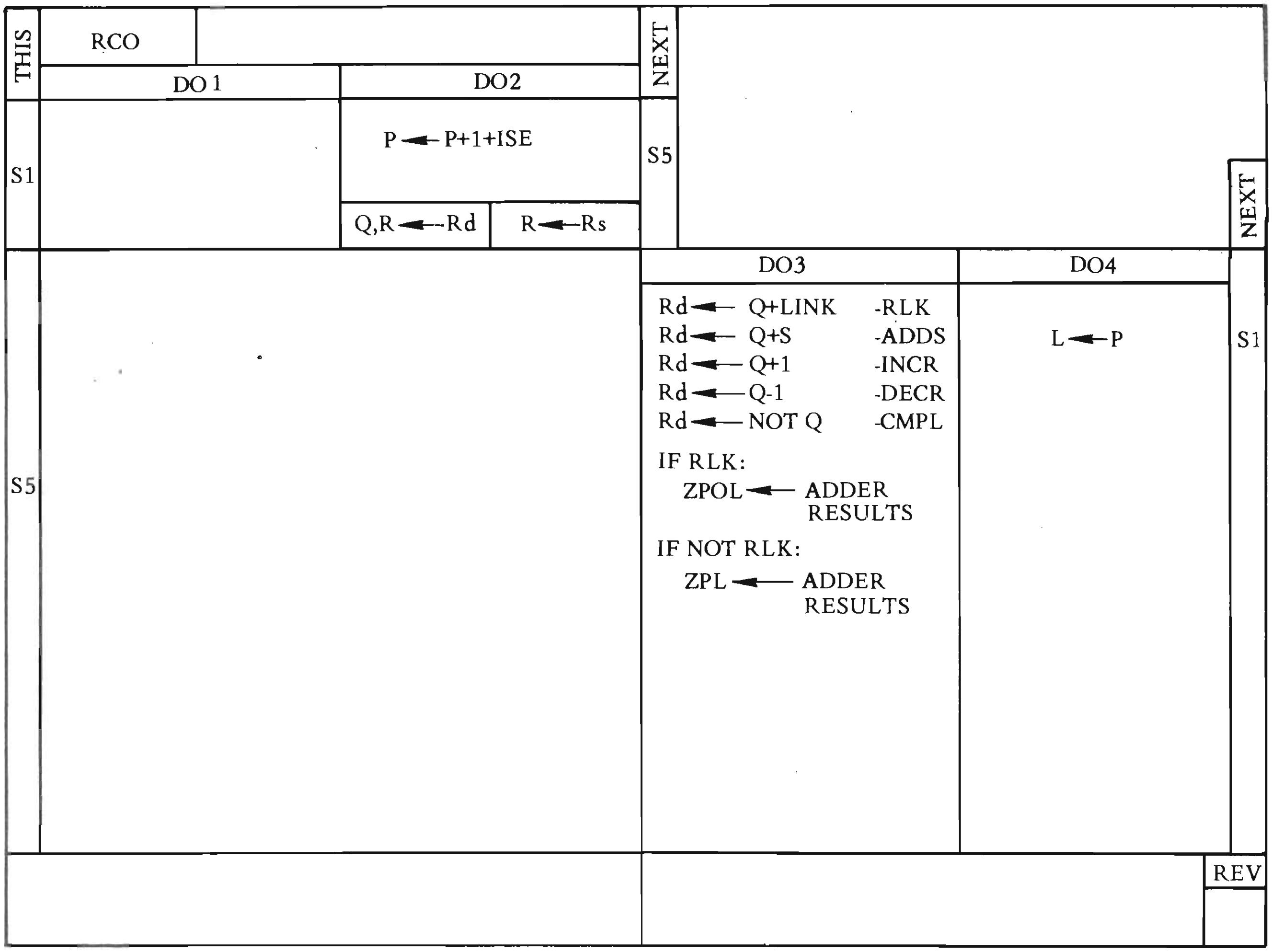
S5D03: Various operations occur depending on the bit configuration of bits 0-4. The operations are described as follows:

Register Add Link. Rd ← Q+Link. The Q-register now contains the destination register. The contents of the Link indicator are added to the Q and the sum replaces the destination register decoded by bits 5, 6 and 7. The ALU results affect the Z, P, O and L indicators.

Add Shift Counter to Register: Rd ← Q+S. The contents of the shift counter (bits 0-3) are added to Q which contains the destination register. The sum is placed back in the destination register. The ALU results affect the Z,P,O and L indicators.

Increment: Rd ← Q+1. If an increment is requested, plus one is added to Q and the sum is placed back into the destination register. The ALU results affect the Z, P and L indicators only.

Decrement: Rd ← Q-1. The ALU is configured so that the augend side can be decremented, the addend side can't. The Q-register is selected through the augend side and is decremented. The result is placed back into the destination register. The ALU results affect only the Z, P and L indicators.



THIS RCO

NEXT

DO 1

DO2

S1

P ◄— P+1+ISE

Q,R ◄— Rd

R ◄— Rs

S5

NEXT

DO3

DO4

S5

Rd ◄— Q+LINK -RLK
Rd ◄— Q+S -ADDS
Rd ◄— Q+1 -INCR
Rd ◄— Q-1 -DECR
Rd ◄— NOT Q -CMPL

IF RLK:
ZPOL ◄— ADDER RESULTS

IF NOT RLK:
ZPL ◄— ADDER RESULTS

L ◄— P

S1

REV

Figure 6-24. RLK, ADDS, INCR, DECR, CMPL Instruction Sequence Chart

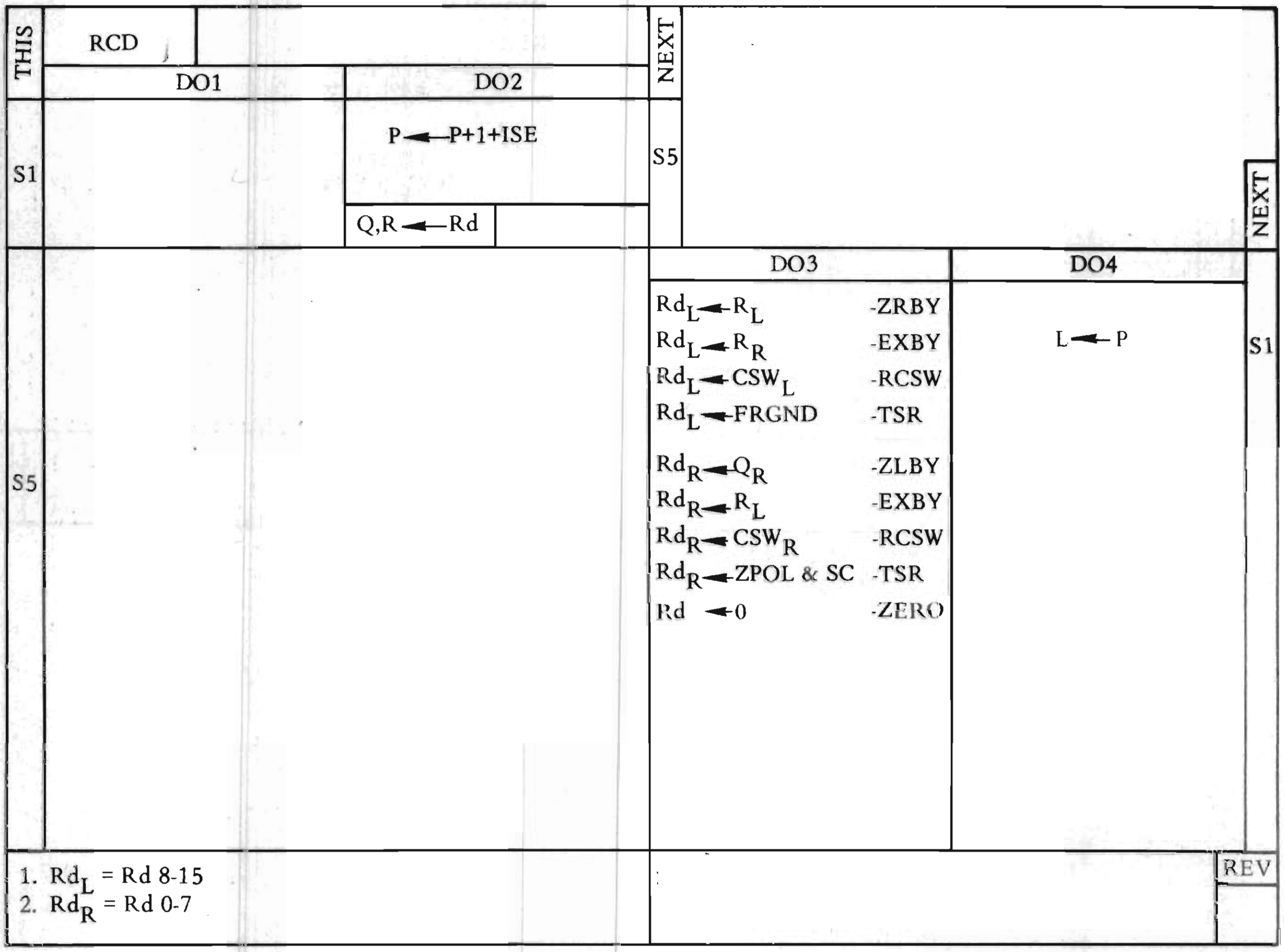
The ALU is configured so that the augend side can be complemented.

Complement: Rd ← not Q. The complement of the Q-register is placed into the destination register. Only the Z, P and L indicators are affected.

S5DO4: L ← P. The incremented P counter replaces the contents of the L-register and the sequencer goes back to S1.

The instruction sequence chart in Figure 6-25 is associated with the following register change instructions.

- Zero Right Byte (ZRBY)
- Zero Left Byte (ZLBY)
- Read Console Switches (RCSW)
- Zero Register (ZERO)
- Exchange Byte (EXBY)
- Transfer Status to Register (TSR)



DO3	
$Rd_L \leftarrow R_L$	-ZRBY
$Rd_L \leftarrow R_R$	-EXBY
$Rd_L \leftarrow CSW_L$	-RCSW
$Rd_L \leftarrow$ FRGND	-TSR
$Rd_R \leftarrow Q_R$	-ZLBY
$Rd_R \leftarrow R_L$	-EXBY
$Rd_R \leftarrow CSW_R$	-RCSW
$Rd_R \leftarrow$ ZPOL & SC	-TSR
$Rd \leftarrow 0$	-ZERO

DO4: $L \leftarrow P$

1. Rd_L = Rd 8-15
2. Rd_R = Rd 0-7

Figure 6-25. ZRBY, ZLBY, RCSW, TSR, EXBY, ZERO Instruction Sequence Chart

S1D02: Q, R ← Rd, P ← P+1+ISE. The destination register decoded by bits 5, 6 and 7 replaces the contents of the Q and the R-registers. The P counter is incremented and placed back into P.

S1D03: Different events occur depending on the instruction decoded (bits 0-4).

Zero Right Byte: $Rd_L \leftarrow R_L$. The right byte is zeroed by virtue of no select on it, as the left byte only is passed through the ALU back to the register. Therefore, the left byte is saved. Rd_L is equal to destination register bits 8-15. Bits 8-15 of the R-register replace bits 8-15 of the destination register while zeroes replace bits 0-7 of the destination register.

Zero Left Byte: $Rd_R \leftarrow Q_R$. A select Q right to right causes bits 0-7 of the Q-register to be selected onto the augend bus then into the destination register. No select into the ALU for the left byte yields zeroes in bits 8-15 of the destination register.

Exchange Bytes: $Rd_L \leftarrow R_R$, $Rd_R \leftarrow R_L$. A select R-register right to left (SRRL) causes bits 0-7 of the R-register to be transferred to bits 8-15 of the destination register. A SRLR causes bits 8-15 to be transferred to bits 0-7 of the destination register.

Read Console Switches: $Rd_L \leftarrow CSW_L$, $Rd_R \leftarrow CSW_R$. The contents of the console switches (bits 0-15) are placed in bits 0-15 of the destination register.

CAUTION

If this instruction is followed immediately by an interrupt or a cycle steal, the console switch data may be OR'ed with the cycle steal address or the interrupt vector. To avoid this problem, substitute the Read Console Switches into Register instruction (RCSR) or the Read Console Switches into Memory instruction (RCSM). These are Programmed Input/Output instructions, discussed in Section 6.6.2.

Transfer Status to Register: Rd_L ← FRGND, Rd_R ← ZPOL & SC. The status indicators are transferred to the following bits of the destination register.

Indicator	Destination Register Bit
FRGND	8
ZERO	7
PLUS	6
OFLO	5
LINK	4
Shift Counter	0-3

S5DO4: L ← P. The incremented P counter replaces the contents of the L-register.

The instruction sequence chart shown in Figure 6-26 is associated with the following register change instructions:

Display Register (DSPL)
Transfer Register to Status (TRS)
Restore Interrupt System Enable (RISE)
Subroutine Return (RTRN)
Execute Register Contents (XEC)

THIS	RCS		NEXT
	DO1	DO2	

S1	IF NOT XEC:
	NO ACTION
	IF XEC:
	M←Q
		IF NOT XEC:
		P← P+1+ISE
		IF XEC:
		P←P+ISE
		Q,R←Rd R←Rs
			S5

		DO3	DO4	NEXT

S5		K←Q -DSPL
		FRGND ZPOL & S←Q -TRS
		ISE←Q15 -RISE
		P←Q -TRP
		M← Q (NOTE 1) -XEC
			L←P
			IF XEC, NO MEMORY START CLOCK AND NO CLEAR M.
				S1

1. FOR FAST (OVERLAPPED) ROM PROCESSING ONLY

REV

Figure 6-26. DSPL, TRS, RISE, RTRN, XEC Sequence Chart

S1D01: No action if not an execute (XEC) object instruction. If it is an execute object instruction the contents of Q replace the contents of M.

S1D02: Q, R ← Rd, R ← Rs, P ← P+1+ISE if not XEC, P ← P+ISE if XEC. The P counter is incremented if it is not an execute (XEC). If it is an XEC then the P counter is not incremented and a P ← P is done, because P will be incremented when the object instruction is executed.

In addition, the destination register replaces the contents of the Q and the R-registers. The source register replaces the contents of the R-register. The sequence then goes to S5.

S5D03: Different events occur depending on the decode of bits 0-4.

Register Console Display: K ← Q. The contents of the Q-register replace the contents of the K-register.

Transfer Register to Status: FRGND, ZPOL and S ← Q. The contents of the Q-register are transferred to FRGND, Z, P, O and L and the shift counter.

Restore ISE: ISE ← Q15. Bit 15 of the destination register which is in Q is transferred to the ISE indicator.

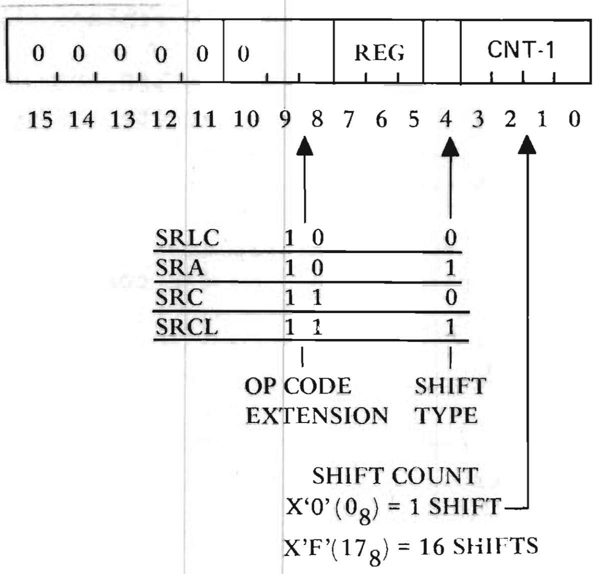
Transfer Register to P: P ← Q. The contents of the destination register (Q) replace the P counter. This is the subroutine return instruction.

Execute: M ← Q. This only happens in D03 for fast ROM processing, otherwise it happens when S1 is entered again at D01.

S5D04: L ← P. The incremented P counter replaces the contents of the L-register if not an XEC instruction. If this is an XEC, no memory start clock or set L is issued and the M-register is not cleared. This is because the contents of M must be transferred to the I REG at the next D01. The sequencer then goes back to S1.

6.10 SHIFT INSTRUCTIONS

There are four instructions in the shift group and each of these instructions are of the following format:



The number of shifts specified is performed. The destination register is specified by bits 5, 6 and 7 as follows:

Bit Code 7 6 5	Register
0 0 0	A
0 0 1	X
0 1 0	Y
0 1 1	Z
1 0 0	B
1 0 1	C
1 1 0	D
1 1 1	E

Refer to Figure 6-27 for a shift instruction flowchart and to Figure 6-28 for the instruction sequence chart for the shift instructions.

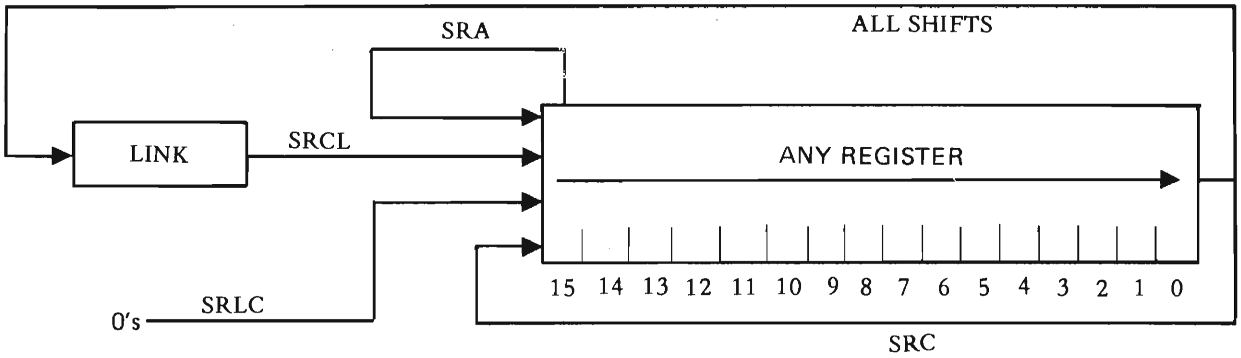


Figure 6-27. Shift Instruction Flowchart

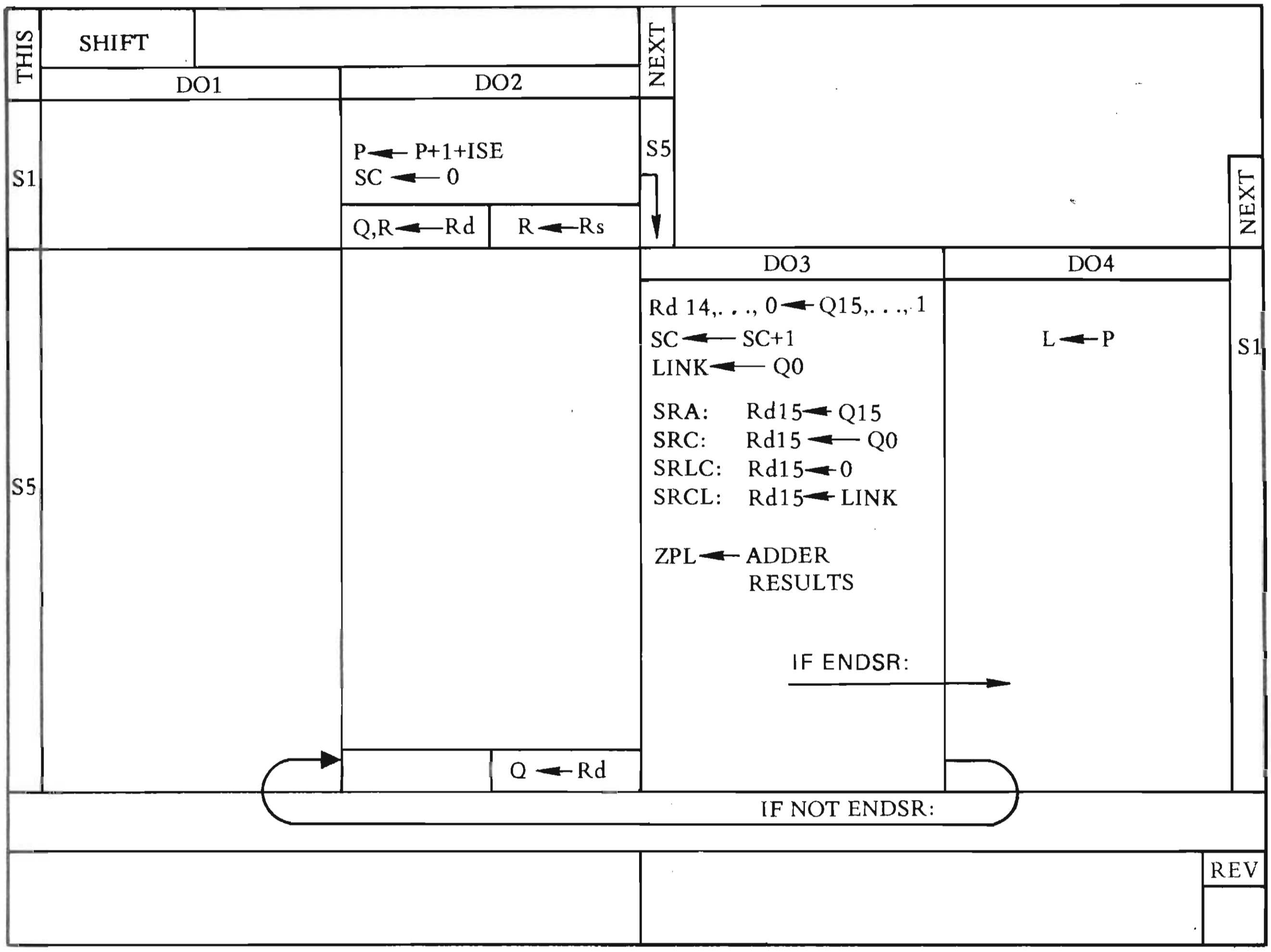


Figure 6-28. Shift Instruction Sequence Chart

S1DO2: Q, R ← Rd, P ← P+1+ISE, SC = 0, R ← Rs. The destination register replaces the Q and R-registers. The P counter is incremented and placed back into P. The shift counter is zeroed. The source register replaces the contents of the R-register though it is not used in this sect of instructions.

S5DO3: Rd14, ..., 0 ← Q15, ..., 1, SC ← SC+1, Link ← Q0. Bits 1-15 of the Q-register replace bits 0-14 of the destination register. This effectively shifts the register right one place. The shift counter is incremented. Bit 0 of the Q-register is placed into LINK.

In S5DO3, different events occur depending on the shift instruction (SRA, SRC, SRLC, SRCL) decoded. These are described in the following paragraphs. Refer to the shift instruction flowchart in Figure 6-27. The ALU results effect the Z, P and L indicators.

Shift Right Arithmetic (SRA): Rd15 ← Q15. The destination register is shifted right as specified by the shift count. The sign (bit 15) is maintained and propogated to the right as specified by the shift count. Bits are shifted out of bit 0 into LINK. Bits shifted out of LINK are lost. If the end shift (ENDSR) signal is present, the sequencer goes to S5DO4 to terminate. If not present, it goes to S5DO2 where the destination register replaces the Q-register, S5DO3 is again entered and the sequencer goes through the loop again (Q-register shifted right and the shift counter is incremented). This continues until the ENDSR signal occurs. One shift is accomplished for each pass through S5DO3.

Shift Right Circular (SRC): Rd15 ← Q0. The destination register (Q) is shifted right the number of positions specified by the shift count. Bit 0 of the Q-register is shifted into bit 15 of the destination register. Bits are shifted out of bit 0 into LINK. Bits shifted out of LINK are lost. If the ENDSR signal is not present, the sequencer goes to S5DO2 then back through S5DO3 and where the Q-register is shifted right one place and the shift counter is incremented. One shift is accomplished for each pass through S5DO3.

The sequencer continues to go through this loop until the ENDSR signal is present. When this occurs, the sequencer goes to S5DO4 where the instruction is terminated.

Shift Right Logical and Count (SRLC): Rd15 ← 0. A zero is shifted into bit 15 of the destination register. If the ENDSR signal is not present, the sequencer goes back to S5DO2 where the destination register replaces the contents of the Q-register. In S5DO3, the Q-register is shifted right one place and the shift counter is incremented. The sequencer goes through this loop until the ENDSR signal occurs (shift counter counted out or until LINK contains a one, whichever occurs first).

Bits are shifted out of bit 0 into LINK. Bits shifted out of LINK are lost. One shift is accomplished for each pass through S5DO3.

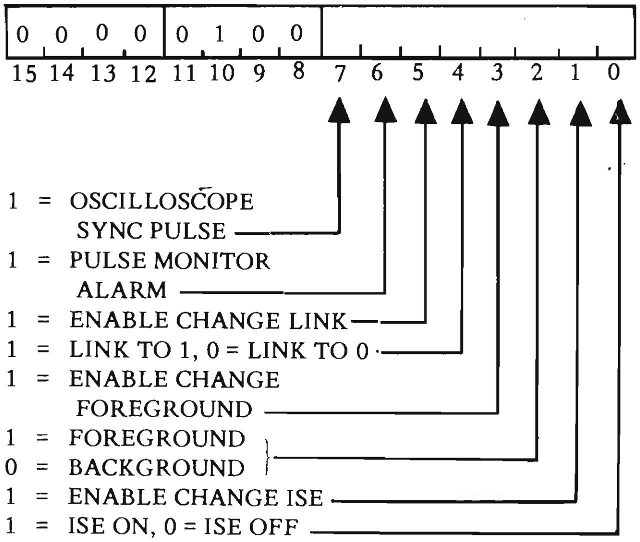
When the ENDSR is present, the sequencer goes to S5DO4 so that the instruction can be terminated.

Shift Right Circular through Link (SRCL): Rd15 ← LINK. The LINK is shifted into bit 15 of the destination register. Bit 0 is shifted into the LINK. If the ENDSR signal is not present the sequencer goes back to S5D02 to initiate another shift pass. In S5D03, the Q-register is shifted right one place and the shift counter is increment. The instruction goes through the loop until the ENDSR signal occurs. The sequencer then goes to S5D04 where it terminates. One shift is accomplished for each pass through S5D03.

S5D04: L ← P. The contents of the P counter replace the contents of the L-register and the sequencer goes back to S1.

6.11 CONTROL INSTRUCTIONS

The eight instructions in the control group are of the following format. The instruction sequence chart associated with the Control instructions is shown in Figure 6-29.



THIS

CONTROL

	DO1	DO2	NEXT
S1		P ← P+1+ISE Q,R ← Rd	S5

	DO3		DO4	NEXT
S5	SYNC-I ← 1	-SYNC	L ← P	S1
	OMA INIT	-PMA		
	LINK ← 0	-LKR		
	LINK ← 1	-LKS		
	FRGND ← 0	-BMS		
	FRGND ← 1	-FMS		
	ISE ← 0	-INH		
	ISE ← 1	-INE		

REV

Figure 6-29. SYNC, PMA, LKR, LKS, BMS, FMS, INH, INE
Instruction Sequence Chart

S1D02: Q, R ← Rd, P ← P+1+ISE. The S1D02 state is identical for all eight instructions in this group. The destination register replaces the contents of the Q and R-register (this always happens). The P counter is incremented and placed back into P. The sequencer then goes to S5D03.

S5D03: Different events occur depending on the instruction decoded. The eight instructions are described separately in the following paragraphs.

Generate Sync Pulse (SYNC): SYNC-I ← 1. Execution of this instruction places a 240-nanosecond pulse on the SYNC-I line in the I/O cable. The pulse occurs during S5D03. (If fast ROM processing is in process this pulsewidth is 180 nanoseconds.)

Pulse Monitor Alarm (PMA): OMA Initiate. The OMA timer does not begin timing until this instruction is executed for the first time. Thereafter, it must be continually executed within 192 ms intervals to prevent the OMA from becoming activated.

This instruction resets the Operations Monitor Alarm timer each time the instruction is executed. Failure to execute this instruction within 192+0, -1 milliseconds after the previous execution activates the OMA. With the OMA activated, the computer automatically switches from run mode to the idle mode and the safe signal is removed from the safe line (SFEC-I).

Link Reset (LKR): LINK ← 0. This instruction places a 0 into the LINK indicator.

Link Set (LKS): LINK ← 1. This instruction places a 1 into the LINK indicator.

Background Mode Set (BMS): FRGND ← 0. This resets the foreground flip-flop and sets the processor into the background mode. Following the execution of this instruction, the eight background registers A', X', Y', Z', B', C', D' and E' are used by all instructions that specify registers. The foreground registers A, X, Y, Z, B, C, D and E are not addressable while the computer is operating in the background mode. If the foreground/background option is not installed, machine is always in background mode.

Foreground Mode Set (FMS): FRGND ← 1. This sets the foreground flip-flop. Following the execution of this instruction, the eight foreground registers A, X, Y, Z, B, C, D and E are used by all instructions that specify registers. The background registers A', X', Y', Z', B', C', D' and E' are not addressable while the processor is operating in the foreground mode. With the foreground/background option installed, the processor is placed into foreground mode by pressing the console SYSTEM RESET switch.

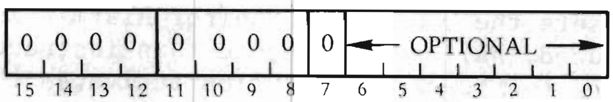
Interrupt Inhibit (INH): ISE ← 0. The interrupt system enable indicator is reset by this instruction. Following the execution of this command, normal program sequencing will not be interrupted when receiving interrupt requests from the Relative Time Clock, the Serial Controller or the attached external devices. The power-fail and automatic restart interrupts cannot be inhibited.

Interrupt Enable (INE): ISE ← 1. The interrupt system enable indicator is set by this instruction. Following the execution of this instruction, interrupt requests from the Relative Time Clock (if armed), from the Serial Controller (if armed) and from attached external devices (if armed) may interrupt normal program sequencing to demand servicing by an associated interrupt subroutine.

S5DO4: L ← P. The S5DO4 state is common to all instructions in this group. The contents of incremented (in S1) P counter replaces the contents of the L-register.

6.12 WAIT INSTRUCTION

The WAIT instruction is of the following format. Refer to Figure 6-30 for the sequence chart associated with the wait instruction.



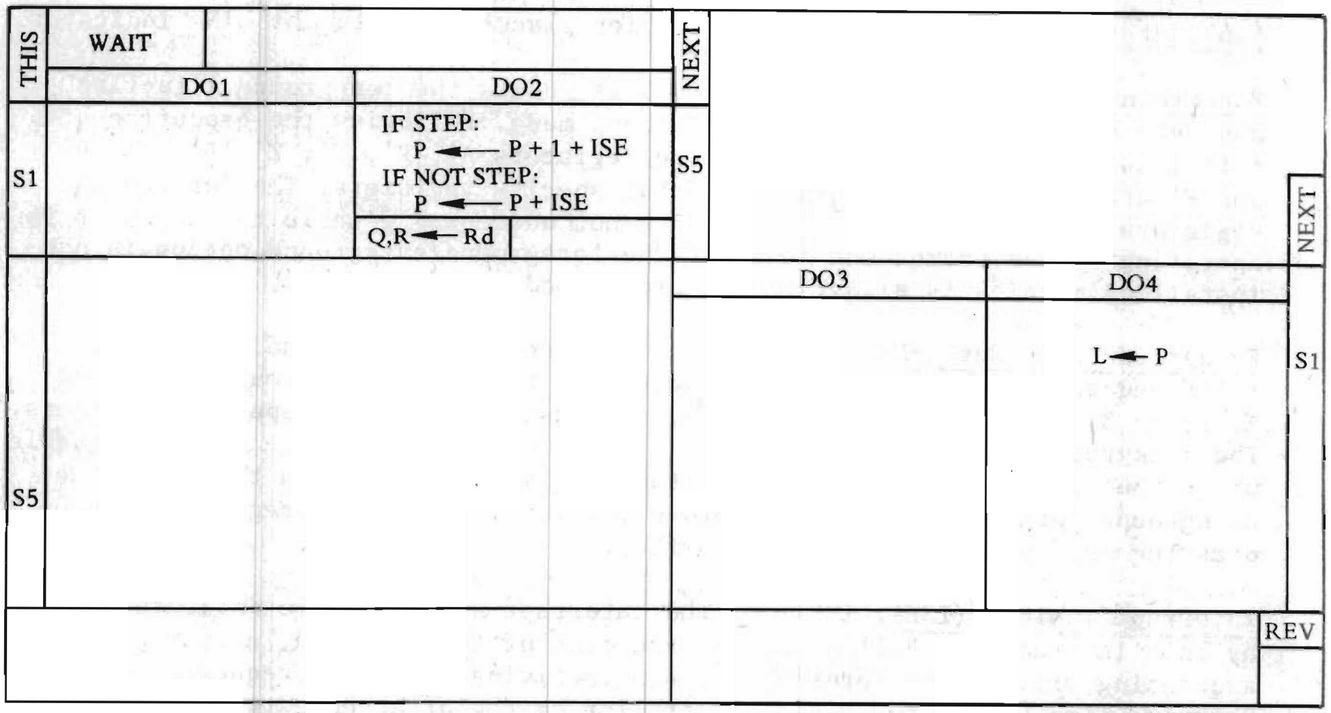


Figure 6-30. Wait Instruction Sequence Chart

This instruction causes the processor to discontinue incrementing the P-register, hence it halts at the P location of the WAIT command. External timing continues to operate. Both interrupts and DMA requests can be serviced while the processor is in the WAIT mode.

S1D02: Q, R ← Rd. In S1D02, if the STEP switch has been activated, the P counter is incremented (P+1) and placed back into P. The sequencer then goes to S5D03.

If the STEP switch is not activated, the processor remains in the wait mode because the P counter is not incremented in S1D02, but is sent through the ALU and placed back into P, and the same memory location is accessed.

S5D04: L ← P. The contents of the P counter replace the contents of the L-register and the sequencer goes to S1.

6.13 MULTIPLY, MPY (OPTION)

The optional hardware multiply has the following characteristics:

a. Fixed point
b. Positive numbers only
c. Variable length

The registers involved in the multiply instruction are shown below.

Register	Function
A	Multiplicand (15 bits)
C	Multiplier (0-15 bits)
B,C	Product (A) x (C), (0-30 bits)

The op code (hex) is:

15	12	8	4	0
0	0	8	n	

n = number of bits in C which are multiplied. All 15 bits are used.

In a multiply instruction n must be large enough to ensure that all "1" bits in C (prior to multiplication) are shifted out (right) after n shifts.

The instruction sequence chart associated with the multiply instruction is shown in Figure 6-31.

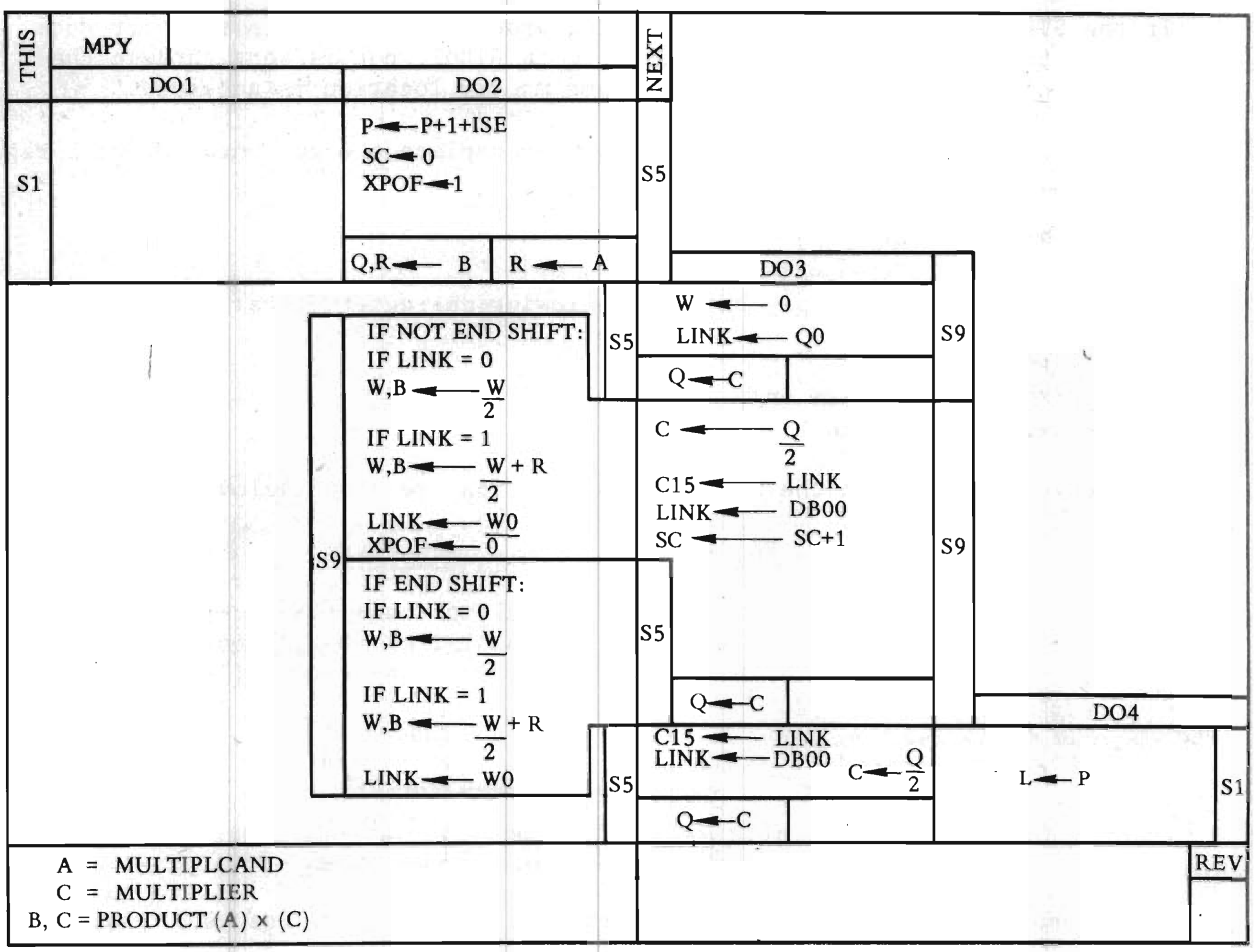


Figure 6-31. MPY Instruction Sequence Chart

S1DO2: Q, R ← B, P ← P+1+ISE, SC ← 0, XPOF ← 1, R ← A. The contents of the B-register replace the contents of the Q and the R-registers even though this data will not be used. The P counter is incremented (P+1) and is placed back into P. The shift counter is reset to zero and the ZPOF flip-flop is set. The contents of the A-register replace the contents of the R-register. The multiplicand (A) is now in R. The sequencer then goes to S5.

S5DO3: Q ← C, W ← 0, LINK ← Q0. The contents of the Q-register are replaced with the contents of the C-register. This places the multiplier in Q.

The W-register is zeroed so it can be used as a working register in "building" the product. No indirecting or indexing is done in this instruction so the contents of the W-register are of no concern to us at this point, but it will be used in S9. Bit 0 of the Q-register (the multiplier) is placed into the LINK indicator. The sequencer then goes to S9.

S9DO2: Different events occur in S9 depending on whether or not an end shift signal (ENDSR) has occurred. If an end shift signal has not occurred, the events described below occur in S9DO2. The sequencer then goes to S9DO3. Certain events, described later, occur in DO3, then the sequencer goes back to S9DO2. This loop continues until the end shift signal occurs and the sequencer goes to S5DO3.

If LINK contains a zero, the W-register is shifted right one place and placed back into W and the B-register. In the sequence chart W/2 indicates a right shift.

If LINK contains a one, the W-register is shifted right one place and added to the R-register (which contains A). The sum is placed into the W and the B-register.

Bit 0 of the W-register (before being set to the sum) is placed into LINK and the XPOF flip-flop is reset. The sequencer then goes to S9DO3.

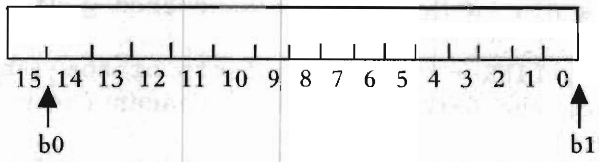
S9DO3: Q ← C, C ← Q/2, C15 ← LINK, LINK ← DB00, SC ← SC+1. The contents of C replace the contents of Q. The Q-register is shifted right one place (Q/2) with LINK entering bit 15 and placed back into C. Bit 0 of the data bus is placed into LINK. The shift counter is incremented and placed back into the shift counter. The sequencer then goes back to S9DO2.

S5DO3 with $\overline{\text{XPOF}}$: Q ← C, C ← Q/2, C15 ← LINK, LINK ← DB00. The contents of C replace the contents of Q. The Q-register is shifted right one place with LINK entering bit 15 and placed into C. Bit 0 of the data bus is placed into LINK. The sequencer then goes to S5DO4.

S5DO4: L ← P. The updated P counter replaces the contents of the L-register and the sequencer goes back to S1.

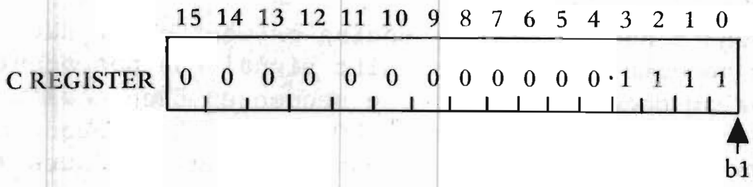
EXAMPLE OF MULTIPLY INSTRUCTION

The op code of a multiply instruction is 008n. The A-register binary point begins between bits 14 and 15 as shown below and goes from left to right.



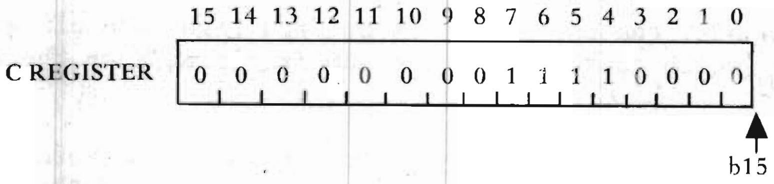
The C-register maximum binary point is defined by the value in bits 0-3 of the multiply instruction.

Assume that the op code is 8F and that the C-register contains the following:



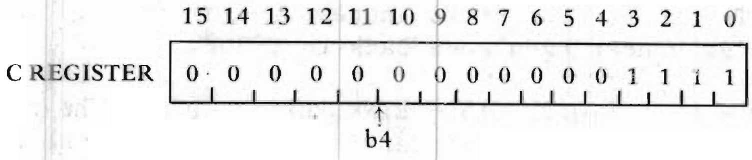
Therefore the C-register contains F@15.

Again assume the multiply op code is 8F, but this time the C-register contains the following:



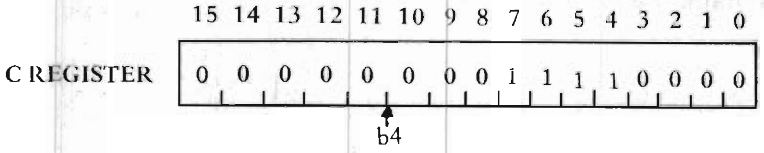
The C-register contains F0@15.

Assume the multiply op code is 84 and the C-register contains the following:



The C-register contains F@4.

Assume the multiply is op code is 84 and the C-register contains the following:



In this case, the C-register contains a negative number (-0) which is illegal.

In the following example, assume the conditions in the following paragraph. Refer to Figure 6-31 for the graphic development of the product of a sample multiply problem.

Prior to the execution of the instruction the A and the C-registers must be set up as follows. We are not concerned with the B-register at this time.

```
Op code = 008F
Set A-register = 000F@15
Set C-register = 000F@15
```

Keep in mind that the LSB of C is not used in product.

Refer also to the multiply sequence chart in Figure 6-31. In S5D03, the contents of C replace the contents of Q. So now Q and C contains 000F as shown in Figure 6-32. The L3 column in Figure 6-32 means the state of the LINK indicator after D03. The Q-register bit 0 contains a 1 and this is placed in LINK at S5D03 and the W-register is reset. The sequencer then goes to S9D02.

In S9D02, end shift has not occurred since the instruction was just entered, and LINK contains a 1. The W-register shifted (zeros at this time) is added to R (which contain A-register, 000F). The sum of this addition is placed in B. The B-register could contain anything the first time through. L2 means the value of link after D02. W-register bit 0, which is a 0, is placed into LINK and the XPOF flip-flop is reset. The sequencer then goes to S9D03.

In S9D03, the contents of C replace the contents of Q. The value of the shift counter is 0 at the time. The Q (C) register is shifted right one place and now contains 0007 and a 1 is placed into LINK. The B and the W-register contain 000F. The shift counter is incremented by 1, compared to the I-register and the sequencer goes to S9D02. This comparison determines when the desired count in the shift counter is reached.

In S9D02 the W-register is shifted right one place (now contains 0007) and W0 is placed into LINK. The shifted value in W is added to R (contains A-register, 000F). The sum

```
   0000 0000 0000 0111
 + 0000 0000 0000 1111
 ---------------------
   0000 0000 0001 0110
```

is placed in the W and the B-registers and the sequencer goes to S9D03.

This loop continues until the shift counter is equal to 15. B-register contains zeros, LINK is zero and C contains the partial product. B and C-registers must be shifted one more time in S9D02 and in S5D03·$\overline{\text{XPOF}}$. The value in C is now 1C2 which is the correct answer.

ENTER S5·DO3·XPOR

	B REG																L	C REG (IN Q)																L
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	After DO2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	After DO3
W, B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\boxed{X}$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	$\boxed{1}$
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0▲	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0▲	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0	0	0	0	1	1
3	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0▲	0

A = 000F @ 15 Product (in B and C) = 0000 01C2
C = 000F @ 15 = E1 @30
MPY CMD = 008F
▲ = binary point in C

Figure 6-32. Multiply Example

The product in the B and the C-register is:

```
        B      C

      0000   01C2

        = E1 @ 30
```

The sequencer then goes to S5D04 to terminate the multiply instruction.

NOTE

Refer to the SPC-16 System Reference manual (88A00243A) for a different example of the multiply instruction.

6.14 DIVIDE, DIV (OPTION)

The op code for divide is:

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
0	0	A	n

where n is equal to the number of bits in the quotient, that is, the number of shift and subtraction operations. In a divide instruction, n must be large enough to ensure that all 1 bits in C (prior to division) will be shifted (left) out after n shifts.

The registers used in a divide instruction are as follows:

Register	Function
A	Divisor
B	Dividend, most significant
C	Dividend, least significant
B	Remainder (after division)
C	Quotient = (B,C)/(A)

NOTE

The number in B before division must be less than the number in A. The remainder R is formed such that after division, multiplying the quotient by the divisor and adding the remainder to the least significant half (C-register) with any carry bit out of C added to B will result in the original dividend in B and C.

Refer to the SPC-16 system reference manual (88A00243A) for an example of a hardware divide.

The instruction sequence chart associated with the divide instruction is shown in Figure 6-33.

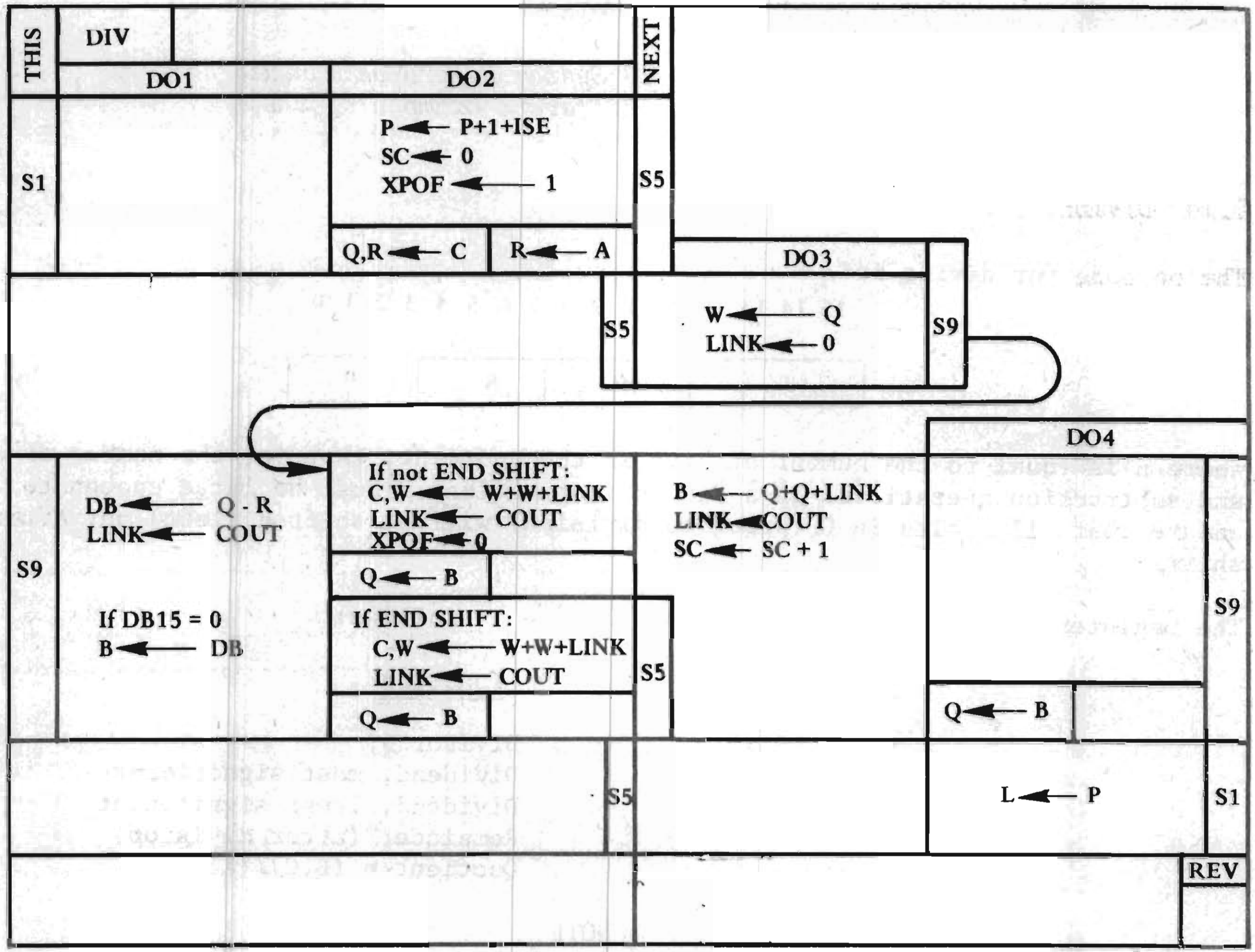


Figure 6-33. DIV Instruction Sequence Chart

S1D02: Q, R ← C, P ← P+1+ISE, SC ← 0, XPOF ← 1, R ← A. The contents of the C-register replace the contents of the Q and the R-registers. The P counter is incremented, the shift counter is zeroed, the XPOF flip-flop is set and the contents of the A-register replace the contents of the R-register. The sequencer then goes to S5D03.

S5D03·XPOF: W ← Q, LINK ← 0. The contents of Q (which contains C) replaces the contents of the W-register and the LINK indicator is reset to zero. The sequencer then goes to S9D02.

S9D02: The first time through the end shift signal has not occurred. Q ← B, C, W ← W+W+LINK, LINK ← COUT, XPOF ← 0. The contents of the B replace the contents of Q. The W-register is shifted left one place (W+W) and placed into the C and the W-registers. The W+W function is accomplished by selecting W into the augend and controlling the ALU with the necessary control functions to cause the augend to add to itself. The LINK indicator is added to it. The carry out (result of the addition) is placed into LINK and the XPOF flip-flop is reset. The sequencer then goes to S9D03.

S9D03: B ← Q+Q+LINK, LINK ← COUT, SC ← SC+1. The Q-register is shifted left one place, the left shift of Q is accomplished by selecting Q into the augend and controlling the ALU with the necessary control functions to cause the augend to add to itself. LINK is added to it and Q is then placed into B. The carry out is placed into LINK. The shift counter is incremented, compared to the number in I, and placed back into the shift counter. The sequencer then goes to S9D04.

S9D04: Q ← B. The contents of B replace the contents of the Q-register. The sequencer then goes back to S9D01.

S9D01: DB ← Q-R, LINK ← COUT. If DB15=0, B ← DB. Q-R is done and placed on the data bus. The carry out is placed into LINK. If DB15=0, the contents of the data bus replace the contents of the B. DB15=0 represents a successful division of the divisor into the partial dividend. If DB15=0 then this step of the division indicates that the divisor is greater than the partial dividend. Only if the division step is successful is the result written into the B-register. This division approach is referred to as non-performing division. The sequencer then goes to S9D02.

S9D02: Assume at this point that the sequencer has gone through the S9D01, S9D02, S9D03 and S9D04 loop the designated number of times (as specified by the I-register). This time, assume the end shift signal has occurred. Q ← B, C, W ← W+W+LINK, LINK ← COUT. The contents of B replace the contents of Q. The W-register is shifted left one and LINK is added to it. The sum (W+W+LINK) is placed into the C and the W-register. Carry out is placed into link. The sequencer then goes to S5D03·XPOF.

S5D03·$\overline{\text{XPOF}}$: Nothing happens. Sequencing advances to S5D04.

S5DO4: L ← P. The incremented P counter replaces the contents of the L counter. The sequencer then goes back to S1.

6.15 INTERRUPT (S7) AND CYCLE-STEAL (S8) SEQUENCE STATES

The interrupt and cycle-steal sequence states are shown in Figure 6-34. Any instruction can be interrupted at its completion (i.e., after it completes sequence state 5) except JMP and JSR.

In any sequence state, if there is a cycle-steal request (DMCR), at the end of DO3 the sequencer activates S8. The "stolen" cycle lasts from S8·DO4 through S8·DO3, at which time the previously active sequence state DO4 is continued.

"Turn" indicates that fast (overlapped) processing is taking place. The conditions for turning are ROM and non-memory-referencing.

THIS	INTERRUPT AND CYCLE-STEAL SEQUENCING				NEXT		NEXT
	DO1	DO2	DO3			DO4	
ANY S		NORMAL ACTIVITY	DMCR:	NORMAL ACTIVITY	S8·DO4		
S5		NORMAL ACTIVITY	$\overline{\text{DMCR}}$:	$\overline{\text{TURN}}$·$\overline{\text{XPOF}}$· ($\overline{\text{SR}}$I+ENDSR)	S5 ·DO4	$\overline{\text{IREQ}}$: L ← P	S1·DO1
				TURN	S1· DO2		
				XPOF	S9·DO2	IREQ: INB L ← DMA	S7·DO1
				SRI·$\overline{\text{ENDSR}}$	S5.DO2		
S7		E,P ← P+ISE	ISE ← 0 P ← M			P, L ← P	S1·DO1
			$\overline{\text{DMCR}}$		S7·DO4		
			DMCR		S8·DO4		
S8		M ← M OPERATE I/O OR I/O ← M	$\overline{\text{DMCR}}$		RETURN TO ACTIVE STATE	L ← INB	S8·DO1
			DMCR		S8·DO4		

Figure 6-34. Interrupt and Cycle-Steal Sequencing

SECTION 7
SPC-16 MEMORY

7.1 INTRODUCTION

The General Automation SPC-16 memory is comprised of random-access lithium ferrite core modules in 4K, 8K and 16K sizes. Systems that use 4K, and either 8K or 16K modules may only use one 4K module.

A functional block diagram of a 4K memory module is shown in Figure 7-1. The functional arrangement of an actual 4K or 8K memory module is given in Section 3.8.

The remaining parts of this section describe the theory and implementation of the 4K memory module only, but most of the information is directly applicable to 8K modules as well.

7.2 MAGNETIC CORE OPERATION

The storage element in the SPC-16 memory is a magnetic toroidal core. The core is of ferrite material forming a closed flux path that can be saturated to either a positive or negative flux density by a current applied through the center of the toroid. Because of the high percentage of magnetic flux that remains in the core after it has been saturated (remanence), and because of its sharp, rectangular hysteresis loop, this material allows permanent storage and efficient high-speed operation. Graphically, the formation of the hysteresis loop of a typical magnetic core is shown in Figure 7-2.

With the core unmagnetized, an increasing magnetizing force (H) increases flux density (B) on an S-shaped curve. As the core nears saturation, an increase in H causes little change in B because the core cannot support a greater flux. When the magnetizing force returns to zero, the flux in the core remains nearly the same; the core has "remembered" the magnetizing force and its positive polarity.

When the magnetizing force is again applied, but in the negative direction, the flux in the core changes very little until the field strength reaches the coercive force of the material. At this point, the core flux switches very rapidly from positive saturation to negative saturation. Again, the flux is retained after the negative magnetizing force is removed; the core has "remembered" the magnetizing force and its negative polarity.

In the magnetic core memory, the states of positive and negative flux saturation are used to represent binary "1" and "0", respectively. The state of the core can be changed by simply passing a current in the direction desired through the center of the core. A change in the state of the core can be sensed by means of the current induced in a line through the center of the core. Thus, we have the means of controlling both the sensing and storing of information in the core.

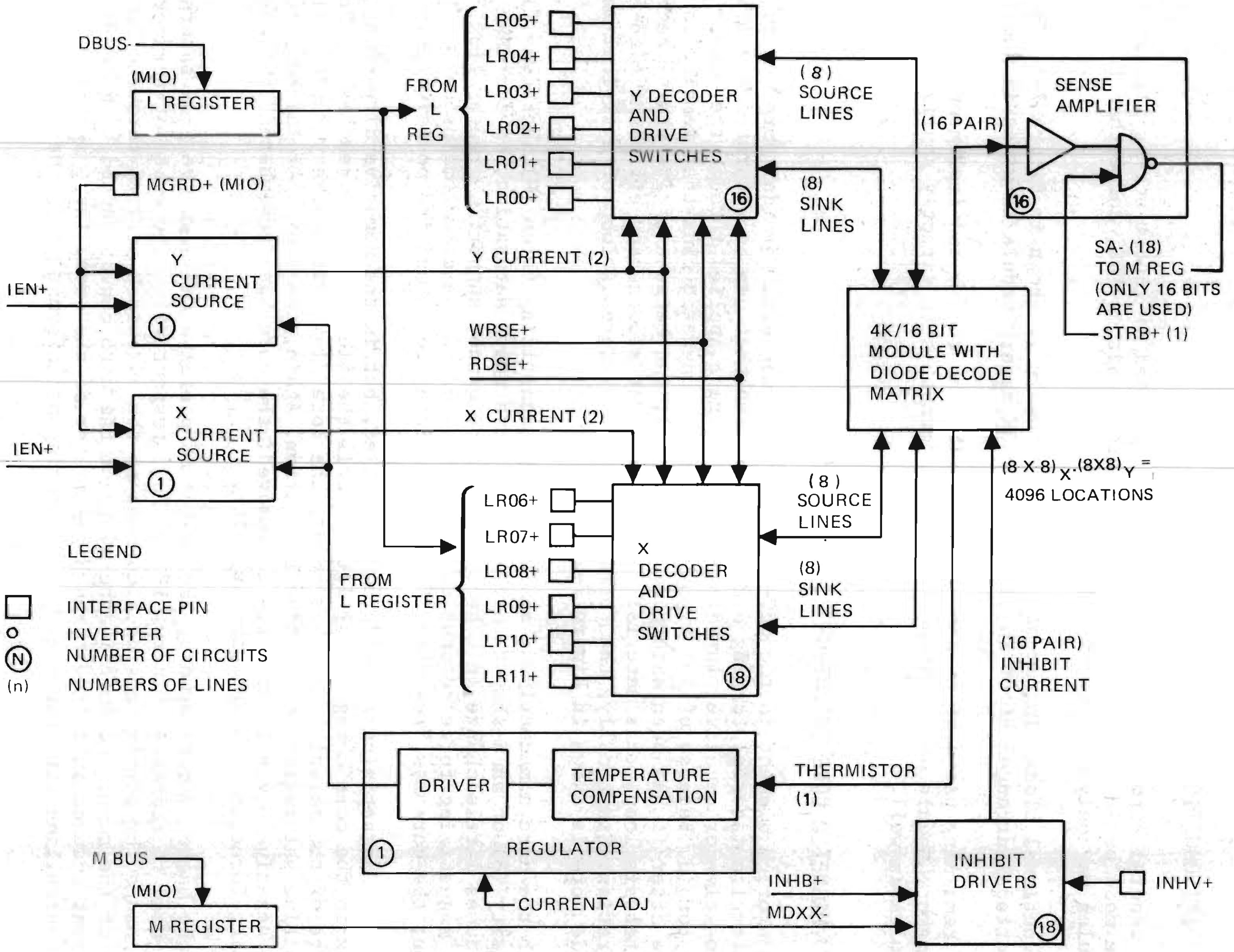
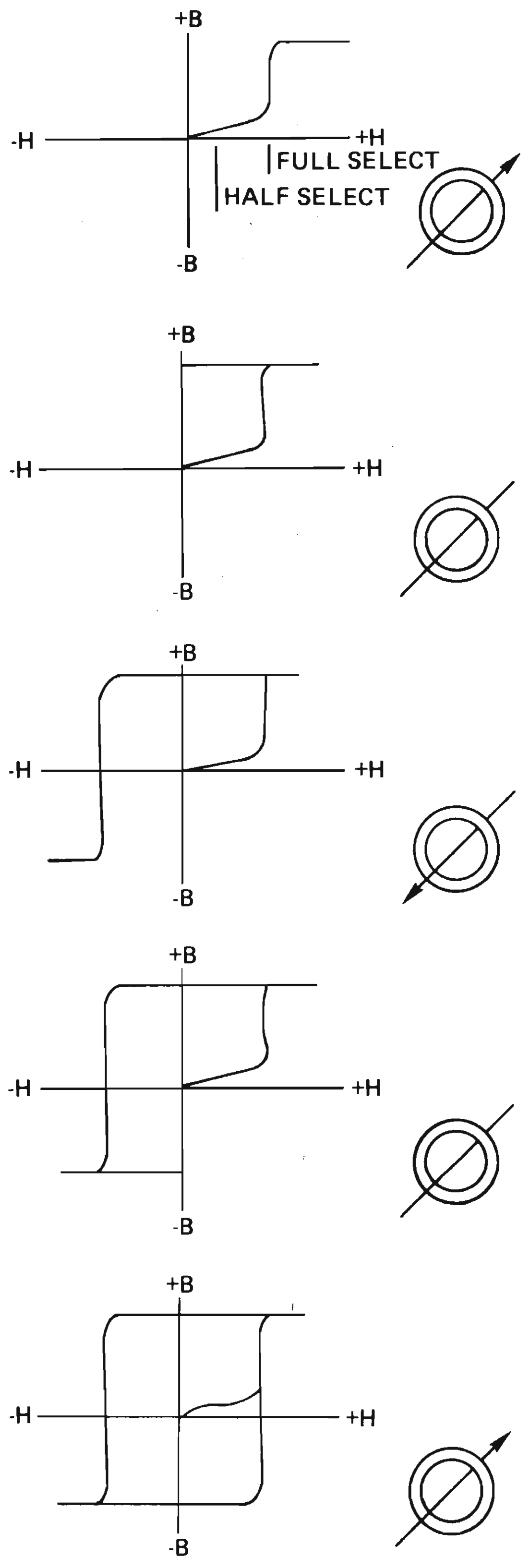


Figure 7-1. Functional Block Diagram of Memory



Full select magnetizing current increases flux density to saturation.

When current is reduced to zero, most of the flux remains.

When negative full select magnetizing current reaches coercive strength, the core is switched to negative saturation.

When current is reduced to zero, most of the flux remains.

When positive full select magnetizing current reaches coercive strength, the core is switched back to positive saturation. In memory operation, the core never returns to zero flux.

Figure 7-2. Ferrite Core Hysteresis Loop

7.2.1 Memory Core Control Wires

Through each core pass three wires (see Figure 7-3), each of which carries current that either affects or reflects changes in the state of core magnetization:

a. An X-drive line passes through all the cores in the same row.

b. A Y-drive line passes through all the cores in the same column.

c. A sense/inhibit line passes through all the cores in a mat.

The magnitude of each of the X-drive current and Y-drive current is half the value required to switch a core to the opposite state. When both currents are pulsed through a core in the same direction at the same time, switching will occur. A bit is read by applying the coincident X-drive and Y-drive currents in the direction that sets the core to the binary "1" state. A bit is written by applying the coincident currents in the opposite direction so that the core is set to the "0" state.

7.2.2 Memory Cycles

The writing of a bit may be inhibited by applying inhibit current through the sense/inhibit line in coincidence with, but in the opposite direction of, the X and Y drive current. The inhibit driver produces full select current which is equally divided in the two sections of the sense/inhibit line transformer. When a bit is inhibited, the net current through the core is -1/2 select, which is inadequate to switch the core to the "0" state.

When a bit is read, a voltage is induced in the sense/inhibit line if the core switches from the "0" to "1" state (corresponding to a stored "1"). If a bit is read and the core is in the "1" state (corresponding to a stored "0"), a very small voltage is induced in the sense/inhibit winding. The sense amplifier discriminates between a stored "1" and "0" and produces a digital output signal.

During a read cycle, the timing logic turns on the X and Y read drive switches associated with the address bus, turns on the X and Y current sources, clears the memory data register and strobes the sense amplifier at the time when the signal/noise ratio of the core output is greatest.

During a write cycle, the timing and control logic turns on the X and Y write drive switches associated with the address bus, turns on the X and Y current sources, and turns on those inhibit drivers that are associated with a zero bit in the memory data register.

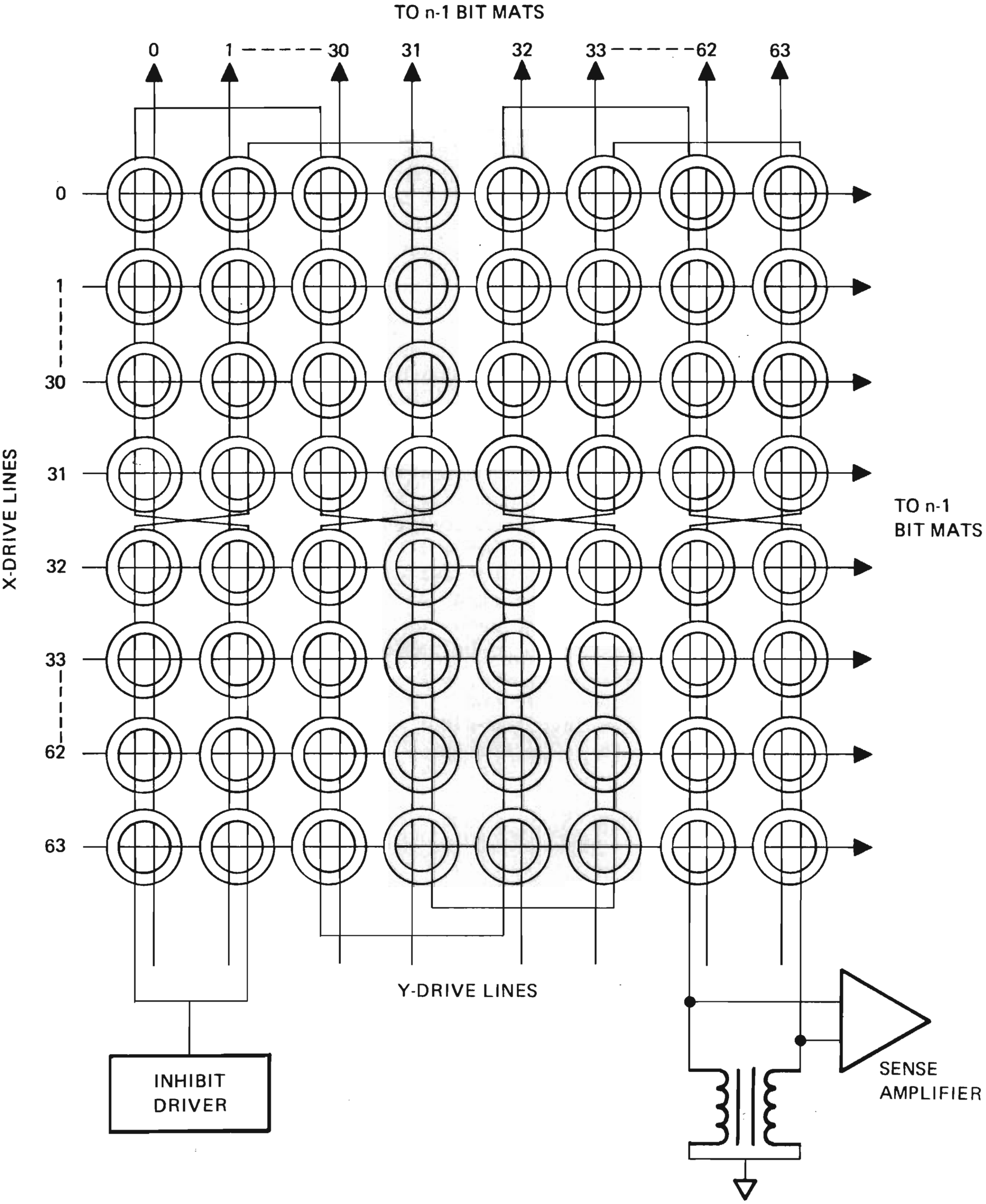


Figure 7-3. Memory Core Mat

7.3 DECODER AND DRIVE SWITCHES

There are 64 X-drive lines and 64 Y-drive lines, forming a 64 x 64 square matrix. The selection of a bit is accomplished by addressing two of 128 lines (one X and one Y), thereby defining one of 4096 intersections. Similarly, each group of 64 X- and Y- drive lines can be addressed from an 8 x 8 matrix of drive lines forming 64 intersections. This is accomplished by diode-decode matrices. In the system, a diode matrix board is an integral part of the memory assembly. The effect is to minimize the number of lines required for interfacing to the core modules.

Figure 7-4 shows the schematic of a typical X- or Y-drive line. To read the indicated line, the read source and read sink switches are closed, forcing 1/2 select current in the indicated direction. Conversely, to write, the write source and write sink switches are closed, forcing 1/2 select current in the opposite direction.

Figure 7-5 shows a typical source switch operating in conjunction with a binary-to-octal decoder. The sink switch schematic is identical except that RDSE and WRSE are transposed. The system is implemented such that LR00+, LR01+ and LR02+ select the Y source switches; LR03+, LR04+ and LR05+ select the Y sink switches; LR06+, LR07+ and LR08+ select the X source switches; and LR09+, LR10+ and LR11+ select the X sink switches. One of the eight outputs of the binary-to-octal decoder is grounded which is dependent upon the binary representation of the address bus input. If a read cycle is to be performed, RDSE+ goes positive, forcing current through R_1, the primary of T_1 and into the decoder sink. The current in the primary of T_1 is coupled to the secondary and turns on Q_1, allowing current from the current source to flow into the diode decode matrix and module. Conversely, if a write cycle is to be performed, WRSE+ goes positive and turns on Q_2.

7.4 REGULATOR AND CURRENT SOURCES

7.4.1 Regulator

The regulator provides reference voltages (XRV and YRV), shown in Figure 7-6, to the current sources that control the final value of the X and Y drive currents. CR1 generates a zener reference that is divided by R4 and R5. R5 is a potentiometer that is used to adjust the ambient X and Y drive currents. The adjustable voltage from R5 is applied through R6 to the base of Q_1. R6, R2, R3, THERMISTOR A35-B35 and XYDR+ bias Q_1.

If module temperature rises, THERMISTOR resistance increase, which increases the base bias on Q_1 and reduces the output drive currents. A change in XYDR+ likewise effects the output drive currents.

Resistor R7 and THERMISTOR A31 and B31 are used to regulate the input 15V by maintaining a constant current drain on the 15V. If the 15V decreases, less current will flow through the THERMISTOR, causing it to cool, decreasing the THERMISTOR resistance, which causes the 15V level to rise. This permits more current to flow through the THERMISTOR, causing its resistance to rise and thereby stabilizing the 15V level.

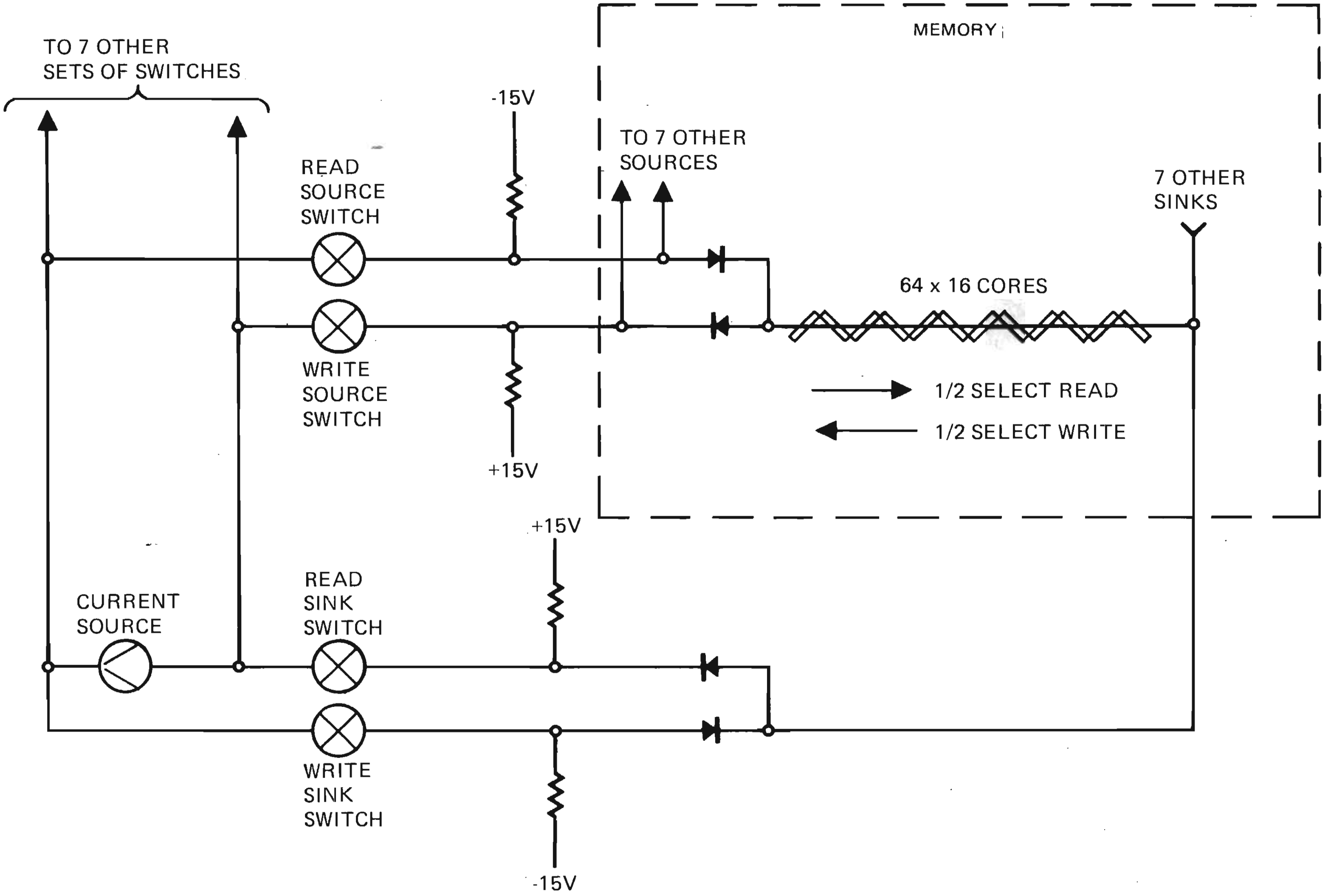


Figure 7-4. Typical X or Y Drive Line

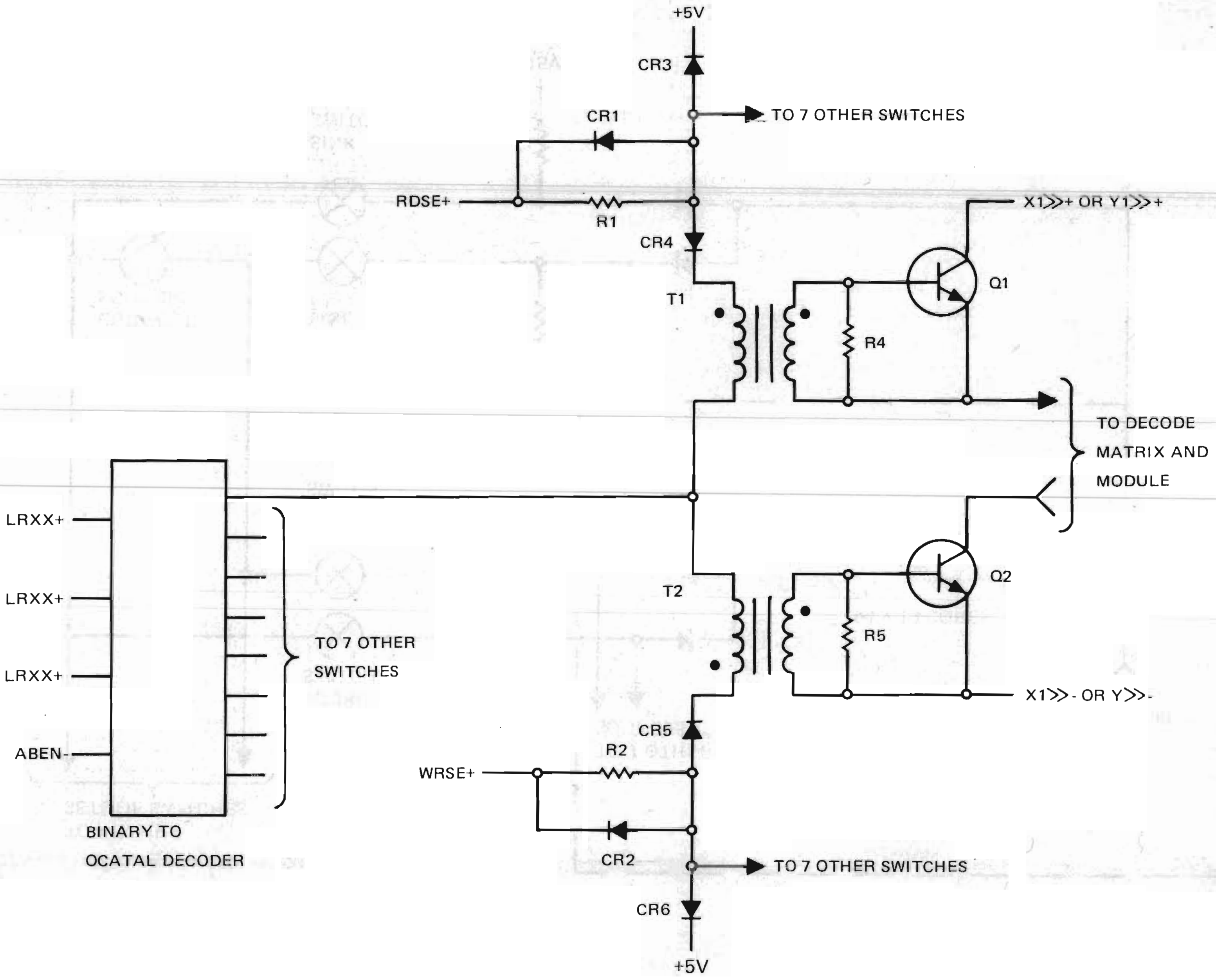


Figure 7-5. Source Switch Schematic

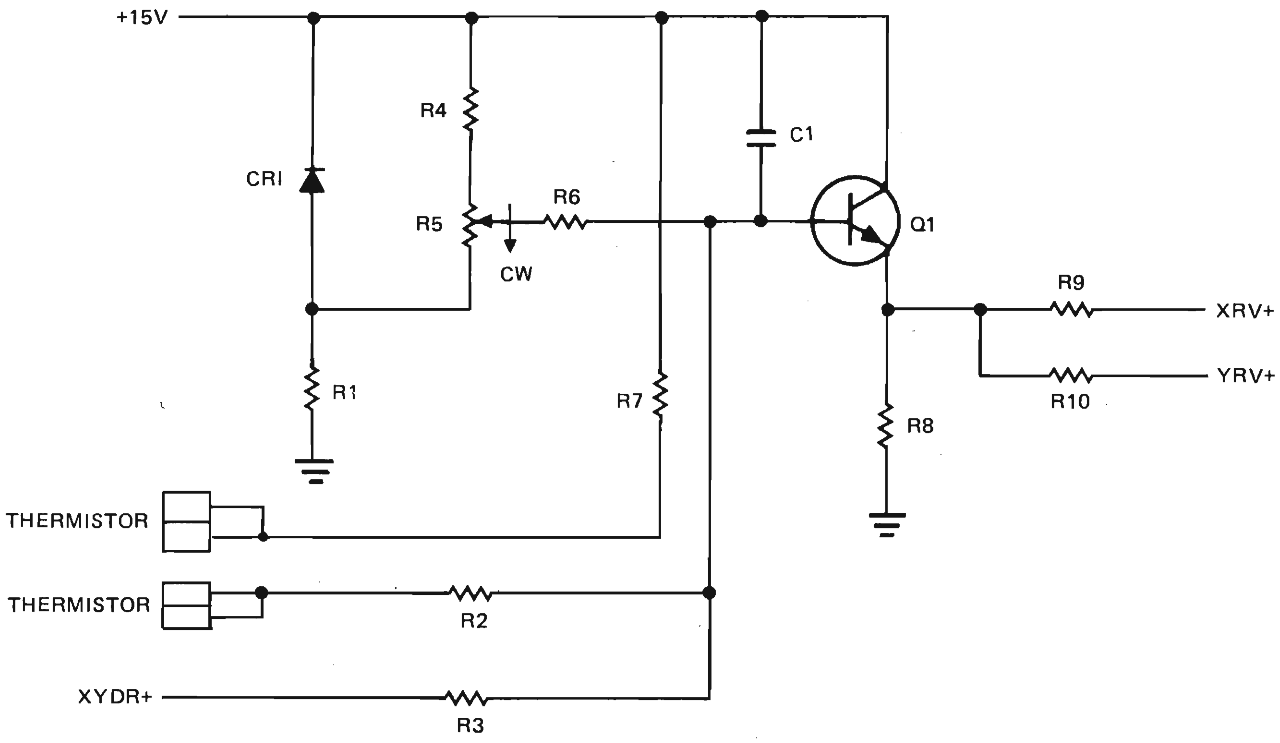


Figure 7-6. Regulator Schematic

7.4.2 Current Sources

Figure 7-7 shows the X current source which is functionally identical to the Y current source. The current source provides 1/2 select current (nominally 400 ma) through the X drive lines of the module. The current is gated by IEN, adjustable by XRV, and disabled by MGRD+. In addition, the current waveform has a controlled rise time (nominally 60 nanoseconds) which is adjusted by XRV such that it reaches final value at the same time regardless of current magnitude.

7.4.3 Quiescent Conditions - Current Source Off (Power at Nominal Voltage Levels)

Under quiescent conditions, MGRD+ is at -15V and forces Q_2 off. Q_1 and Q_4 are current sources with magnitude dependent upon XRV with respect to +15V. The voltage at the base of Q_3, with respect to -15V, is proportional to the final value current. IEN+ is low, turning on Q_5 and forcing Q_6 off. Q_7 is on and the voltage at the base of Q_8 is slightly more positive than -15V by virtue of bias current through R_{13}, CR_2, CR_3 and Q_7. Under these conditions, Q_8, Q_9, Q_{10} and Q_{11} are off and XI>>+ and XI>>- are at zero current.

7.4.4 Current Turn On

The MGRD is still at -15V. When IEN+ goes high, Q_5 goes off, Q_7 goes off Q_2 goes on and the current Q_4 charges C_2 in a linear positive ramp with respect to -15V. The current through Q_8, Q_9, Q_{10}, and Q_{11} also increase in a linear ramp. When the voltage across C_2 with respect to -15V is at final value, the current from Q_4 is clamped through R_9. CR_1 and Q_3 and the current through Q_8, Q_9 Q_{10} and Q_{11} ceases to increase. CR_7, CR_8 and R_{23} damp the natural resonant frequency of the stack. The common mode transformer, T_1, forces a constant current through the module determined by Q_8, Q_9, Q_{10} and Q_{11}.

7.4.5 Current Turn Off

The MGRD is still at -15V. When IEN+ goes low, Q_5 turns on, Q_7 turns on and C_2 is discharged through CR_2, CR_3 and Q_7 thereby turning off Q_8, Q_9, Q_{10} and Q_{11}. CR_6 and R_{22} damp the natural resonant frequency of the stack on turnoff and R_{21} and CR_5 allow the magnetizing inductance of T_1 to recover.

7.4.6 Memory Guard (Power Not at Nominal Voltage Levels)

MGRD floats when the power is not at nominal levels. Q_2 goes on and the base of Q_8 is clamped to one diode drop more positive than the -15V level by virtue of CR_1, regardless of XRV, +15V, IEN or the magnitude of -15V. Under these conditions Q_8, Q_9, Q_{10} and Q_{11} are off and no current can flow through the stack.

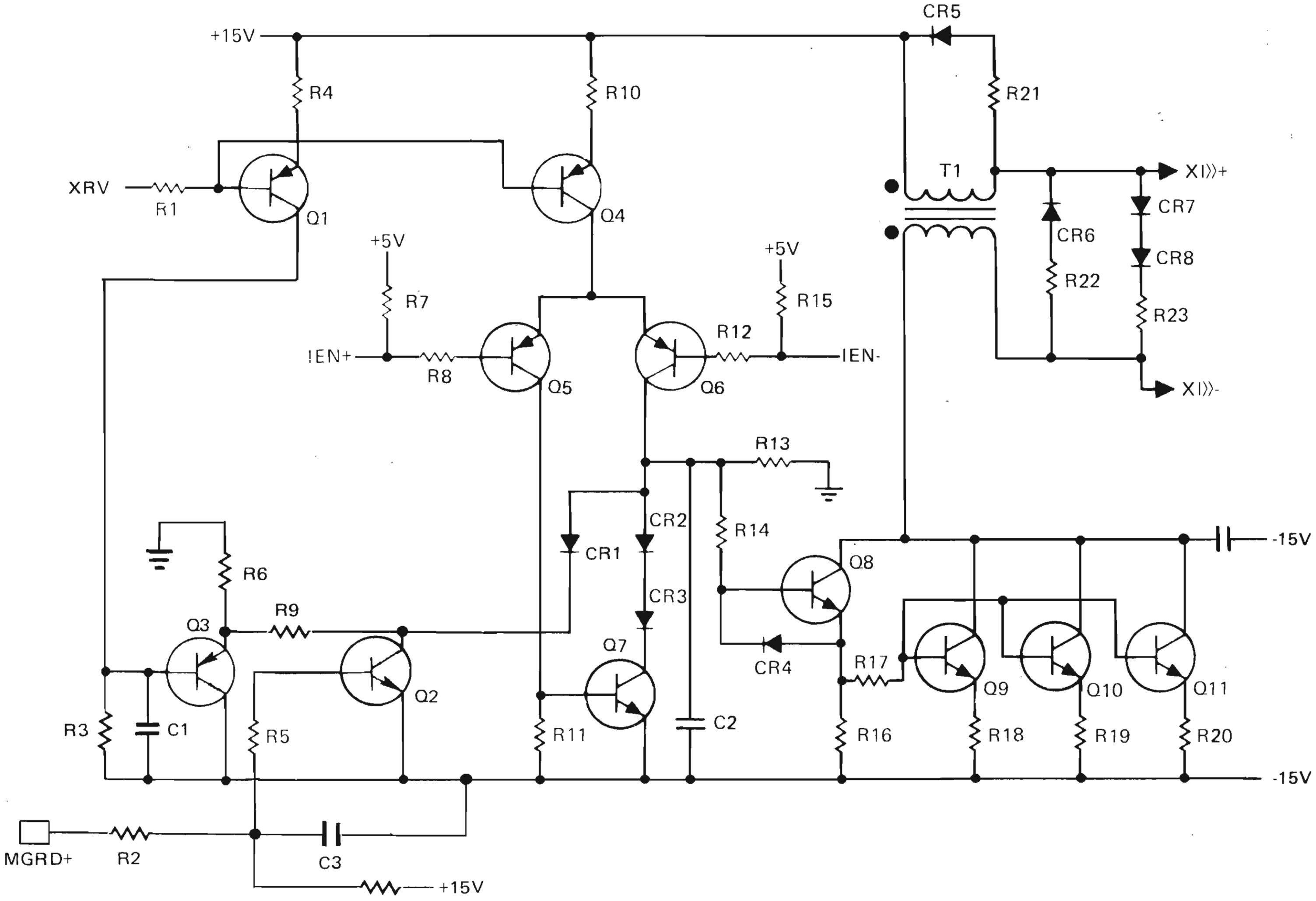


Figure 7-7. X Current Source Schematic

7.5 SENSE AMPLIFIER AND INHIBIT DRIVERS

The sense amplifiers and inhibit drivers operate in conjunction with common bit lines in the module. Figure 7-8 shows the inhibit driver, module sense line and sense amplifier for a single bit. If a memory data bit is false during a write cycle, the inhibit control flip-flop (INHB+) turns on Z_1. Current flows through the common base amplifier, Q_1, which turns on Q_2 and Q_3. The inhibit current flows from +INHV through R_4 and R_5 into the core module. The magnitude of the inhibit current is determined by the parallel combination of R_4 and R_5 and the absolute magnitude of the inhibit voltage +INHV.

During a read cycle, Q_2 and Q_3 are off. If a core switches in the 4096-core mat when the X and Y drive currents are applied, the core output is applied to the sense amplifier. The core output is amplified, threshold detected by the adjustable threshold voltage (THRV) and ANDed with the strobe (STRB+) in the sense amplifier. Z_3 inverts the sense amplifier output and applies it to a flip-flop of the memory data register. R_9 and R_{10} terminate the differential output of the core mat.

7.6 TIMING AND CONTROL

The memory timing is shown in Figure 7-9. The MSTC+ memory start signal is a synchronizing pulse sent to memory from the Timing Control Board.

NOTE

All data transfer between the memory and processor is accomplished during DO1 and DO2 times.

7.7 MEMORY LOGIC SIGNAL MNEMONICS

The memory logic signal mnemonics are listed in Table 7-1.

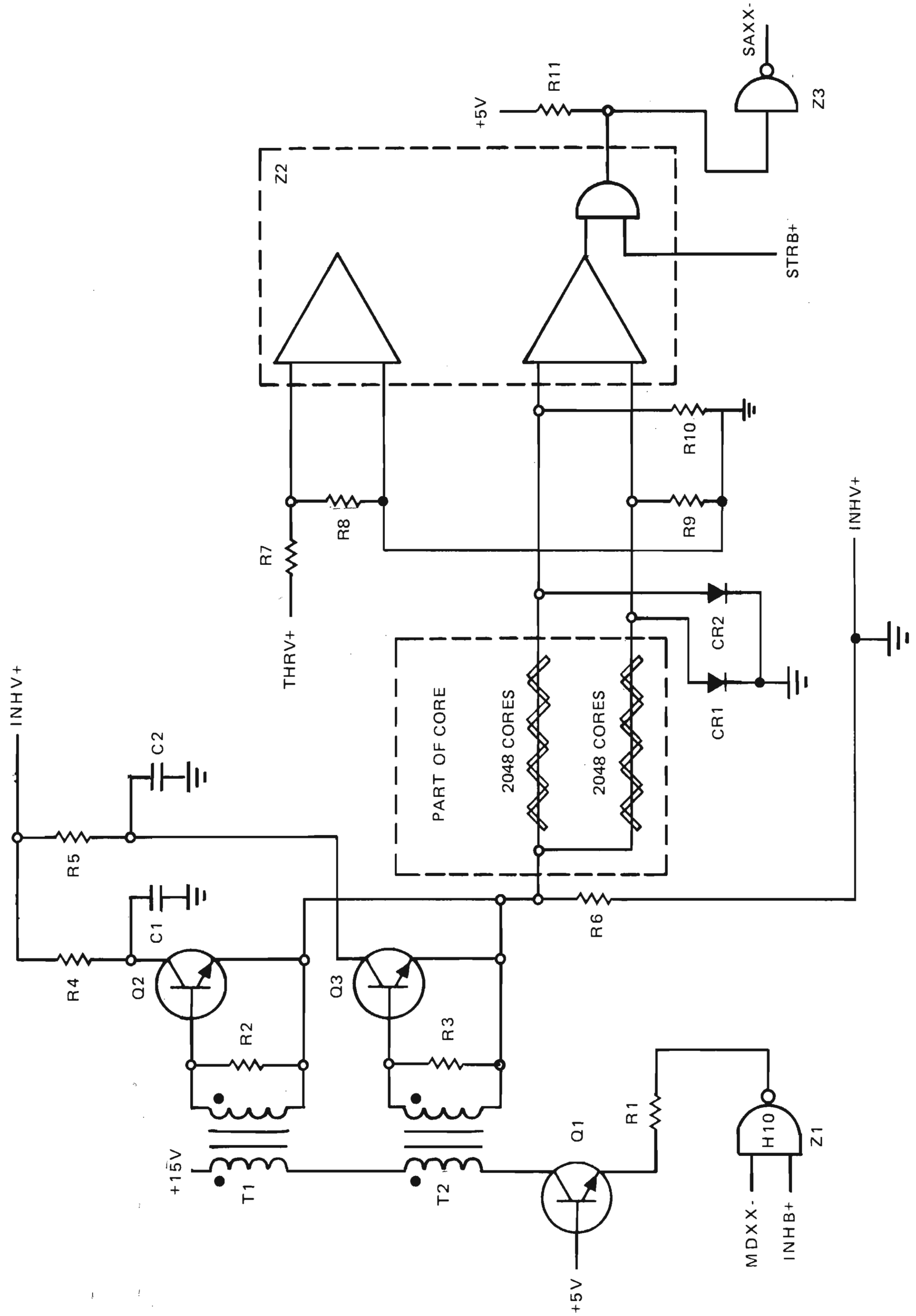


Figure 7-8. Sense Amplifier and Inhibit Driver Schematic

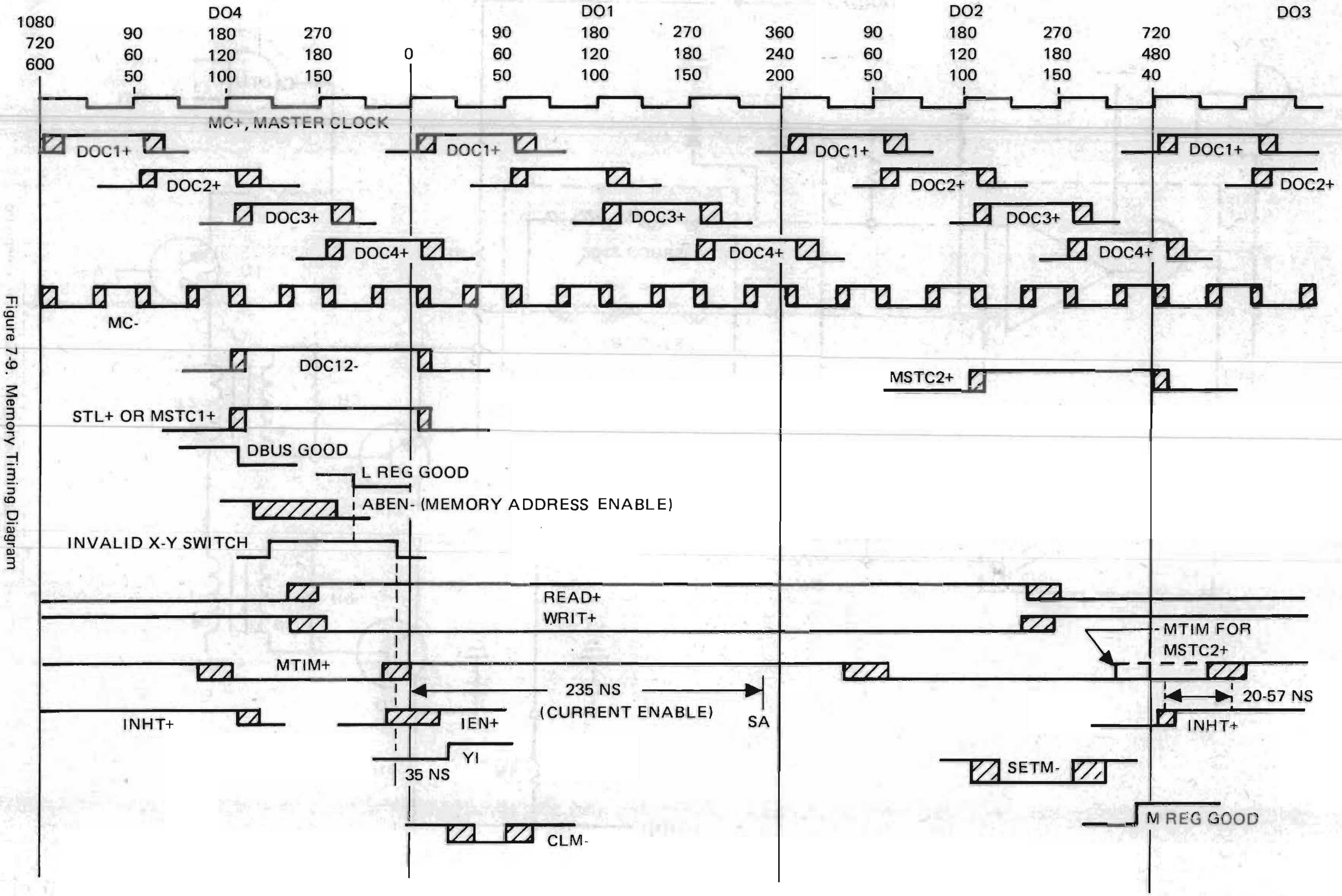


Figure 7-9. Memory Timing Diagram

Table 7-1. Memory Board Signal Mnemonic List

Mnemonic	Definition
ABEN	Address buffer enable
IEN	Drive current enable
INHA	Inhibit time, control A
INHB	Inhibit time, control B
INHV	+20V inhibit driver voltage
INHT	Inhibit time
LR00 through LR10	Location register bits
MCYC	Memory cycle
MD00 through MD17	Memory register bits
MGRD	Memory guard relay. The presence of this signal on the memory boards inhibits X- and Y- drive current to the memory module(s).
MSTA	Memory stack select
MSTC	Memory start clock
MTIM	Current source enable
RDSE	Read switch enable
READ	Read out of memory
SA00 through SA17	Sense amp bits
STRB	Strobe sense amps
THRV	Threshold voltage
WRIT	Write into memory
WRSE	Write switch enable
X0A through X57	X drive sink output to module
X0A through X7A	X drive anode output to module

Table 7-1. (Continued)

Mnemonic	Definition
XOC through X7C	X drive cathode output to module
XI>>	X drive current
XYDR	X, Y drive
XY15	15V to X and Y lines
XYRV	X, Y temperature compensated regulator voltage
YOA through Y7A	Y drive anode output to module
YOC through Y7C	Y drive cathode output to module
YI>>	Y drive current
Y50 through Y57	Y drive sink output to module

SECTION 8
SPC-16 POWER SUPPLY

The SPC-16 processor power supply is designed to supply regulated dc power at the following voltage and current levels:

+5V @ 25A max. full load
+15V @ 5A max. full load
-15V @ 5A max. full load
+20V @ 6A average full load

8.1 POWER FAIL DETECT CIRCUIT

The power fail detect circuitry is housed on the power failure detector module. The inputs to this module are unregulated +5V and ±15V. The outputs are the power fail detect (PFD>-) signal and the restart (RS>>+) signal. During power fail both PFD>- and RS>>+ are at ground.

Figure 8-1 is a timing diagram illustrating the manner in which a power failure sequence is initiated if an interruption of the AC line voltage causes DC voltage in the computer to fall below prescribed operating levels. The power failure detection circuit monitors the unregulated DC voltage to detect the improper power condition. If the AC input voltage drops below 105 vac for more than 1 millisecond, the power failure detection circuit triggers the power shutdown sequence.

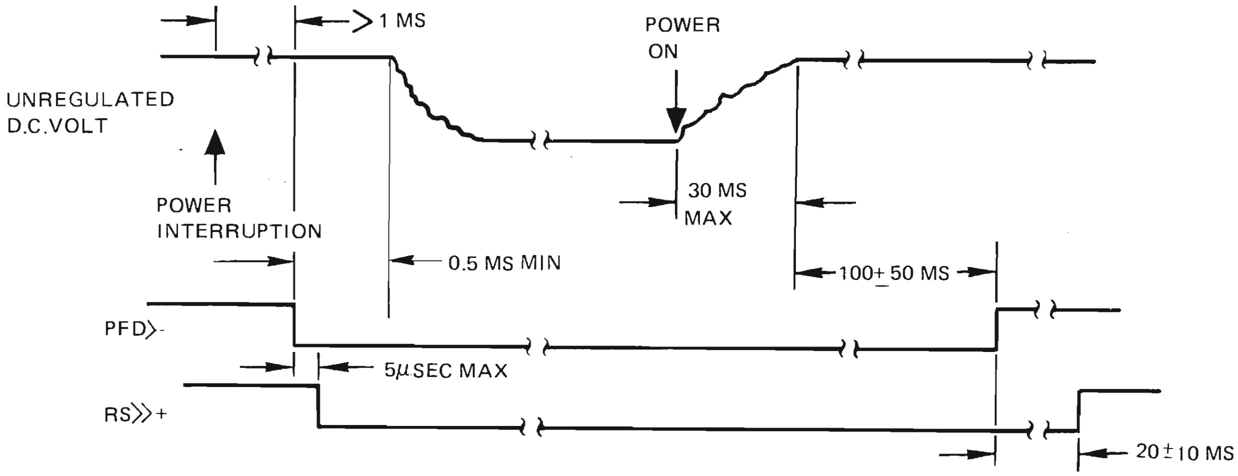


Figure 8-1. Timing Diagram of Power Fail/Automatic Restart

The power shutdown sequence is initiated at the end of the instruction that is currently being executed. This sequence places the computer in idle mode, which stops all memory cycles and then disables the memory drive circuits. Once a power shutdown sequence has been started, the full power-failure/automatic-restart operation must be completed. This operation takes a minimum of 60 and a maximum of 210 milliseconds to complete (depending on the duration of the transient or failure).

When the AC input voltage drops below 105 vac (200 vac European) for longer than 1 millisecond, the power failure detection circuit requests an interrupt and triggers the power shutdown sequence. At the time the interrupt is requested, the OMA timer is initialized to time-out in 100 R/W memory cycles and any further PMA instructions or DMA requests are ignored. Therefore, after the power-fail interrupt request, the program has 100 R/W memory cycles (minus the time of the longest instruction) to execute whatever shutdown routines it requires. After 100 memory cycles, the OMA times out and the machine goes through a normal alarm sequence.

The automatic restart feature of the computer generates an interrupt each time a power-on sequence is initiated.

SECTION 9
PREVENTIVE MAINTENANCE PROCEDURES

General Automation systems require little routine or preventive maintenance. The electro-mechanical peripheral units incorporated in the system are subject to normal wear from moving parts and should be lubricated and checked at regular intervals. Basic preventive maintenance schedules for peripheral equipment commonly used in General Automation systems are provided in the OEM manuals sent with the system.

9.1 CPU PREVENTIVE MAINTENANCE

The preventive maintenance required to ensure optimum performance of the SPC-16 computer is highly influenced by environmental conditions and the use to which the unit is subjected. Periodic checks, visual inspection and cleanliness are the basis for preventive maintenance.

The first step in any preventive maintenance schedule is careful and thorough visual inspection for signs of dirt, wear, cracks, binds, loose connections or loose hardware.

Accumulations of dust and dirt probably indicate malfunctioning of the fans. Such accumulations in electronic circuits act as an insulating blanket by preventing efficient heat dissipation and trap moisture. The former condition causes overheating and subsequent component breakdown; the latter condition allows signal crosstalk. If a fan becomes inoperative, it should be replaced immediately.

9.1.1 Fan Removal and Replacement

Removal and replacement of the fan assembly is the same for all SPC-16 models. Two exhaust fans mounted on the inside of the chassis side-panel provide air flow cooling for the processor components.

Remove the fans according to the following procedure (see Figure 9-1):

1. Gain access to the fans by removing six screws that fasten the fan assembly side-panel to the chassis.

2. Pull the side-panel out and remove AC-line cord and grommet from the grommet slot on the edge of the MIB.

3. Provide enough slack in the AC-line cord to allow the side-panel to lie flat.

4. Disconnect the AC-line cord from the connectors of the fan to be replaced.

5. Remove the four screws that hold the fan to the side-panel and lift off the fan assembly.

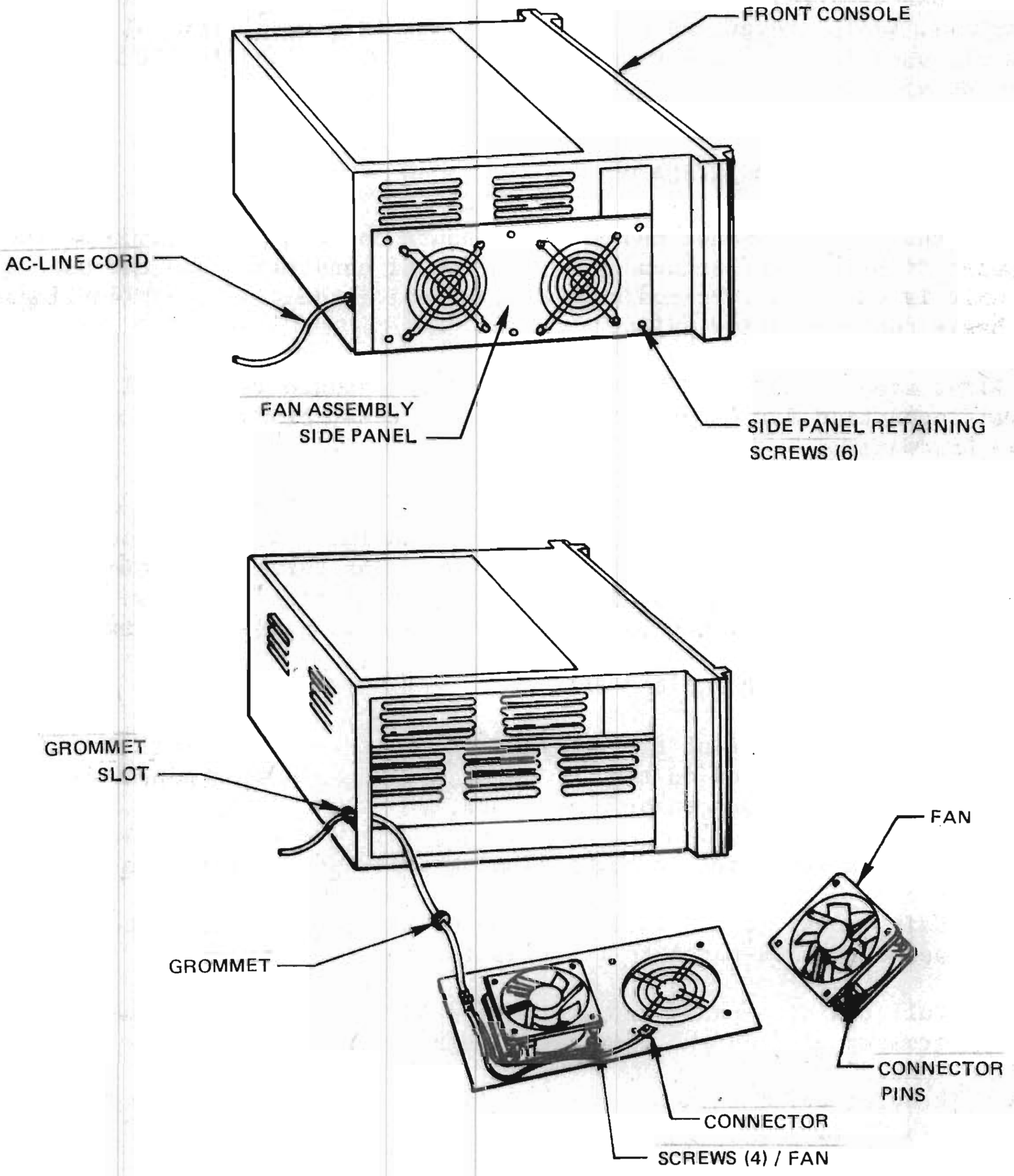


Figure 9-1. Removing Fans (All Models)

A new fan can be installed and the side-panel reassembled by performing this procedure in the reverse order. Be certain to mount the new fan with its AC power connector pins accessible to the AC-line cord connectors.

9.1.2 Cleaning the CPU

Use a small brush moistened with isopropyl (90% or more type) to clean CPU components of dust, dirt, and grime. AVOID use of other chemical cleaning agents, NEVER use water on any computer system component. If a brush is found to be impractical to clean some component, Kim-wipes or Q-tips may be substituted.

9.2 POWER SUPPLY OUTPUT CHECK AND ADJUSTMENT

Power supply output voltage levels within the SPC-16 Central Processing Unit should be verified in the course of routine or preventive maintenance as well as during troubleshooting. The +5V, +15V and -15V levels may be checked by measuring certain external pin locations on the Master Interconnect Board.

Each voltage level should be within ±2% of the specified value. If a voltage is found to lie beyond these tolerances, an internal power supply adjustment is required.

PROCESSOR POWER SUPPLY VERIFICATION

The output voltage levels of the Processor Power Supply can be easily verified by measuring the voltages at the DC Power Input Connector (J17) on the rear of the SPC-16 chassis. Pin assignments for the J17 connector are as shown in Figure 9-2.

Pin	Signal
1	+20V
2	COMM
3	COMM R
4	+20V
5	COMM
6	COMM
7	COMM
8	COMM
9	RS>>+
10	+15V
11	+5V
12	PFD>-
13	-15V
14	+5VR
15	+5V

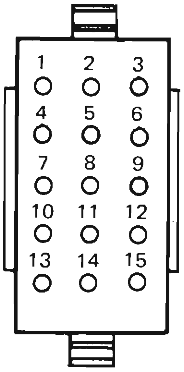


Figure 9-2. Pin Assignments; DC Power Input Connector

Each of the four voltage output levels (+5V, +15V, -15V and +20V) must be within +2% of the specified level. The pins to be checked are summarized in Table 9-1.

Further, the power supply outputs must have a peak-to-peak AC ripple of less than 0.5%. These limits, also shown in Table 9-1, should be verified with an oscilloscope. The CPU should be in the RUN mode when checking ripple.

Table 9-1. Power Verification Points

Pin Number*	Nominal Voltage (+2%)	Allowable AC Ripple, Peak-To-Peak
J17 2	Ground	
J17 11	+5V (+4.9V to +5.1V)	25mV
J17 10	+15V (+14.7V to +15.3V)	75mV
J17 13	-15V (-14.7V to -15.3V)	75mV
J17 1	+20V (+19.6V to +20.4V)	100mV

*NOTE

These pin numbers apply to both the SPC-16 40/60/80 and 45/65/85 series.

If the above AC ripple limits are exceeded, the power supply should be replaced.

If an output level adjustment is required, a separate voltage adjustment potentiometer is provided in the power supply for each output (+5V, +15V, -15V and +20V).

To gain access to the potentiometers:

1. Remove the power supply assembly from the rack or cabinet in which it is mounted.

2. Remove the bottom cover of the power supply. The potentiometers are now accessible through the bottom of the power supply assembly.

3. Measure the output at J17 with a VOM or digital voltmeter and, if any output registers outside +2% nominal value specified for that output, adjust the appropriate potentiometer to bring the value within the range. Figure 9-3 illustrates the four potentiometers provided for this adjustment.

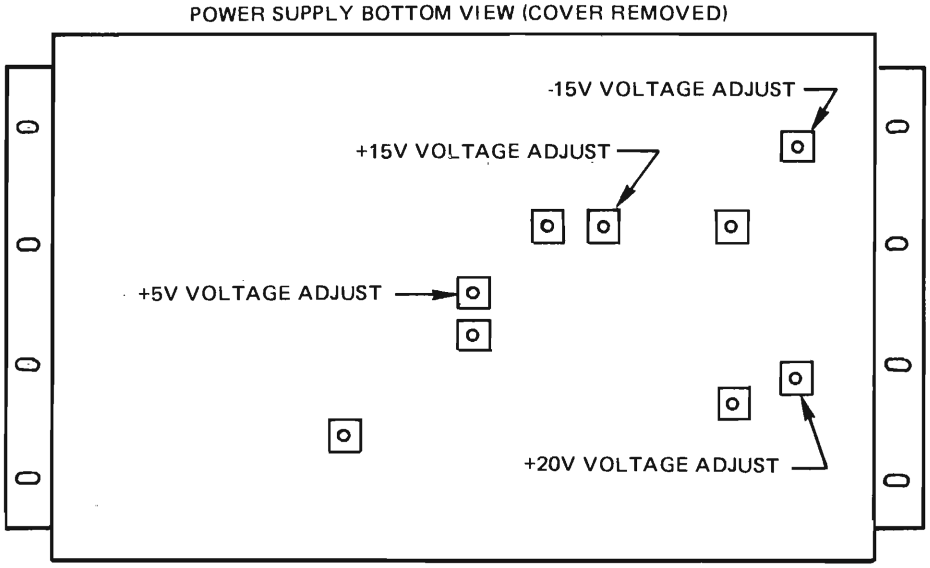


Figure 9-3. Power Supply Adjustment Potentiometers

SECTION 10
CORRECTIVE MAINTENANCE PROCEDURES

Malfunctions may occur in one of three areas of an SPC-16 system: in the processor, in an Input/Output controller, or in a peripheral unit. This manual deals only with maintenance of the processor; separate General Automation documentation is available for controller maintenance and each Original Equipment Manufacturer supplies a maintenance manual for his own product.

The processor troubleshooting procedure in this section is listed in the order an experienced technician would be expected to perform a troubleshooting assignment. That is, after observing the nature of the malfunction, the technician would make a visual inspection of all external cable connections, power and panel indicators. If the existing malfunction is not detected in the course of these preliminary checks, the following procedure must be implemented until the malfunction is located:

- Run available test programs (Section 10.1).
- If the malfunction involves a DMA I/O device, check the Data Channel (Section 10.2).
- Perform an operational check of the processor using the Display Console (Section 10.3).
- Check power supply (Section 9.2)
- Replace processor boards with available spares (Section 10.5).

The above are arranged in decreasing priority. That is, since the test programs often serve to pinpoint a malfunction with minimum troubleshooting time invested, they are the logical first step in many maintenance procedures. If the test programs load and execute properly, they can indicate a malfunctioning processor board almost immediately. If the Teletype bootstrap will execute, the Teletype Test and Verify program should be run. Errors indicated by this program require maintenance to the Teletype itself, in most cases. The Processor Test and Verify program and the Memory Test Program should be run if the computer is giving erroneous results. Both programs "point to" faulty processor boards.

If the malfunction involves input/output with high-speed peripheral units connected to the Data Channel Module, the test procedure given in Section 10.2 should be used.

If the test programs will not load or will not execute properly, the operational check routine given in Section 10.3 should be run. This routine consists of an investigation of data transmission between the processor's registers. It also investigates the integrity of a few simple memory reference instructions.

A "dead" system (i.e., when all operation has ceased) indicates the need to check power, as described in Section 10.4, and adjust if necessary. However, if the CPU has power and none of the above methods have succeeded in curing the malfunction, spare boards should be substituted for those in the processor on a one-for-one basis, as described in Section 10.5.

Section 10.6 describes the methods for removing and replacing subassemblies within the CPU.

Although the premise of this manual is to troubleshoot malfunctions to the board level, the technician may troubleshoot to the chip level by using the sequence diagrams given in Section 6 in conjunction with General Automation's logic diagrams. The diagrams are as follows:

Board	Assembly Number	Logic Drawing Number
Console	31D01446A	90C01446A
Timing	31D01811A	90C01811A
Macro	31D01331A	90C01331A
Arithmetic	31D01333A	90C01333A
MIO	31D01628A	90D01628A
MIB (SPC-16) 40/60/80)	31D01640A	90D01640A
MIB (SPC-16 45/65/85)	31D01648A	90D01648A
Mem.Expansion MIB (SPC-16 45/65/85)	31D01650A	90D01650A
4K Memory	31D01635A	90D01635A
8K Memory	31D01594A	90D01594A
16K Memory	31D01783A	90D01783A

The following tools are recommended for on-site testing and repair of malfunctioning subassemblies:

a. Tektronix Model 453 dual-trace oscilloscope or equivalent.

b. Triplett VOM Model 630-A multimeter.

c. Standard Field Service tool kit.

Refer to General Automation Workmanship Standards, document number 88A00119A, Section 8, for complete printed circuit rework standards.

If on-site repair is impractical, faulty boards may be returned to General Automation for repair. If it becomes necessary to ship a piece of equipment to General Automation, the greatest care should be given to repacking.

HEXADECIMAL NUMBER REPRESENTATION

Throughout the remainder of this section, patterns of data bits are represented in a hexadecimal numbering code. The hexadecimal code is a method of representing data in groups of four bits. The rightmost bit of the group has a value of 1, the second bit has a value of 2, the third has a value of 4, and the leftmost bit has a value of 8. So any four-bit group can represent base 10 numbers between zero (bit pattern '0000') and fifteen (bit pattern '1111').

The code to represent this range of numbers if given below:

Base 10 Number	Bit Pattern	Hexadecimal Value
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Numbers having a base of 16 (i.e., "hexadecimal" numbers) are formally represented with a subscript 16 just as the "standard" numbers, base 10 ("decimal"), are represented with a subscript of 10 and binary numbers are given a 2 subscript. So, it can be written that

$$14_{10} = 1110_2 = E_{16}.$$

In this section, an X and lower-case quotation marks are used in place of subscripts to denote hexadecimal notation. For example, CDEF_{16} is written X'CDEF'.

Since most of the numerical references in this section are to 16-bit registers or the 16-bit Console Data Entry switches, four-place hexadecimal notation (such as X'FFFF' or X'CDEF') occurs frequently. Keeping in mind that each hex place represents four bit positions, it is evident that a four-place hex number represents sixteen bit positions. For example, X'CDEF' represents bit settings 1100 1101 1110 1111 and X'29B3' represents bit settings 0010 1001 1011 0011.

10.1 USING THE DIAGNOSTIC PROGRAMS

The first step in most maintenance routines involves running three test programs. These programs are:

- SPC-16 Teletype Test & Verify Program, which performs a programmed sequence of functional tests of the ASR-33 or ASR-35 Teletype controllers and units.
- SPC-16 Processor Test & Verify Program, which performs a programmed sequence of functional tests of the SPC-16 series Central Processing Units.
- SPC-16 Memory Test Program, which performs a programmed sequence of functional tests to exercise portions of memory not tested by the Processor Test and Verify Program.

10.1.1 Program Loaders

Programs may always be loaded into the computer's memory manually by entering the individual instruction codes and data via the console. However, if a program that reads data from a peripheral unit and places it into memory in a sequential fashion is already in memory, it may be executed and used to load another program automatically.

Two data formats are used for programs: bootstrap binary and PGS binary. Most system software, as well as the object tapes of user programs that have been assembled or compiled, are in PGS format. The Teletype Test and Verify Program is in PGS format. The smallest loading program that will accept programs in PGS format is the PGS Loader; it consists of more than 250 instructions and it is, therefore, impractical to load manually. The Basic Utility System, BUS-16 (see Subsection 10.2.1), will also load PGS format tapes.

A few basic system programs, including the PGS Loader program, are supplied in simple binary format. Verification software is generally supplied in paper tape form. Subsequent discussions will deal only with paper tape. A binary tape consists of a sequence of data bytes, each representing half an instruction or data word. These bytes may be input via an appropriate peripheral unit (teletype paper tape reader or high speed paper tape reader) and stored in sequential words (two data bytes per word) in memory. The program that performs this operation is known as a bootstrap loader.

The bootstrap loader that loads binary tape format from the teletype (Teletype Bootstrap) consists of only eight instructions; thus, it may be easily keyed in from the computer console. The bootstrap loader that loads binary tape format from the high speed paper tape reader (High Speed Paper Tape Reader Bootstrap) consists of 23 instructions; it may be keyed in manually or loaded from the teletype using the Teletype Bootstrap (the High-Speed Paper Tape Reader Bootstrap program is supplied on paper tape in binary format). Either bootstrap loader may be used to load the more complex PGS loader, which is also supplied in binary format. Those systems supplied with a High Speed Paper Tape ROM will load binary tapes.

LOADING THE TELETYPE BOOTSTRAP

The Teletype Bootstrap will be loaded into the eight memory locations X'0051' through X'0058'.

Hexadecimal	Instruction	Operand
0620	ZERO	X
0524	LOOP DSPL	X
10FF	TEST	RDY,TY
73FE	JMP	$-1
18BF	DTIR	A,TY
9900	STBY	A,0,X,1
072E	INCR	X
73F9	JMP	LOOP

In the discussion which follows,the circled numbers refer to switches and indicators on the SPC-16 console. These console switches and indicators are discussed in Appendix D.

The procedure for loading the Bootstrap Loader into memory is:

a. Turn CONSOLE ENABLE key lock to the horizontal position. Set RUN/IDLE switch ⑪ to IDLE.

b. Depress SYSTEMS RESET ⑫.

c. Select P-register switch ④.

d. Set Data Entry switches ⑦ to the Bootstrap beginning address minus one (X'0050').

e. Press ENTER ⑧.

f. Disengage P switch ④.

g. Select I-register switch ⑥.

h. Set Data Entry switches ⑦ to hexadecimal X'5000' which is the hexadecimal code for the "STA 0,0" instruction.

i. Press ENTER ⑧ .

j. Select SAVE I switch ⑨ so that the "STA 0,0" instruction will be retained in the I-register for repeated execution.

k. Select the A-register for display by disengaging switches ②through⑥.

l. Set Data Entry switches ⑦ to the hexadecimal code for Loader instruction (see above Loader program listing).

m. Press ENTER ⑧ .

n. Press STEP ⑩ . The Loader instruction will be stored (through STA instruction) at the specified location and the program will be incremented to the next memory location.

o. Repeat steps k. through n. for each of the eight Bootstrap Loader instructions.

p. Disengage SAVE I switch ⑨ .

q. Depress SYSTEMS RESET ⑫ .

Verify that the bootstrap was keyed in correctly by displaying the eight memory locations as follows:

r. Repeat steps b. through j. with the exception that in step h. the Data Entry switches ⑦ should be set to hexadecimal code X'4000', which loads the A-register with the contents of memory as specified by the P-register.

s. Press STEP. The contents of the memory location specified by the P-register will be loaded into the A-register and displayed on the Register Display Indicators ① .

t. Repeat Step s. for each of the 8 memory locations containing a Bootstrap instruction.

u. Disengage SAVE I switch ⑨ .

An incorrect display probably indicates error in loading the bootstrap. To correct coding of the erroneously loaded instruction, the instruction must be reloaded by the above procedure.

EXECUTING THE TELETYPE (TTY) BOOTSTRAP

With the bootstrap in memory, the following steps will be used to load programs using the bootstrap (the circled numbers refer to switches and indicators on the console. They are discussed in Appendix D):

1. Set the D-register to the starting address at which the program is to be loaded. The D-register is selected when the General Purpose Register Select switches ② are set to '110'.

2. Set the P-register to the starting address of the bootstrap, '0051'.

3. Place the program paper tape into the teletype paper tape reader. The first binary frame must be over the read station. Skip over visual header, if any.

4. Press SYSTEMS RESET ⑫ .

5. Set the RUN/IDLE switch ⑪ to RUN.

6. Press STEP switch ⑩ .

7. Turn the reader on.

8. When the program has been loaded, turn the reader off. The loading process doesn't stop after the last data byte has been read but continues until the physical end of the tape is reached or until the reader is turned off.

9. Set the RUN/IDLE switch ⑪ to IDLE.

10. Depress SYSTEMS RESET ⑫ .

The above procedure will be used to load the High Speed Paper Tape Reader Bootstrap (if applicable to the system) as well as the PGS Loader, the Teletype and Processor Test and Verify Programs and the Memory Test Program.

LOADING THE HIGH SPEED PAPER TAPE READER (HSPTR) BOOTSTRAP

If the system includes a high speed paper tape reader, programs read in from the reader must be loaded by the High Speed Paper Tape Reader (HSPTR) Bootstrap.

The HSPTR Bootstrap instructions are as follows:

Hexadecimal		Instruction	
0620	PTB	ZERO	X
0640		ZERO	Y
6007	PTB1	JSR	IBY
2DFE		SKZ	PTB1
9900	PTB2	STBY	A,0,X,1
072E		INCR	X
0859		ADD	Y,A
6002		JSR	IBY
0544		DSPL	Y
73FA		JMP	PTB2
11C8	IBY	TEST	1,X'08'
73FE		JMP	$-1
1308		CTRL	3,X'08'
10C8		TEST	0,X'08'
73FE		JMP	$-1
13C8		TEST	3,X'08'
73FE		JMP	$-1
1888		DTIR	A,X'08'
0815		RTR	A,A
05E3		RTRN	E

If the paper tape version of the HSPTR Bootstrap (70B00PB) is available, it may be loaded into memory by following the procedure given in the preceding paragraph. Starting address for the HSPTR Bootstrap (step 1) is X'0059'.

If the Paper tape is unavailable, the HSPTR Bootstrap may be keyed in by hand. The loading procedure is given in steps a through u of the paragraph titled "Loading the Teletype Bootstrap"; however, the HSPTR Bootstrap's "address minus one" in step d is X'0058'.

EXECUTING THE HSPTR BOOTSTRAP

With the High Speed Paper Tape Reader Bootstrap in memory, programs supplied on binary-formatted paper tape may be loaded as follows:

1. Set the D-register to the starting address at which the program is to be loaded. The D-register is selected when the General Purpose Register Select switches ② are set to '110'.

2. Set the P register to the starting address of the bootstrap, X'0059'.

3. Place the program paper tape into the teletype paper tape reader. The first binary frame must be over the read station. Skip over visual header, if any.

4. Press SYSTEMS RESET ⑫ .

5. Set the RUN/IDLE switch ⑪ to RUN.

6. Press STEP switch ⑩ .

7. Turn the reader on.

8. When the program has been loaded, turn the reader off.

9. Set the RUN/IDLE switch ⑪ to IDLE.

10. Depress SYSTEMS RESET ⑫ .

10.1.2 Diagnostic Test Programs

With the appropriate loader stored in memory, the technician can load and execute one or more of the following GA-supplied diagnostic test programs to isolate an existing malfunction:

- The SPC-16 Teletype Test & Verify Program, which is described in program manual 88A00185A.

- The SPC-16 Processor Test & Verify Program, which is described in program manual 88A00184A.

- The Spc-16 Memory Test Program, which is also described in program manual 88A00184A.

The referenced manuals give all the necessary information needed to load and execute these programs. They also supply an error directory that can be used to isolate the malfunctioning component.

VERIFICATION OF TIMING BOARD OPERATION

Upon encountering an error, the Processor Test & Verify Program puts the processor in a WAIT state. Since the processor remains in this state until the STEP switch is pressed, the technician may check operation of the Timing Board with an oscilloscope during this recycling period.

It should be noted that more than one processor board is listed as possibly faulty for many of the error conditions in the Error Directory. Further, the Timing Board is often listed as one of the possibilities in a multi-board selection. The above mentioned check permits the technician to pass judgement on the integrity of the Timing Board and thereby narrow the list of possibilities.

The timing diagram for the WAIT condition is given in Figure 10-1. Signals shown are defined in Appendix A.

Timing Board pin numbers at which these signals may be scoped are given in Table 10-1.

A "+" or "-" suffix on a signal mnemonic indicates true or complemented signal, respectively.

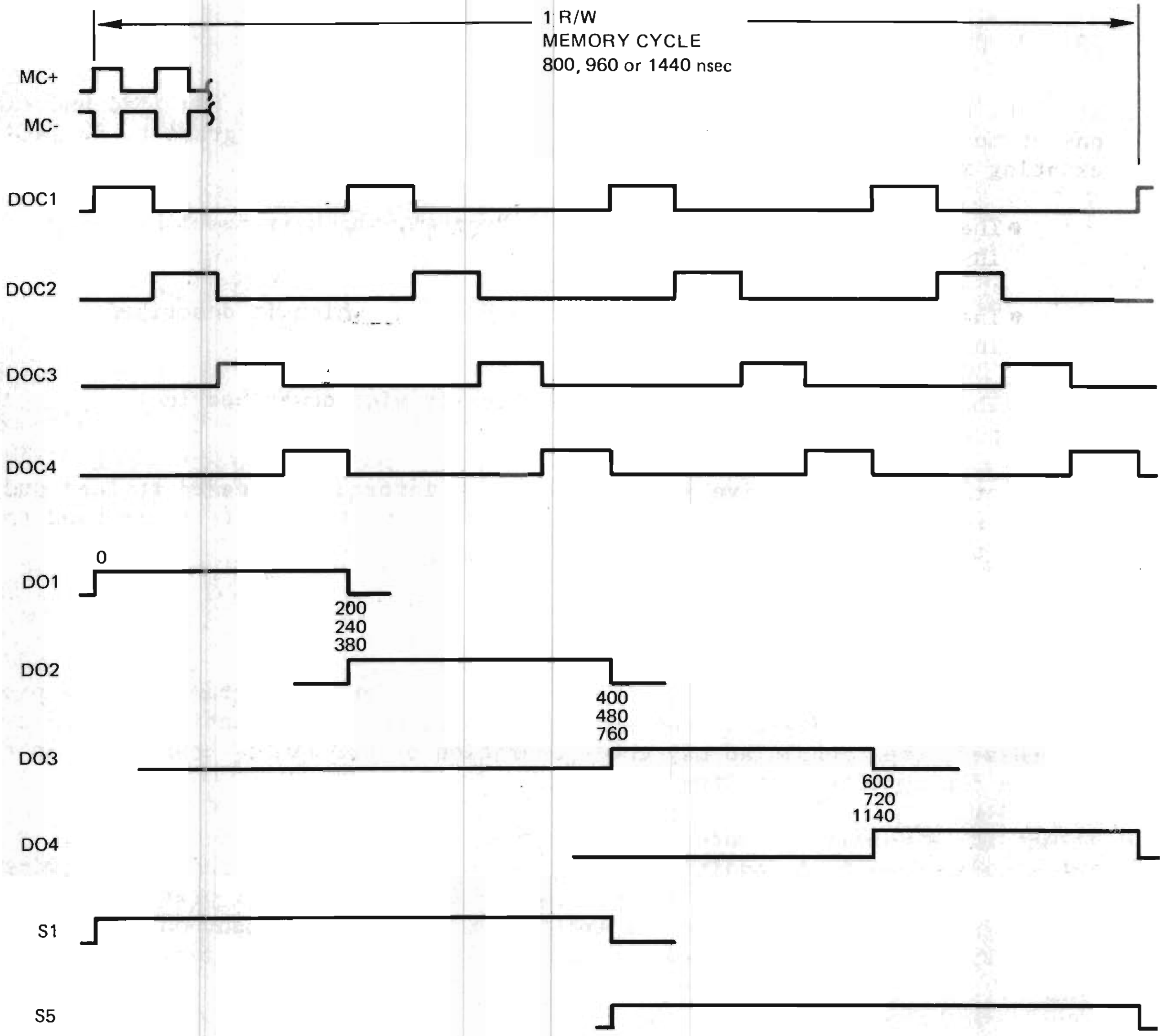


Figure 10-1. Timing Diagram--WAIT

Table 10-1. Timing Board Pin Assignments

Pin No.	Signal	Pin No.	Signal
P1-01	+5V	P2-56	DO3+
P1-02	+5V	P2-59	DO2-
P1-03	Gnd	P2-65	DO3-
P1-04	Gnd	P2-71	S1-
P1-38	S5-	P2-72	MC+
P1-41	DO1-	P2-79	DO4+
P1-54	DOC2-	P2-80	DO4-
P1-63	DOC3-	P2-97	Gnd
P1-71	DOC4+	P2-98	Gnd
P2-52	DOC2+	P2-99	+5V
P2-54	DO1+	P2-100	+5V

In addition to the timing, the technician should take note of the waveform shapes. A severly degraded Master Clock (MC) pulse shape, for example, may not be attaining a "true" state long enough to perform its timing control function properly.

10.2 VERIFICATION OF DATA CHANNEL OPERATION

General Automation high-speed peripheral units that use the Direct Memory Access (DMA) form of input/output communicate with the SPC-16 Central Processing Unit via a Data Channel Module (MHSDC-8). Of course, a user-supplied data channel module may be designed to interface directly with the I/O bus for DMA operations, but this section is written based on a General Automation module.

A single Data Channel Module provides eight high-speed channels and will service up to eight peripheral controllers. It maintains a block count register (SCR), and an address register (CAR) for each active data channel. The block count register specifies the number of data items to be transferred; it is decremented as each item is transferred. The address register contains the address of the data table being transferred; it is incremented at each item transfer.

The Data Channel Module processes cycle-stealing requests and supplies the appropriate data transfer address to the I/O bus. A Data Channel Module communicates with the Central Processing Unit via the I/O Bus, but it communicates with controllers associated with its data channels via the Data Channel Bus.

A Data Channel Module can operate in three modes with respect to each of its eight high-speed data channels. These are termed the initiate mode, the data transfer mode, and the chaining mode.

When a controller wishes to initiate a DMA block transfer of data (usually in response to an "execute I/O" instruction issued by the CPU via programmed input/output), it requests service from the Data Channel Module and enters the initiate mode. The Data Channel Module fetches the starting address and number of items to be transferred from a pair of memory locations dedicated to that controller's data channel. The contents of these locations, which were loaded under program control, are loaded into the Channel Address Register (CAR) and Scan Control Register (SCR), respectively.

Once the initialization phase is complete, the controller may request data transfers at will. It requests service from the Data Channel Module when it is ready to send or receive data. The Data Channel Module increments the CAR and decrements the SCR, for the controller in question, at each data channel request. When the block count is exhausted, the Data Channel Module signals the controller via a "Block Complete" line on the Data Channel Bus.

Blocks of data items of different lengths located at various locations in memory can be transferred by a single program-initiated DMA operation using the chaining mode. The operations performed in this mode are similar to the operations performed in the initiate mode. However, it retrieves each new CAR from the locations specified by the current CAR and the new SCR from the location specified by the new CAR.

A complete description of the MHSDC-8, together with test procedures, is given in GA's Multi High-Speed Data Channel-8 Technical Manual (88A00415A).

10.3 OPERATIONAL CHECK-OUT OF SPC-16 PROCESSOR VIA DISPLAY CONSOLE

Employed properly, the Display Console is an effective tool for use in troubleshooting the SPC-16 Processor.

This section provides a test procedure for checking the integrity of data transmission in the SPC-16 Processor using only the facilities of the Display Console. The procedure is intended as a routine that will allow the technician to isolate malfunctioning console boards using neither Test and Verify programs nor any test equipment.

The subsections of Section 10.3 are:

- Subsection 10.3.1 tests data transfer between the Display Console and the operational registers in the Central Processing Unit.
- Subsection 10.3.2 tests data transfer between the Display Console and core memory.

The troubleshooting techniques contained in this section narrow a malfunction to a single processor board in most cases.

10.3.1 Loading Registers from Console

The following sequence of operations must be implemented before proceeding to test the console:

a. Plug in AC cord.
b. Turn power on. Power switch is located at rear of machine.
c. If console switches are disabled, insert key into Console Key Lock ⑬ and turn clockwise.
d. Set the RUN/IDLE switch ⑪ to IDLE.
e. Select Data Entry switches ⑦ to hexadecimal value X'0000'.
f. Select Status register switch ③ .
g. Press ENTER ⑧ . The General Purpose registers in the background mode have been selected.

By applying the sequence below, the technician loads the I, W, P and General Purpose (background mode) registers with data from the console and then displays it on the Register Display indicators to visually check the validity of the "round trip" (display, to register, and back) transmission. Data transmitted to each register varies from data to the previous register so that the Register Select hardware may be checked at the same time.

The register loading sequence is:

a. Disengage Register Select switches ② through ⑥ .

b. Set Data Entry switches ⑦ to X'FFFF'.

c. Select I-register switch ⑥ .

d. Press ENTER ⑧ . The content of the I register is displayed on the Register Display Indicators ① .

e. Disengage I-register switch ⑥ .

f. Disengage Data Entry switch 0.

g. Repeat steps c. through e. for W register switch ⑤ .

h. Disengage Data Entry switch①

i. Repeat steps c. through e. for P-register switch ④ .

j. Disengage Data Entry switch 2.

k. Repeat steps c. through e. for General Purpose switches ② in the order below. Follow each "register switch-off" with a "rightmost Data Entry switch-off" operation.

Register	Select Switches ②
A'	000
X'	001
Y'	010
Z'	011
B'	100
C'	101
D'	110
E'	111

At this point registers I, W, P and the General Purpose registers should be loaded with values differing by one Register Display Indicator bit as follows:

Register	Hexadecimal Value
I	FFFF
W	FFFE
P	FFFC or 7FFC*
A'	FFF8
X'	FFF0
Y'	FFE0
Z'	FFC0
B'	FF80
C'	FF00
D'	FE00
E'	FC00

*Disregard value of bit 15 for P-register.

l. Visually check whether all registers above do indeed contain the values listed by sequentially displaying the contents of each using the appropriate Register Select switches ② through ⑥ .

If a discrepancy occurs between the expected contents and the indicated (as displayed) contents of a register, the desired data should be re-keyed onto the Data Entry switches ⑦ Steps c, d, e and l should then be performed for the register in question.

m. If the discrepancy still exists after recheck, skip to "Register Transmission Errors from Console Loading" at the end of this subsection. If all values check, continue.

Now the General Purpose registers in the foreground mode (A, X, Y, Z, B, C, D, E) must be loaded and checked. The sequence is:

a. Disengage Register Select switches ② through ⑥ .

b. Set Data Entry switches 8 and 9 (⑦). Switches 10-15 remain set from the preceding sequence.

c. Select Status register switch ③ .

d. Press ENTER ⑧ .

e. Disengage Status register switch ③ .

f. Press ENTER ⑧ . The contents of the Data Entry switches have been loaded into the A' register.

g. Load the remaining General Purpose registers (foreground mode) by the following sequence:

{ Set Data Entry switch } then → { Set Gen. Pur. Select switches } then → { Press ENTER }

	② to
7	001
6	010
5	011
4	100
3	101
2	110
1	111

At this point the General Purpose registers in the foreground mode should be loaded with values differing by one Register Display Indicator bit as follows:

Register	General Purpose Select Switches ②	Hexadecimal Value
A	000	FF00
X	001	FF80
Y	010	FFC0
Z	011	FFE0
B	100	FFF0
C	101	FFF8
D	110	FFFC
E	111	FFFE
Status	-	0100

h. Visually check whether these registers do indeed contain the above values by sequentially displaying the contents of each register using the appropriate Register Select switches ((2) through (6)). If a discrepancy occurs between the expected contents and the actual (as displayed) contents of a register, the desired data should be keyed onto the Data Entry switches (7) . Steps g. and h. should then be performed for the register in question.

i. If the discrepancy still exists after recheck of the register in question, continue. If all registers check with the expected values, go to subsection 10.3.3.

REGISTER TRANSMISSION ERRORS FROM CONSOLE LOADING

An error detected in loading the SPC-16 registers from the Display Console may be attributed to one of the following sources:

- Several registers prove erroneous. This condition indicates one of the following errors:

    1) Register Display Indicator light(s) burned out. This error is easily detected, since the same lights will appear unlit for all registers where they should appear lit.

    2) Augend, Addend, Data Bus or Arithmetic/Logic Unit malfunctioning. Data transmission error involving several registers is probably due to malfunctioning of one of these elements. Since they are all located on the Arithmetic Board, the Arithmetic Board is probably the best board to replace in a multi-error situation.

       If only the General Purpose Registers are giving erroneous displays, the malfunctioning board is even more certainly the Arithmetic Board since these registers are all located on this board.

    3) K (Display) register malfunctioning. If replacement of the Arithmetic Board did not clear up the errors, or if the technician has no Arithmetic Board on hand, a replacement of the console Board (which contains the K register) is the next alternative.

- I-register appears erroneous, other registers check. The I-register is located on the Macro Control Board, but is loaded from the M-register, which is located on the Memory I/O (MIO) Board, so either of these boards may be faulty.

- One or more of the General Purpose Registers appear erroneous. Three possible sources may cause this error.

1) Register Select malfunctioning. If the Register Display Indicators are actually displaying the contents of the wrong General Purpose register, the Register Select network is malfunctioning. This network is located on the Arithmetic Board. However, if that wrong register is actually the corresponding register in the unselected mode, then mode select (foreground or background) is malfunctioning. For example, if the B register (foreground mode) was selected for display, but the Indicators are displaying the B register (background mode), then mode select is malfunctioning. This condition signals malfunction on the Macro Control Board, which contains mode select.

2) Register malfunctioning. If the displayed register contents contains some incorrect bits, the register itself is malfunctioning. The Arithmetic Board should be replaced.

3) Augend Bus, Arithmetic/Logic Unit, or Data Bus malfunctioning. Indicated errors in several General Purpose registers signal malfunctioning buses or ALU. The Arithmetic Board should be replaced.

- P or W registers appear erroneous. Bus or register malfunctions related to these registers signal replacement of Arithmetic Board.

- Status register appears erroneous. Buses related to the Status register are located on the Arithmetic Board; the Status register itself is located on the Macro Control Board, However, if the Status register is the only erroneous register, the problem is more likely solved by replacement of the Macro Control Board.

10.3.2 Loading Memory from Console

Since the integrity of console communication with the SPC-16 registers has been verified in the preceding section, console communication with memory must now be investigated. A "worst case" is used for checking. In software nomenclature, this case features memory addressing that is "indirect and base-relative with indexing". That is, the address of operand to be displayed is the sum of (a) the contents of the memory location specified by the sum of the D-register and the displacement in the instructions and (b) the contents of the index register. In equation form, the Effective Address (address of the operand) is:

$$EA = (D + Disp) + Index$$

The following parameters are employed in this example:

o D-register contents = X'0EFD' }
o Displacement = X'02' } D + Disp = X'0EFF'
o Index (X-register contents) = X'0011'
o Contents of location '0EFF' - X'00F0'

So Effective Address = X'00F0' + X'0011' = X'0101' is the address of the memory location containing the operand. Let the operand itself be X'AAAA', which is an alternating pattern of ones and zeroes. If the instruction works correctly, then, the Display Console Indicators will contain X'AAAA'.

LOADING MEMORY VIA DIRECT ADDRESSING

As a first step in preparing to execute the indirect, base-relative, indexed test instruction, memory location X'0EFF' must be loaded with a value of '00F0'. In performing this task, the integrity of the "direct addressing" mode is tested.

Load location X'0EFF' as follows:

a. Select P-register switch ④ .

b. Set Data Entry switches ⑦ to X'0EFE'. Since the P-register will be incremented before the data is loaded into memory, this step indicates that the memory location selected is X'0EFF'.

c. Press ENTER ⑧ . Address enters P-register.

d. Disengage P switch ④ .

e. Select I-register switch ⑥ .

f. Set Data Entry switches ⑦ to X'5000', which is the pattern for instruction "STA 0,0" (Store A-register in memory at location specified by P-register).

g. Press ENTER ⑧ . Instruction enters I-register.

h. Disengage I-register switch ⑥ .

i. Press SAVE I switch ⑨ .

j. Set Data Entry switches ⑦ to X'00F0'.

k. Press ENTER ⑧ . Data enters A-register.

l. Press STEP switch ⑩ . The data (X'00F0') will be stored at the specified location (X'0EFF') and the P-register will be incremented.

m. Disengage SAVE I switch ⑨ .

A check must be made to verify the above operation:

n. Repeat steps a. through d. This procedure re-addresses the appropriate memory location.

o. Select I-register switch ⑥ .

p. Set Data Entry switches ⑦ to X'4000', which is the pattern for LDA 0,0 (Load contents of memory location specified by P-register into the A-register).

q. Press ENTER ⑧ . Instruction enters I-register.

r. Disengage I-register switch ⑥ .

s. Press SAVE I switch ⑨ .

t. Press STEP switch ⑩ . The contents of the selected memory location have been loaded into the A-register and are displayed on the Register Display Indicators ① .

u. Disengage SAVE I switch ⑨ .

If proper data transmission has occurred between the registers and memory the Register Display Indicators will display X'00F0'.

MEMORY TRANSMISSION ERRORS, DIRECT ADDRESSING

If the Register Display Indicators do not display X'00F0', however, erroneous console communication with memory is indicated. Therefore, malfunction exists in either the Memory and Input/Output (MIO), Arithmetic, or Timing Board.

Perhaps the easiest first check in tracking down the area of malfunction is to view the contents of the P-register, which is done by selecting P-register switch ④ .

A display X'0EFE' indicates that the anticipated P-register incrementation did not accur. Since the integrity of P-register loading has been previously verified, the error most likely resides in either the timing or the addition capability of the Arithmetic/Logic Unit. If possible, the technician should first replace the Arithmetic Board, which contains the Arithmetic/Logic Unit. If not possible, the timing should be checked as follows:

a. Disengage all Data Entry switches ⑦ .

b. Select I-register switch ⑥ .

c. Press ENTER ⑧ . A WAIT instruction is placed into the I (Instruction) register.

d. Press STEP switch ⑩ . The WAIT instruction is executed.

The procedure to test the integrity of the Timing Board in a WAIT state is given in subsection 10.1.2 ("Verification of Timing Board Operation" paragraph).

A correct P-register value (i.e., X'0EFF') indicates that either the Memory and Input/Output Board or the Timing Board is faulty. The Timing Board may be tested using the procedure given above. Correct timing eliminates the Timing Board from suspicion of malfunction and isolates the MIO Board as faulty.

LOADING EFFECTIVE ADDRESS CONTENTS INTO MEMORY FROM CONSOLE

With location X'0EFF' properly loaded, the second step in preparing for execution of the indirect, base-relative, indexed test instruction is to load the Effective Address X'0101' with a value of X'AAAA'. This operation is easily accomplished by re-doing steps a. through m. above, but using X'0100' and X'AAAA' as the Data Entry Switch values in steps b. and j., respectively.

LOADING TEST INSTRUCTION INTO MEMORY FROM CONSOLE

There are two ways to execute the test instruction. Certainly, the simplest way is to load the instruction into the I-register and execute. However, since the purpose of this manual is to troubleshoot the system, a much more interesting approach involves loading the instruction into memory and then commanding the system to fetch and execute it. The second approach is implemented here since it tests a "fetch and execute" procedure rather than merely an "execute" procedure.

The instruction to be executed is:

```
LDR A, *2,X,1
```

This instruction commands the system to "Load" the A-register with the contents of the memory location addressed by the sum of the contents of memory location 'D+2' and the contents of the X-register.

The instruction will be loaded into memory location '0021' by re-doing steps a. through m. above, but using X'0020' and X'CD02' as the Data Entry switch values in steps b. and j., respectively.

The configuration of core memory following the preceding steps is shown in Figure 10-2.

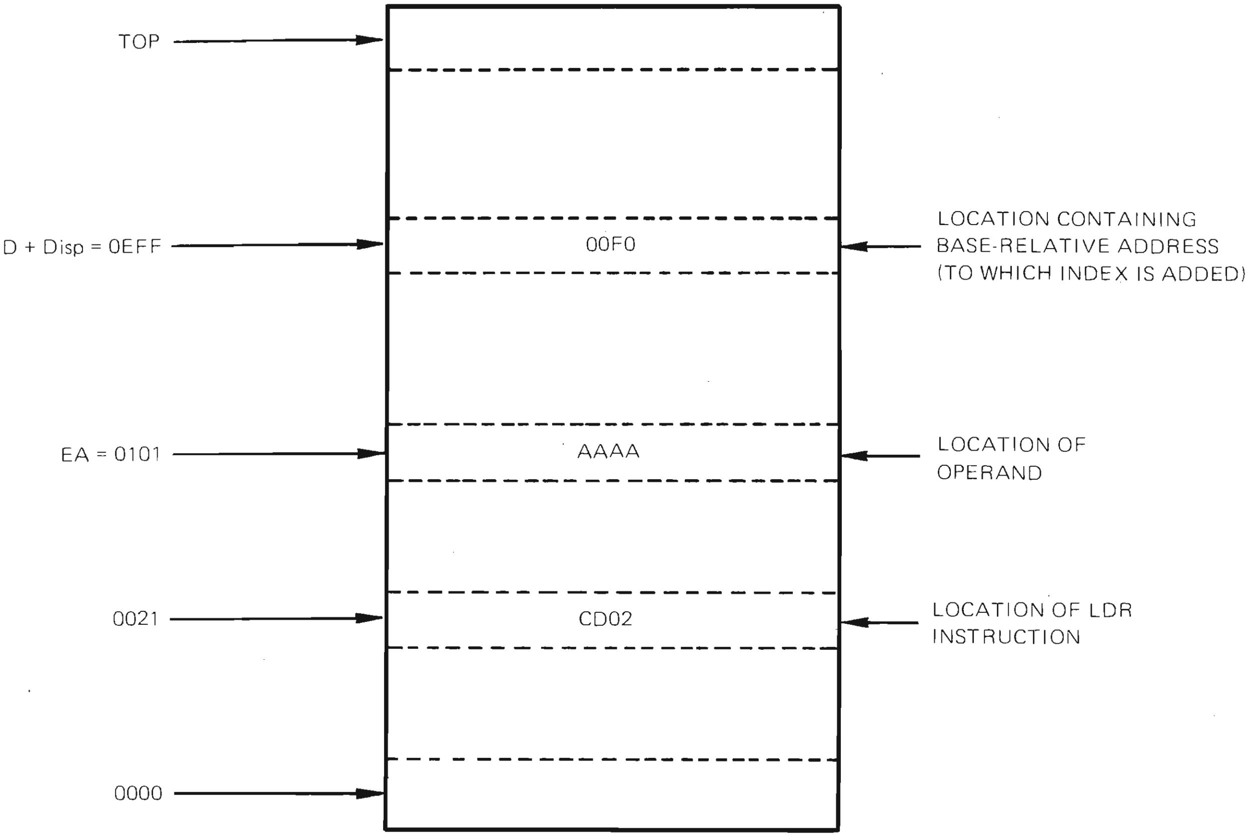


Figure 10-2. Memory Allocation for Base-Relative Indirect LDR Instruction With Indexing

LOADING D AND X-REGISTERS FROM CONSOLE

The next step in preparing for the test is to load the D and X-registers with the required X'OEFD' and X'0011' values, respectively.

To load these registers:

a. Set Data Entry switches ⑦ to X'0011'.

b. Set General Purpose Register Select switches ② to '001'.

c. Press ENTER ⑧ . X-register is loaded.

d. Set Data Entry switches ⑦ to X'OEFD'.

e. Set General Purpose Register Select switches ② to '110'.

f. Press ENTER ⑧ . D-register is loaded.

g. Disengage General Purpose Register Select switches ② .

FETCH AND EXECUTE INSTRUCTION

Now that the memory and the appropriate registers have been loaded, all that remains is to address the memory location containing the test instruction ('0021') and execute. The sequence is as follows:

a. Select P-register switch ④.

b. Set Data Entry switches ⑦ to X'0021'.

c. Press ENTER ⑧ .

d. Disengage P-register switch ⑨ .

e. Press STEP ⑩.

The Register Display Indicators should now read X'AAAA' if the instruction was properly executed.

MEMORY TRANSMISSION ERRORS FROM CONSOLE LOADING

A display of any pattern other than X'AAAA' on the Register Display Indicators signals faulty execution of the instruction.

The logical first step in tracking down the malfunctioning area is to verify that memory is loaded in accordance with Figure 10-2. Since the contents of location X'0EFF' has been previously verified, the contents of locations X'0021' (which contains the instruction) and X'0101' (which contains the operand) must be verified.

The verification sequence is:

a. Select P-register switch ④ .

b. Set Data Entry switches ⑦ to X'0020'.

c. Press ENTER ⑧ .

d. Disengage P switch ④ .

e. Select I-register switch ⑥ .

f. Set Data Entry switches ⑦ to X'4000', which is the pattern for "LDA 0,0'".

g. Press ENTER ⑧ .

h. Select SAVE I switch ⑨ .

i. Disengage I-register switch ⑥ .

j. Press STEP switch ⑩ .

At this point, if the Register Display Indicators display X'CD02', continue to step k. If not, the LDR instruction was not properly loaded into memory and the steps in "Loading Test Instruction Into Memory From Console" should be redone. Recurrence of the previous error signals malfunction as described in "Memory Transmission Errors, Direct Addressing".

k. Select P-register switch ④ .

l. Set Data Entry switches ⑦ to X'0100'.

m. Press ENTER ⑧ .

n. Disengage P-register switch ④ .

o. Press STEP switch ⑩ .

p. Disengage SAVE I switch ⑨ .

A display of X'AAAA' on the Register Display Indicators indicates proper loading of the operand into memory and informs the technician that the existing error is based in the execution of the test instruction. An incorrect display, however, necessitates reloading the operand per the "Loading Effective Address Contents Into Memory From Console" section. Recurrence of the previous error signals malfunction as described in "Memory Transmission Error, Direct Addressing".

Verification of the contents of the preceding memory locations eliminate the Memory and Input/Output Board as the source of error. The Arithmetic Board has been excluded as a source of error by previous tests in this section. Further, the previously-executed "LDA" and "STA" instructions justify exclusion of the Timing Board as the faulty element. The conclusion one must reach, then, is that the test instruction did not execute properly because it was either improperly decoded or the indirect addressing option was faulty. In either case, the malfunction resides on the Macro Control board, so that board should be replaced.

10.4 CHECK OF POWER SUPPLY VOLTAGES

The power supply voltages can be checked and adjusted using the procedure given in subsection 9.2.

10.5 PROCESSOR BOARD SUBSTITUTION

Possibly the fastest way to remedy a processor malfunction is to substitute all available spares on a one-to-one basis with boards in the processor. The procedure to remove and replace the boards is given in Section 4.2.

The procedure for isolating a fault by board substitution is as follows:

1. Remove power from the computer.

2. Exchange CPU boards with available spare boards.

3. Re-apply power and check to see whether malfunction is still present. If malfunction remains, go to Section 10.3; if not, continue.

4. If the replacement of several spare boards has eliminated the malfunction, exchange the spares (now in the console) with their original-board counterparts one by one, verifying proper operation after each change.

   When the malfunction reappears, it is indicated that the last board exchanged is faulty and requires permanent replacement.

5. Continue to replace boards until all faulty boards have been replaced.

10.6 REMOVAL AND REPLACEMENT OF CPU SUBASSEMBLIES

The following subassemblies in the CPU are subject to removal and replacement in the field:

1. CPU, memory and controller boards.
2. Console board.
3. Master Interconnect board(s).
4. Fans

Although the removal and replacement procedures provided below apply specifically to the major subassemblies of the mainframe, they are also generally applicable to subassemblies in the memory extension chassis and external I/O enclosure.

CAUTION

Shut power off before removing or installing any part near a printed circuit board. When removing or installing a part, use care to avoid damaging the components mounted on the board or the circuit traces etched on the board.

Appendix C gives a parts lists for the major subassemblies.

10.6.1 CPU, Memory and Controller Boards

All printed circuit boards mounted in a card rack (CPU, memory and peripheral controller boards) are removed and installed through the front of the cabinet. To do this, first open the console panel 90°. Do not strain the two ribbon cables connected to the console board. Then remove the desired board by pulling on both extractor tabs.

CAUTION

Do not use pliers to remove the boards.

Two points should be kept in mind when installing a board in its card slot:

1. It is important to be certain both rear corners of the board get started in their tracks. Otherwise, the board could be pushed against an adjacent board, possibly damaging components or etched circuits.

2. If the board meets excessive resistance when it reaches the edge connector at the back of the card rack, either the connector is not aligned with the card slot or contacts within the connector are loose or damaged.

10.6.2 Console Board

To remove the console board, first open the console panel and disconnect the two ribbon cable paddle boards from the console board connectors (see Figure 10-3). Then remove the four screws that hold the console board in place.

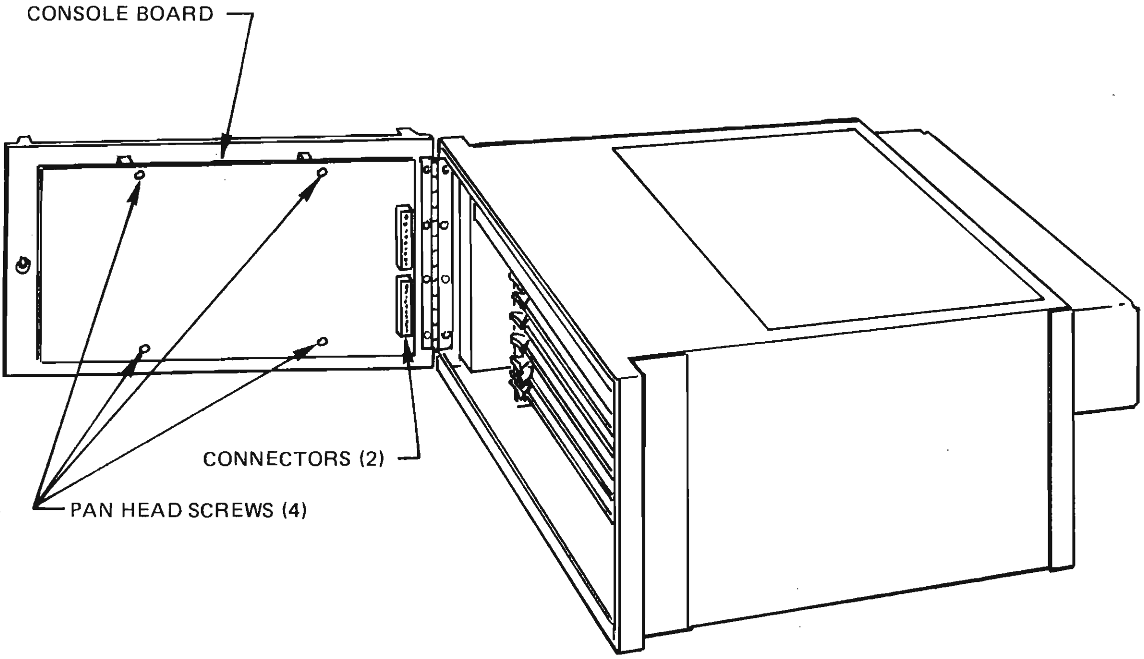


Figure 10-3. Removing Console Board

10.6.3 Master Interconnect Board(s)

The MIB contains the connectors for CPU, memory and controller boards on one side of the board and connectors for outside cabling on its reverse side. The 40/60/80 models have one MIB, 45/65/85 models have two.

SPC-16 40/60/80 SERIES MIB

Removal of the MIB may be required to check the condition of the board or connectors on the board. The procedure for removal and replacement of the MIB is provided below; Figure 10-4 illustrates this procedure.

1. Disconnect the AC-line cord from the power source.
2. Remove all peripheral cable paddle boards from their respective edge connectors.
3. Open the console panel and remove all CPU, memory and controller boards from the mainframe.

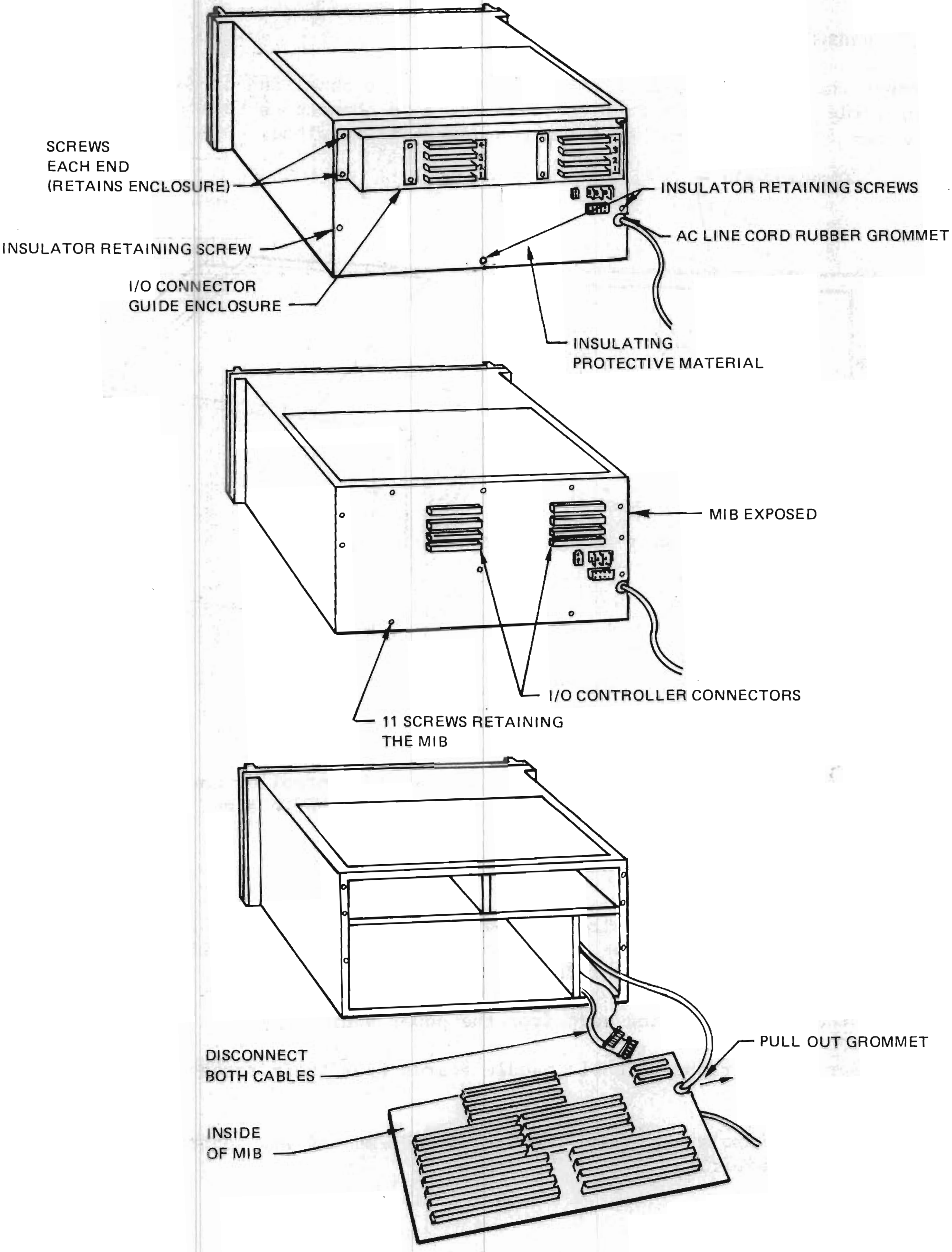


Figure 10-4. Removing MIB (40/60/80)

4. Remove the screws that hold the peripheral cable connector enclosure and insulator panel in place.

5. Remove the remaining screws that fasten the MIB to the mainframe chassis and carefully pull the board out away from the chassis just enough to expose the two interior ribbon cable paddle boards.

6. Disconnect these two paddle boards from their MIB connectors.

7. Remove the AC-line cord and rubber grommet from the slot in the MIB.

8. The MIB can now be examined and, if necessary, repaired or replaced.

To replace the MIB and reassemble the rear panel of the mainframe, carry out the following procedure:

1. Insert the AC-line cord and rubber grommet in the slot provided on the edge of the MIB.

2. Slide the MIB along the power cord toward the mainframe until the board is close enough for the two internal ribbon cable paddle boards to be inserted in the appropriate connectors on the MIB.

3. Insert both paddle boards.

4. Position the MIB against the mainframe chassis so that the MIB retaining screws can be started in their holes.

NOTE

Before fully tightening any of the MIB retaining screws, be certain the MIB internal connectors are aligned with their respective board slots. This alignment can be examined from the front of the chassis with the console panel open.

5. When the interior connectors are in line with the board slots, tighten the MIB retaining screws.

6. Fasten the insulator panel and peripheral cable connector in place.

7. Return the CPU, memory and controller boards to their respective slots in the mainframe chassis.

8. Insert the peripheral cable paddle boards into the appropriate I/O connectors.

9. Plug the AC-line cord into the power source.

SPC-16 45/65/85 SERIES MIB's

Because some memory modules in the 45/65/85 series CPU's occupy board slots not covered by the basic MIB used in 40/60/80 series CPU's, an additional MIB is required to accommodate these other modules. The two MIB's in a 45/65/85 series CPU are interconnected by short ribbon cables.

Unlike a 40/60/80 series MIB, these MIB's do not have interior I/O controller connectors. However, one of the MIB's does include an external I/O bus cable connector and enclosure. This connector is the receptacle for the external I/O bus cable paddle board.

These MIB's may be removed and replaced if necessary. The procedure for removal and replacement is provided below; Figure 10-5 illustrates this procedure.

1. Disconnect the AC-line cord from the power source.

2. Remove the I/O bus cable paddle board from its connector and the three MIB interconnect ribbon cables from their connectors.

3. Open the console panel and remove all CPU and memory boards from the mainframe.

4. Remove the screws that hold the channel bar support (3) and protective material (4) in place.

5. Remove the remaining screws that fasten the two MIB's to the mainframe chassis.

6. Carefully pull the basic MIB out away from the chassis just enough to expose the two interior ribbon cable paddle boards.

7. Disconnect these two paddle boards from their MIB connectors.

8. Remove the AC-line cord and rubber grommet from the slot in the MIB.

9. Both MIB's can now be examined and, if necessary, repaired or replaced.

To replace the MIB and reassemble the rear panel of the mainframe, carry out the following procedure:

1. Insert the AC-line cord and rubber grommet in the slot provided on the edge of the basic MIB.

2. Slide the basic MIB along the power cord toward the mainframe until the board is close enough for the two internal ribbon cable paddle boards to be inserted in the appropriate connectors on the basic MIB.

3. Insert both paddle boards.

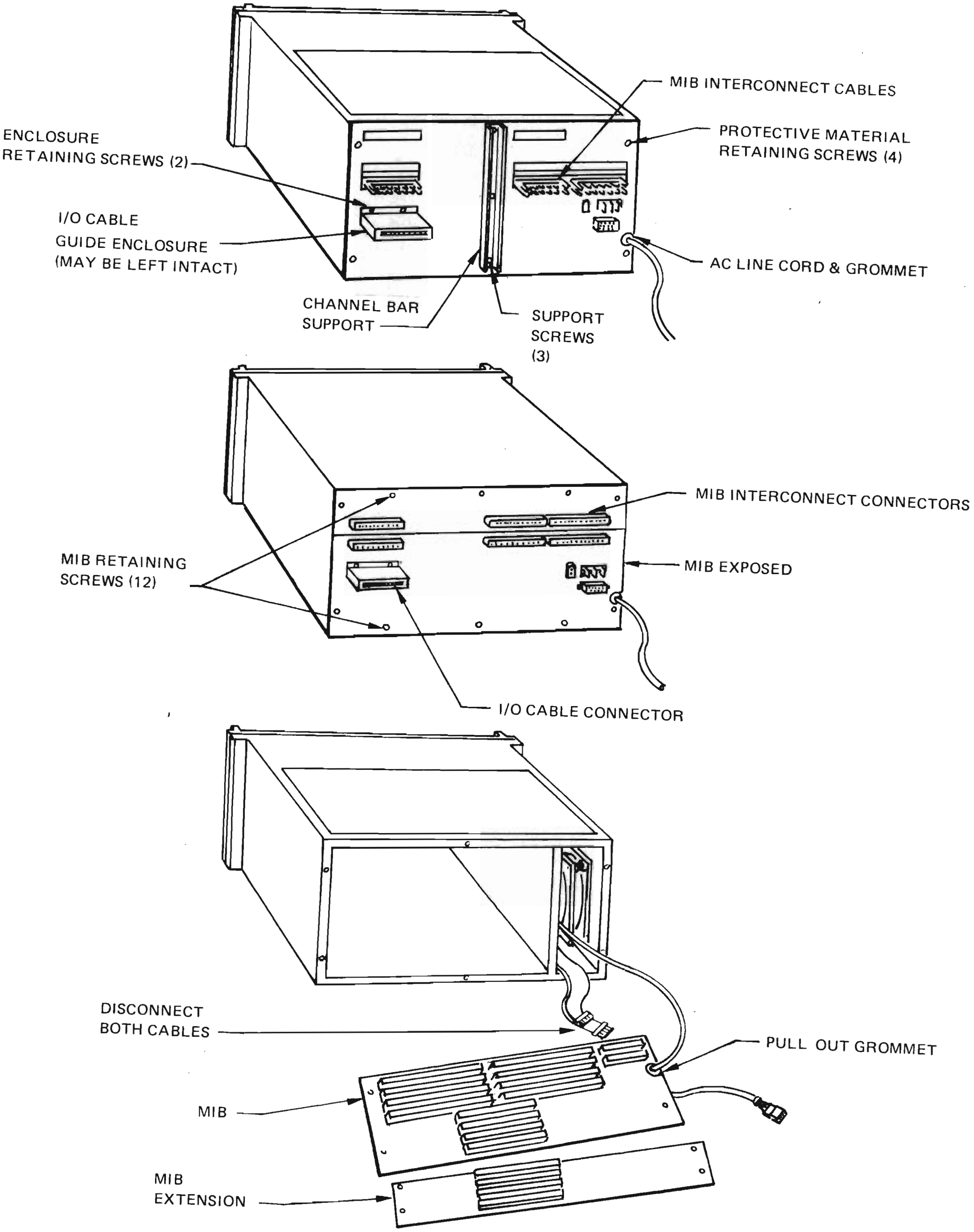

MIB INTERCONNECT CABLES
ENCLOSURE RETAINING SCREWS (2)
PROTECTIVE MATERIAL RETAINING SCREWS (4)
I/O CABLE GUIDE ENCLOSURE (MAY BE LEFT INTACT)
AC LINE CORD & GROMMET
CHANNEL BAR SUPPORT
SUPPORT SCREWS (3)
MIB INTERCONNECT CONNECTORS
MIB RETAINING SCREWS (12)
MIB EXPOSED
I/O CABLE CONNECTOR
DISCONNECT BOTH CABLES
PULL OUT GROMMET
MIB
MIB EXTENSION

Figure 10-5. Removing MIB's (45/65/85)

4. Position each MIB (one at a time) against the mainframe chassis so that the MIB retaining screws can be started in their holes.

NOTE

Before fully tightening any of the MIB retaining screws, be certain the MIB internal connectors are aligned with their respective board slots. This alignment can be examined from the front of the chassis with the console panel open.

5. When the interior connectors are in line with the board slots, tighten the MIB retaining screws.

6. Fasten the insulator panel, channel bar support and exterior I/O bus cable connector in place.

7. Return the CPU and memory boards to their respective slots in the mainframe chassis.

8. Insert the external I/O bus cable paddle board into its connector.

9. Plug the AC-line cord into the power source.

APPENDIX A
SIGNAL MNEMONIC DEFINITION

36	IRO-4 = $36)_8$ = X'1E'
37	DISP for MRX (IRO-4) = $37)_8$ = X'1F'
37 MRX	37 and MRX
ACKT	Acknowledge Time for Interrupt or DMA
AD00/AD15	Addend Bus Bits 00 through 15
ARMC	Interrupt Arm Clock
AU00/AU15	Augend Bus Bits 00 through 15
AUTOR	Automatic Restart
BIT00/BIT15	Bit Select Logic Bits 00 through 15
BITL	Bit Left Select
BITR	Bit Right Select
CACK	Carry Acknowledge Flip-Flop (DMA only)
CACK-D	Carry Acknowledge (DMA Cable)
CINT	Console Interrupt Request
CLDS	Cold Start
CLK1	Serial Controller Clock 1
CLK2	Serial Controller Clock 2
CLRM1	Clear Memory Data Register, one
CLRME	Clear Memory Data Register Enable
CLRML	Clear Memory Data Register Left
CLRMR	Clear Memory Data Register Right
CLRS	Clear Shift Counter
CLRSE	Clear Shift Counter Enable
CMR	Compare Memory with Register Instruction Decode

CNSI	Console Interrupt Switch
CNTL	Control Instruction Decode
CONSE	Console Set Enable
CONSL	Console Select
CONST	Console Set
COUT	Carry Out (from Adder)
CTRLE	Control Instruction Enable
CTRLP	Control Instruction Pulse
D2MSE	DO2 Memory Start Enable
D3	General Purpose I/O Clock
D4MSE	DO4 Memory Start Enable
D1O3	DO1 or DO3
D1O3D	DO1 or DO3 Data Port Enable
DACK	DMA Acknowledge
DACK-D	DMA Acknowledge (DMA Cable)
DB00/DB15	Data Bus Bits 00 through 15
DC9	Serial I/O Bit Clock 9
DC10	Serial I/O Bit Clock 10
DCPA	Serial I/O Set Xmit Mode Device Control Pulse
DCPB	Serial I/O Set Receive Mode Device Control Pulse
DCPC	Serial I/O Set Echo Mode Device Control Pulse
DCPD	Serial I/O Set Break Mode Device Control Pulse
DDTP	DMA Data Transfer Pulse
DDTP-D	DMA Data Transfer Pulse (DMA Cable)
DMA00-D/DMA14-D	DMA Address Lines (DMA Cable)
DMAF0-D/DMAF3-D	Arithmetic Logic Control Lines (DMA Cable)

DMC-I	Direct Memory Cycle (I/O Cable)
DMCR	DMA Cycle Request
DO1	DO Time 1
DO1D	DO1, Data Port Enable
DO2	DO Time 2
DO2D	DO2, Data Port Enable
DO2E	DO2 Enable
DO2E3	DO2 Enable At DO3
DO3	DO Time 3
DO3D	DO3, Data Port Enable
DO4	DO Time 4
DO4D	DO4 Data Port Enable
DO4E	DO4 Enable
DOC1	DO Clock 1
DOC12	DO Clock 1 or 2
DOC1D	DO Clock 1 Data Port Enable
DOC2	DO Clock 2
DOC3	DO Clock 3
DOC4	DO Clock 4
DOPM-D	DMA Operate on Memory (DMA Cable)
DPOLL-D	DMA Priority Clock (DO2)
DREQ-D	DMA Cycle Steal Request (DMA Cable)
DTI	Data Transfer In
DTO	Data Transfer Out
DTP	Data Transfer Pulse
DTP-I	Data Transfer Pulse (I/O Cable)

DTREG	Data Transfer Register
ECHO	Serial I/O Echo Mode Select
EIR1/3-I	External Interrupt 1 through 3 (I/O Cable)
ENDSR	End Shift
ENTER	Enter Console Data Switches into Selected Register
ENTR	Enter Switch
EXBY	Exchange Bytes
EXINH	Extended Memory Inhibit
FADDE	Function Add Enable (XP0)
FAP	Function Address Pulse
FCIN	Function Carry Into Adder (C_n)
FINH	Function Inhibit to Adder (Mode Control)
FRGND	Foreground Mode Select
FX0/3	Function Select Lines to the ALU (S0/S3)
GEN	GEN Instruction Decode
GEN53	GEN Instruction and S5 and D03
GENI	GEN Instruction (GEN·S7·UCE)
GEN0	GEN Zero Instruction Decode from IR8-11
GLDBY	GEN or Load Byte Instruction Decode
GOLD	GEN or any Load (LDA, LDR, LDBY) Instruction
I123	Interrupt Request Levels 1 or 2 or 3
I7I6	IR07 and IR06 (IR06 and IR07 both = 1)
I7N6	IR07 and IR06 (IR07 = 1, IR06 = 0)
IACK	Interrupt Acknowledge
IDLE	Idle Mode (Only DOC's and MC Running)
IDLS	INCM, DECM, LARS, SARS Instruction Decode

IDM	INCM, DECM Instruction Decode
IDM52	IDM at S5 and D02
IHLD	Interrupt Hold
IMSE	Inhibit Memory Start Enable
INDEX	Index Decode
INHT	Memory Inhibit Current Enable Time
INHTD	Memory Inhibit Current Enable Time Data Port Enable
INHV	Memory Inhibit Voltage
IO00/IO15	I/O Data Bits 00 through 15
IOA76	I/O Address 76)$_8$ Decode (from IR0-5)
IOA77	I/O Address 77)$_8$ Decode (from IR0-5)
IOTD	I/O Test Flip-Flop Data Port Enable
IOTT	I/O Test True
IPRS	Interrupt Priority Return Status
IPSO	Interrupt Priority Status Out
IR00/IR15	Instruction Register Bits 00 through 15
IRDMA	Select IR0-3 or DMAF0-3 to ALU Function Bus
IREQ-I	Interrupt Request (I/O Cable)
IRQ1/IRQ7	Interrupt Level Request 01 through 07
ISE	Interrupt System Enable
ISW	I Register Switch
ITCR	Interrupt Cycle Request
JSR	Jump to Subroutine Decode
JUMPS	JMP or JSR Instruction Decode
KBIT5	SKIP or BIT Instructions and S5

KLOCK	System's Console Keylock
KR00/KR15	Console Display Register Bits 00 through 15
LDBY	Load Byte Instruction Decode
LDBYL	Load Byte Left
LDBYR	Load Byte Right
LEFT	Left Byte Select
LEVEL	Interrupt Level Active Indicator
LINK	Link Indicator
LINKD	Link Data Port Enable
LKADS	RLK or ADDS Instruction Decode
LR00/LR11	L (Memory Address) Register Bits 00 through 11
LSARS	Load or Store All Registers and Status Instruction Decode
MAC-D	DMA Multiplex Address Control (DMA Cable)
MACF	DMA Multiplex Address Control Flip-Flop
MC	Master Clock
MCOSC	Master Clock Oscillator
MET	Skip Instruction Condition Met
MGRD	Memory Guard
MPY	Multiply Instruction Decode
MPYS9	Multiply Instruction at S9
MR	Memory Reference Instruction Class Decode
MR00/MR15	Memory Data Register Bits 00 through 15
MRMRX	MR or MRX Instruction Class Decode
MRX	Memory Reference with Indexing Instruction Class Decode

MRX5	MRX Instructions at S5
MRX12	MRX Instructions at N812
MRXX	MRX Instruction with Indexing Selected
MSE	Memory Start Enable
MSRT	Serial I/O Master Reset
MSTO/MST7	Memory Start Select 0 through 7
MSTC	Memory Start Clock
MTIM	Memory Current Time
MTIMI	Memory Time Inhibit
N7I6	Not IR07 and IR06 (IR07 = 0, IR06 = 1)
N812	Not S8 and S1 and DO2
N813	Not S8 and S1 and DO3
N842	Not S8 and S4 and DO2
N843	Not S8 and S4 and DO3
N851	Not S8 and S5 and DO1
N852	Not S8 and S5 and DO2
N853	Not S8 and S5 and DO3
N872	Not S8 and S7 and DO2
N89	Not S8 and S9
N891	Not S8 and S9 and DO1
N893	Not S8 and S9 and DO3
N1109	Not IR11 and IR09 (IR11 = 0, IR09 = 1)
N8DO2	Not S8 and DO2
N8DO3	Not S8 and DO3
N8D23	Not S8 and DO2 and DOC3
N8N12	Not S8 and Not S1 and DO2

NACK4	Not Acknowledge Time and DO4
NBYX	Not Byte Indexed
NODO	No DO Time (DO1·DO2·DO3)
NSRUN	Not SAVI and RUN
OFLO	Overflow Indicator
OFLOC	Overflow Indicator Clock
OFLOD	Overflow Indicator Data Port Enable
OFMET	Overflow Condition Met
OMA	Operations Monitor Alarm
OMAS	Operations Monitor Alarm Set
PDOC3	Powered DOC3
PDOC4	Powered DOC4
PFD	Power Fail Detect
PFIR	Power Fail Interrupt Request
PFRST	Power Fail or Restart Interrupt Request
PGENI	Powered GENI
PIR05, 06, 09, 14	Powered I Register Bits
PISE	Powered ISE
PLUS	Plus Indicator
PLUSC	Plus Indicator Clock
PMTIM	Powered MTIM
PN842	Powered N842
PN852	Powered N852
POLL	Powered Pulse (Skewed D3)
PQR00, 15	Powered Q Register Bits
PR00/PR15	P Register Bits 00 through 15

PROL	Powered ROL Instruction Decode
PS1	Powered S1
PS8	Powered S8
PS9	Powered S9
PSW	P Register Switch
PXPO	Powered XPO
PXPOF	Powered XPOF
QR00/QR15	Q Register Bits 00 through 15
RAB0, 1, 2	Register Address Bus 0, 1, 2
RASW0, 1, 2	Register Address Switch 0, 1, 2
RAUTO	Reset Automatic Restart Flip-Flop
RBIT	Reset Bit Instruction Decode
RCD53	Register Change Destination Instruction Class at S5 and D03
RCDE	Register Change Destination Instruction Class Enable
RCINT	Reset Console Interrupt Request
RCO53	Register Change Operate Instruction Class at S5 and D03
RCOE	Register Change Operate Instruction Class Enable
RCS53	Register Change Source Instruction Class at S5 and D03
RCSE	Register Change Source Instruction Class Enable
RCV	Serial I/O Receive Mode
RCVP	Serial I/O Receive Pulse
RD	Serial I/O Receive Data
RDCL	Read/Clear Memory Function Enable

RDCL-D	Read/Clear Memory (DMA Cable)
READ	Memory Read Time
READR	Memory Read Time Reset
REDY	Serial I/O Ready
RO53	Register Operate Instruction Class at S5 and D03
ROE3, 7	Read Only Memory PF/AR Vector Address Offset Enable
ROL	Register Operate Literal Instruction Class
ROLE	Register Operate Literal Instruction Class Enable
ROM	Read Only Memory
RQIN8	DMA Request In at S8
RQIN-D	Request Input (DMA Cable)
RQSE	R and Q Register Set Enable
RR00/RR15	R Register Bits 00 through 15
RRRO	RO, RCO or ROL Instruction Group Decode
RS	Auto-Restart
RSC1, 2, 4, 8	Register and Status Counter Bits 1, 2, 4, 8
RSRT	Serial I/O Read Start
RTC	Real Time Clock
RTCCK	Real Time Clock Clock
RTCR	Real Time Clock Reset
RUN	Run Mode
RUND	Run Mode Data Port Enable
RUNSW	Run Switch
S1	Sequence State 1 (Instruction Fetch)
S123	S1 or S2 or S3

S1233	S1 or S2 or S3 and DO3
S1234	S1 or S2 or S3 or S4
S1D4	S1 Data Port Enable
S1E1, 2	S1 Enable 1 or 2
S1OS2	S1 or S2
S1RDE	S1 Register Destination Enable
S1SE	S1 Set Enable
S2	Sequence State 2
S237	S2 or S3 or S7
S2D	S2 Data Port Enable
S3	Sequence State 3 (Indirect Addressing)
S3D	S3 Data Port Enable
S4	Sequence State 4
S4D	S4 Data Port Enable
S5	Sequence State 5
S567	S5 and S6 and S7
S5D	S5 Data Port Enable
S5OS6	S5 or S6
S5RDE	S5 Register Destination Enable
S6	Sequence State 6 (Micro Control)
S6D	S6 Data Port Enable
S7	Sequence State 7 (Interrupts)
S7OS8	S7 or S8
S8	Sequence State 8 (Data Channels)
S9	Sequence State 9 (Multiply-Divide)
S9D	S9 Data Port Enable

SA00/SA15	Sense Amplifier Bits 00 through 15
SACKA	Sequence State Advance Clock A
SACKB	Sequence State Advance Clock B
SARS	Store All Register and Status Instruction Decode
SAVEI	Save I Register
SAVI	Save I Register Switch
SBITL	Set Bit Left
SBITR	Set Bit Right
SC00/SC03	Shift Counter Bits 00 through 03
SCC	Shift Count Complete
SCCK	Shift Counter Clock
SCLK	Shift Clock Serial Controller
SCSW	Select Console Switches to I/O Bus
SDC	Serial Data Common
SDECM	Select-1 for Decrement Memory Instruction
SDI	Serial Data In
SDO	Serial Data Out
SDMA	Select Direct Memory Access
SERCC	Serial Controller Clock Control
SETIC	Set I Register Clock
SETIE	Set I Register Enable
SETIF	Set I Register Flip-Flop
SETIS	Set I Register
SETME	Set Memory Data Register Enable
SETP	Set P Register
SETQ	Set Q Register

SETR	Set R Register
SETW	Set W Register
SFEC-I	System Safe, Electronic (I/O Cable)
SFIR	Select to ALU Function Bus from I Register
SHFQ	Shift Q Register
SHFW	Shift W Register
SHNQ	Shift in Q Register (Bit 15 Input during any Shift Instruction)
SI	Select I Register
SIN	Select Input
SIO	Select I/O Bus
SIP	Select and Increment P Register
SKIP	Skip Instruction Decode
SLE	Select E Register
SLPE4	Select P Register Enable at DO4
SLRS	Select Source Register
SMMS6	Select Memory Data Register Most Significant 6 Bits
SM04	Select Memory Data Register Bits 0 through 4
SM9	Select Memory Data Register Bit 9
SM58	Select Memory Data Register Bits 5 through 8
SMX8	Select Memory Data Register and Extend Bit 8
SMX9	Select Memory Data Register and Extend Bit 9
SORBL	Set or Reset Bit Left
SORBR	Set or Reset Bit Right
SP	Select P Register

SQLL	Select Q Register Left to Left
SQRR	Select Q Register Right to Right
SR	Select R Register
SR1	Select R Register Enable 1
SR2	Select R Register Enable 2
SRAS	Select Register Address Switches
SRCCE	Shift Right Circular or Circular Link Instruction Enable
SRCL	Shift Right Circular Link
SRI	Shift Right Instruction
SRI52	SRI at S5 and DO2
SRI53	SRI at S5 and DO3
SRLAE	Shift Right Logical or Arithmetic Instruction Enable
SRLL	Select R Register Left to Left
SRLR	Select R Register Left to Right
SRRL	Select R Register Right to Left
SRRR	Select R Register Right to Right
SRSXE	Select Register Source for XIO Enable
SRTSW	System Reset Switch
SS	Select Status
STA	Store A Instruction Decode
STAR	Store A or R Instruction Decode
STBY	Store Byte Instruction Decode
STBYL	Store Byte Left
STBYR	Store Byte Right
STEP	Step Mode

STK	Set Console Display Register
STI	Set I Register
STL	Set L Register
STML	Set Memory Data Register Left
STMR	Set Memory Data Register Right
STPE	Set P Register Enable
STPE1	Set P Register Enable 1
STPSW	Step Switch
STQE	Set Q Register Enable
STQE2	Set Q Register Enable at DOC 2
STQE4	Set Q Register Enable at DOC 4
STRE	Set R Register Enable
STRE2	Set R Register Enable At DOC 2
STRE4	Set R Register Enable at DOC 4
STSW	Status Switch
STWE	Set W Register Enable
SW	Select W Register
SYNC	Sync Pulse
SYRT	System Reset
SZPOL	Select Indicators and Foreground
TEST	Test Instruction Decode
TEST-I	I/O Test (I/O Cable)
TTYI	Teletype Interrupt
TURN	Turn Function Select
UCE	Micro Control Enable
UEOI	Micro End of Instruction

UMMS6	Micro Select M Register Most Significant 6 Bits
URAB0, 1, 2	Micro Register Address Bus Bits 0, 1, 2
USCCK	Micro Shift Counter Clock
USDMA	Micro Select DMA
USTP	Micro Set P Register
USTW	Micro Set W Register
USHFQ	Micro Select Shift Right Q Register
USI	Micro Select I Register
USIN	Micro Select In
USIO	Micro Select I/O Input
USM04	Micro Select M Register Bit 0 through 4
USM9	Micro Select M Register Bit 9
USM58	Micro Select M Register Bit 5 through 8
USMX8	Micro Select M Register Extend Bit 8
USMX9	Micro Select M Register Extend Bit 9
USP	Micro Select P Register
USQLL	Micro Select Q Register Left to Left
USQRR	Micro Select Q Register Right to Right
USRLL	Micro Select R Register Left to Left
USRLR	Micro Select R Register Left to Right
USRRL	Micro Select R Register Right to Left
USRRR	Micro Select R Register Right to Right
USS	Micro Select Status
USTQE	Micro Set Q Register Enable
USTRE	Micro Set R Register Enable

USTRT	Micro Control Start
USW	Micro Select W Register
USZ	Micro Select Indicators and Foreground
UWRIT	Micro Scratch Pad Write Enable
VU	Output Voltage Signal from K Register DAC (AUT•VU)
WAIT	Wait Instruction Decode
WAITI	Wait Increment P Register Select
WR00/WR15	W Register Bits 00 through 15
WRIT	Memory Write Time
WRITE	Scratch Pad Write Clock
WRTE	Scratch Pad Write Enable
WTOID	Wait or Idle
WWPCE	Write, Set W and Set P Clock Enable
WWPCK	Write, Set W and Set P Clock
XD>>	Transmit Data (Serial I/O)
XEC	Execute Instruction Decode
XIO	XIO Instruction Decode
XPO	Extended Processor Option Instructions (MPY, DIV) Decode
XPOF	Extended Processor Option Flip-Flop (on from S5D03 through S9D02)
XPOLC	XPO Link Clock
XPOLD	XPO Link Data
XPOQS	XPO Set Q Register
XS1A, 1B	Scratch Pad Register X Drive Select 1A, 1B
XS2A, 2B	Scratch Pad Register X Drive Select 2A, 2B

XS3A, 3B	Scratch Pad Register X Drive Select 3A, 3B
XS4A, 4B	Scratch Pad Register X Drive Select 4A, 4B
YS1A, B	Scratch Pad Register Y Drive Select 1A, 1B
YS2A, B	Scratch Pad Register Y Drive Select 2A, 2B
YS3A, B	Scratch Pad Register Y Drive Select 3A, 3B
YS4A, B	Scratch Pad Register Y Drive Select 4A, 4B
ZACK	Zero Acknowledge (DMA Only)
ZCC	ZACK and CACK Flip-Flop Clocks
ZDET	Zero Detect
ZERO	Zero Indicator
ZEROC	Zero Flip-Flop Clock
ZEROD	Zero Data Port Enable
ZPLCE	Zero, Plus and Link Clock Enable
ZPOLS	Set Indicators and Foreground
ZRBY	Zero Right Byte Instruction Decode

APPENDIX B
COMPONENT INTERCHANGEABILITY

It is recommended that replacement parts be direct replacements or equivalents. Appendix C lists part numbers for all major subassemblies of each computer model. The system integration package (SIP) supplied with each system will provide part numbers and ordering information for system components other than the computer proper.

B.1 INTERCHANGEABILITY

Users with more than one SPC-16 model may be able to interchange components, especially CPU boards. Therefore, depending on the models a customer owns, it may be unnecessary to have a complete set of spare boards for each model.

Tables B-1 through B-6 show the interchangeability of components for each model in the SPC-16 family. An X indicates complete interchangeability and a 0 indicates that the components are not interchangeable.

Table B-1. SPC-16/40 Component Compatibility

Assembly Number	Part Type	SPC-16/40 Series			SPC-16/45 Series			SPC-16/30 Series		
		40	60	80	45	65	85	30	50	70
31D01333A01	SPC-16/40 Arithmetic Board	X	X	0	X	X	0	X	X	0
31D01331A01	SPC-16/40 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A11	SPC-16/40 Timing Board	X	0	0	X	0	0	X	0	0
31D01640A01	SPC-16/40 Processor MIB Board	X	X	X	0	0	0	0	0	0
31D01640A11	SPC-16/40 MIB (I/O Expansion)	X	X	X	-	-	-	-	-	-
31D01628A61	SPC-16/40 MIO Board	X	X	0	X	X	0	-	-	-
31D01635A11	SPC-16/40 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A61	SPC-16/40 8K Memory Board	X	X	X	X	X	X	0	0	0
31D01783A01	SPC-16/40 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00025A01	SPC-16/40 Processor Chassis	X	X	X	0	0	0	0	0	0
51D00021A01	SPC-16/40 Power Supply	X	X	X	X	X	X	0	0	0
31D01446A01	SPC-16/40 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/40 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/40 Crystal (Basic Clock Frequency)	X	0	0	X	0	0	X	0	0

Table B-2. SPC-16/60 Component Compatibility

Assembly Number	Part Type	SPC-16/40 Series			SPC16/45 Series			SPC-16/30 Series		
		40	60	80	45	65	85	30	50	70
31D01333A01	SPC-16/60 Arithmetic Board	X	X	0	X	X	0	X	X	0
31D01331A01	SPC-16/60 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A01	SPC-16/60 Timing Board	X	X	0	X	X	0	X	X	0
31D01640A01	SPC-16/60 Processor MIB Board	X	X	X	0	0	0	0	0	0
31D01640A11	SPC-16/60 MIB (I/O Expansion)	X	X	X	-	-	-	-	-	-
31D01628A61	SPC-16/60 MIO Board	X	X	0	X	X	0	-	-	-
31D01635A11	SPC-16/60 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A51	SPC-16/60 8K Memory Board	0	X	X	0	X	X	0	0	0
31D01783A01	SPC-16/60 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00025A01	SPC-16/60 Processor Chassis	X	X	X	0	0	0	0	0	0
51D00021A01	SPC-16/60 Power Supply	X	X	X	X	X	X	0	0	0
31D01446A01	SPC-16/60 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/60 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/60 Crystal (Basic Clock Frequency)	0	X	0	0	X	0	0	X	0

Table B-3. SPC-16/80 Component Compatibility

Assembly		SPC-16/40 Series			SPC-16/45 Series			SPC-16/30 Series		
Number	Part Type	40	60	80	45	65	85	30	50	70
31D01333A11	SPC-16/80 Arithmetic Board	X	X	X	X	X	X	X	X	X
31D01331A01	SPC-16/80 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A21	SPC-16/80 Timing Board	X	X	X	X	X	X	X	X	X
31D01640A01	SPC-16/80 Processor MIB Board	X	X	X	0	0	0	0	0	0
31D01640A11	SPC-16/80 MIB (I/O Expansion)	X	X	X	-	-	-	-	-	-
31D01628A71	SPC-16/80 MIO Board	X	X	X	X	X	X	-	-	-
31D01635A11	SPC-16/80 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A31	SPC-16/80 8K Memory Board	0	0	X	0	0	X	0	0	0
31D01783A01	SPC-16/80 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00025A01	SPC-16/80 Processor Chassis	X	X	X	0	0	0	0	0	0
51D00021A01	SPC-16/80 Power Supply	X	X	X	X	X	X	0	0	0
31D01446A01	SPC-16/80 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/80 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/80 Crystal (Basic Clock Frequency)	0	0	X	0	0	X	0	0	X

Table B-4. SPC-16/45 Component Compatibility

Assembly Number	Part Type	SPC-16/40 Series			SPC-16/45 Series			SPC-16/30 Series		
		40	60	80	45	65	85	30	50	70
31D01333A01	SPC-16/45 Arithmetic Board	X	X	0	X	X	0	X	X	0
31D01331A01	SPC-16/45 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A11	SPC-16/45 Timing Board	X	0	0	X	0	0	X	0	0
31D01648A01	SPC-16/45 Processor MIB Board	0	0	0	X	X	X	0	0	0
31D01628A61	SPC-16/45 MIO Board	X	X	0	X	X	0	0	0	0
31D01635A11	SPC-16/45 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A61	SPC-16/45 8K Memory Board	X	X	X	X	X	X	0	0	0
31D01783A01	SPC-16/45 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00026A01	SPC-16/45 Processor Chassis	0	0	0	X	X	X	0	0	0
51D00021A01	SPC-16/45 Power Supply	X	X	X	X	X	X	0	0	0
11D00036A01	SPC-16/45 Memory Expansion Chassis	-	-	-	X	X	X	0	0	0
31D01650A01	SPC-16/45 Memory Expansion MIB	-	-	-	X	X	X	0	0	0
31D01652A01	SPC-16/45 Memory Expansion Logic Display Board	-	-	-	X	X	X	-	-	-
31D01446A01	SPC-16/45 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/45 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/45 Crystal (Basic Clock Frequency)	X	0	0	X	0	0	X	0	0

Table B-5. SPC-16/65 Component Compatibility

Assembly		SPC-16/40 Series			SPC-16/45 Series			SPC-16/30 Series		
Number	Part Type	40	60	80	45	65	85	30	50	70
31D01333A01	SPC-16/65 Arithmetic Board	X	X	0	X	X	0	X	X	0
31D01331A01	SPC-16/65 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A01	SPC-16/65 Timing Board	X	X	0	X	X	0	X	X	0
31D01648A01	SPC-16/65 Processor MIB Board	0	0	0	X	X	X	0	0	0
31D01628A61	SPC-16/65 MIO Board	X	X	0	X	X	0	-	-	-
31D01635A11	SPC-16/65 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A51	SPC-16/65 8K Memory Board	0	X	X	0	X	X	0	0	0
31D01783A01	SPC-16/65 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00026A01	SPC-16/65 Processor Chassis	0	0	0	X	X	X	0	0	0
51D00021A01	SPC-16/65 Power Supply	X	X	X	X	X	X	0	0	0
11D00036A01	SPC-16/65 Memory Expansion Chassis	-	-	-	X	X	X	0	0	0
31D01650A01	SPC-16/65 Memory Expansion MIB	-	-	-	X	X	X	0	0	0
31D01652A01	SPC-16/65 Memory Expansion Logic Display Board	X	X	X	X	X	X	-	-	-
31D01446A01	SPC-16/65 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/65 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/65 Crystal (Basic Clock Frequency)	0	X	0	0	X	0	0	X	0

Table B-6. SPC-16/85 Component Compatibility

Assembly		SPC-16/40 Series			SPC-16/45 Series			SPC-16/30 Series		
Number	Part Type	40	60	80	45	65	85	30	50	70
31D01333A11	SPC-16/85 Arithmetic Board	X	X	X	X	X	X	X	X	X
31D01331A01	SPC-16/85 Macro Board	X	X	X	X	X	X	X	X	X
31D01811A21	SPC-16/85 Timing Board	X	X	X	X	X	X	X	X	X
31D01648A01	SPC-16/85 Processor MIB Board	0	0	0	X	X	X	0	0	0
31D01628A71	SPC-16/85 MIO Board	X	X	X	X	X	X	-	-	-
31D01635A11	SPC-16/85 4K Memory Board	X	X	X	X	X	X	X	X	X
31D01594A31	SPC-16/85 8K Memory Board	0	0	X	0	0	X	0	0	0
31D01783A01	SPC-16/85 16K Memory Board	X	X	X	X	X	X	0	0	0
11C00026A01	SPC-16/85 Processor Chassis	0	0	0	X	X	X	0	0	0
51D00021A01	SPC-16/85 Power Supply	X	X	X	X	X	X	0	0	0
11D00036A01	SPC-16/85 Memory Expansion Chassis	-	-	-	X	X	X	0	0	0
31D01650A01	SPC-16/85 Memory Expansion MIB	-	-	-	X	X	X	0	0	0
31D01652A01	SPC-16/85 Memory Expansion Logic Display Board	-	-	-	X	X	X	-	-	-
31D01446A01	SPC-16/85 Console Board	X	X	X	X	X	X	0	0	0
09D00014A01	SPC-16/85 Console Panel	X	X	X	X	X	X	0	0	0
	SPC-16/85 Crystal (Basic Clock Frequency)	0	0	X	0	0	X	0	0	X

APPENDIX C
PARTS LISTS, COMPUTER MAINFRAME ASSEMBLY

This appendix includes parts lists for the SPC-16 40/60/80 and 45/65/85 computer mainframes. Tables C-1 through C-3 are parts lists for SPC-16 Models 40, 60 and 80, respectively; Tables C-4 through C-6 are parts lists for SPC-16 Models 45, 65 and 85, respectively. Table C-7 reflects MIB-Memory Module compatibility; Table C-7 does not reflect any interchangeability between different speed CPU's.

Table C-1. Computer Mainframe Assembly
SPC-16/40

Part Number	Description	Size, Features	Model Numbers, Notes
31D01635A11	Memory Board	4K x 16	Model 1640-0100 [1] [3]
31D01594A61	Memory Board	8K x 16	Model 1640-0200 [4]
31D01783A01	Memory Board	16K x 16	
31D01333A01	Arithmetic Board		
31D01811A11	Timing Control Board	w/Fail Safe Group	Model 1640-0008
31D01331A31	Macro Control Board	w/FG/BG & MULT/DIV	Model 1640-0004
31D01331A11	Macro Control Board	w/FG/BG	Model 1640-0002
31D01331A21	Macro Control Board	w/MULT/DIV	
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV	
31D01628A61	MIO Board Assembly	w/TTY Controller	[2]
31D01450A01	MIB Assembly	4K Memory Boards only	Model 1640-0001
31D01450A11	MIB Assembly	I/O Expansion	72 Pin
31D01640A01	MIB Assembly		56 Pin [3]
31D01640A11	MIB Assembly		
31D01446A01	Console Board Assy		
09D00014A01	Console Assembly		
11C00025A01	Chassis, Mainframe		
01D00081A	Computer, Mainframe Assembly		

Notes:

[1] A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01450A01, 11).

[2] A 31D01459A21 MIO Board may be used in place of 31D01628A61 in a 4K MIB (31D01450A01, 11).

[3] See Table C-7 for MIB-Memory compatibility.

[4] A 31D01594A31 or A51 Memory Board may be used in place of 31D01594A61.

Table C-2. Computer Mainframe Assembly
SPC-16/60

Part Number	Description	Size, Features	Model Numbers, Notes
31D01635A11	Memory Board	4K x 16	Model 1660-0100 △1 △3
31D01594A51	Memory Board	8K x 16	Model 1660-0200 △4
31D01783A01	Memory Board	16K x 16	
31D01333A01	Arithmetic Board		
31D01811A01	Timing Control Board	w/Fail Safe Group	Model 1660-0008
31D01331A31	Macro Control Board	w/FG/BG & MULT/DIV	Model 1660-0004
31D01331A11	Macro Control Board	w/FG/BG	
31D01331A21	Macro Control Board	w/MULT/DIV	
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV	
31D01628A61	MIO Board Assembly	w/TTY Controller	Model 1660-0001 △2
31D01450A11	MIB Assembly	I/O Expansion	72 Pin } △3
31D01450A01	MIB Assembly	4K Memory Boards only	56 Pin
31D01640A01	MIB Assembly		
31D01640A11	MIB Assembly		
31D01446A01	Console Board Assembly		
09D00014A01	Console Assembly		
11C00025A01	Chassis, Mainframe		
01D00082A	Computer Mainframe Assy		

Notes: △1 A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01450A01, 11).

△2 A 31D01459A21 MIO Board may be used in place of 31D01628A61 in a 4K MIB (31D01450A01, 11).

△3 See Table C-7 for MIB-Memory compatibility.

△4 A 31D01594A31 Memory Board may be used in place of 31D01594A51.

Table C-3. Computer Mainframe Assembly SPC-16/80

Part Number	Description	Size, Features	Model Numbers, Notes	
31D01635A11	Memory Board	4K x 16	Model 1680-0100	△1 △3
31D01594A31	Memory Board	8K x 16	Model 1680-0200	
31D01783A01	Memory Board	16K x 16		
31D01333A11	Arithmetic Board			
31D01811A21	Timing Control Board	w/Fail Safe Group	Model 1680-0080	
31D01331A31	Macro Control Board	w/FG/BG & MULT/DIV	Model 1680-0004	
31D01331A11	Macro Control Board	w/FG/BG	Model 1680-0002	
31D01331A21	Macro Control Board	w/MULT/DIV		
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV		
31D01628A71	MIO Board Assembly	w/TTY Controller		△2
31D01450A01	MIB Assembly	4K Memory Boards only	Model 1680-0001	△3
31D01450A11	MIB Assembly	I/O Expansion	72 Pin	
31D01640A01	MIB Assembly		56 Pin	
31D01640A11	MIB Assembly			
31D01446A01	Console Board Assy			
09D00014A01	Console Assembly			
11C00025A01	Chassis, Mainframe			
01D00083A	Computer Mainframe Assy			

Notes:

△1 A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01450A01, 11).

△2 A 31D01459A31 MIO Board may be used in place of 31D01628A71 in a 4K MIB (31D01450A01, 11).

△3 See Table C-7 for MIB-Memory compatibility.

Table C-4. Computer Mainframe Assembly
SPC-16/45

Part Number	Description	Size, Features	Model Numbers, Notes	
31D01635A11	Memory Board	4K x 16	Model 1645-0100	△1 △3
31D01594A61	Memory Board	8K x 16	Model 1645-0200	△4
31D01783A01	Memory Board	16K x 16		
31D01628A41	MIO Board Assembly	w/o TTY Controller		△2
31D01628A61	MIO Board Assembly	w/TTY Controller	Model 1645-0001	△2
31D01452A01	MIB Assembly	4K Memory Boards only		△3
31D01648A01	MIB Assembly			△3
13C00189A01	Panel, Dummy Exp. MIB			
31D01650A01	MIB Assembly	Expanded Memory		△3
31D01446A01	Console Board Assy			
31D01333A01	Arithmetic Board			
31D01331A31	Macro Control Board	w/FG/BG & MULT/DIV	Model 1645-0004	
31D01331A11	Macro Control Board	w/FG/BG	Model 1645-0002	
31D01331A21	Macro Control Board	w/MULT/DIV		
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV		
31D01811A41	Timing Control Board	w/o Fail Safe Group		
31D01811A11	Timing Control Board	w/Fail Safe Group	Model 1645-0008	
11C00026A01	Chassis, Mainframe			
09D00014A01	Console Assembly		Model 1645-0500	
01D00084A	Computer Mainframe Assy			

Notes:

△1 A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01452A01).

△2 A 31D01459A01 or A21 MIO Board may be used in place of 31D01628A41 or A61, respectively, in a 4K MIB (31D01452A01).

△3 See Table C-7 for MIB-Memory compatibility.

△4 A 31D01594A31 or A51 Memory Board may be used in place of 31D01594A61.

Table C-5. Computer Mainframe Assembly
SPC-16/65

Part Number	Description	Size, Features	Model Numbers, Notes
31D01635A11	Memory Board	4K x 16	Model 1665-0100 △1 △3
31D01594A51	Memory Board	8K x 16	Model 1665-0200 △4
31D01783A01	Memory Board	16K x 16	
31D01628A41	MIO Board Assembly	w/o TTY Controller	△2
31D01628A61	MIO Board Assembly	w/TTY Controller	Model 1665-0001 △2
31D01452A01	MIB Assembly	4K Mem. Bds. only	△3
31D01648A01	MIB Assembly		△3
13C00189A01	Panel, Dummy Exp. MIB		
31D01650A01	MIB Assembly	Expanded Memory	△3
31D01446A01	Console Board Assy		
31D01333A01	Arithmetic Board		
31D01331A31	Macro Control Board	w/FG/BG & MULT/DIV	Model 1665-0004
31D01331A11	Macro Control Board	w/FG/BG	Model 1665-0002
31D01331A21	Macro Control Board	w/MULT/DIV	
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV	
31D01811A31	Timing Control Board	w/o Fail Safe Group	
31D01811A01	Timing Control Board	w/Fail Safe Group	Model 1665-0008
11C00026A01	Chassis, Mainframe		
09D00014A01	Console Assembly		Model 1665-0500
01D00085A	Computer Mainframe Assy		

Notes:

△1 A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01452A01).

△2 A 31D01459A01 or A21 MIO Board may be used in place of 31D01628A41 or A61, respectively, in a 4K MIB (31D01452A01).

△3 See Table C-7 for MIB-Memory compatibility.

△4 A 31D01594A31 Memory Board may be used in place of 31D01594A51.

Table C-6. Computer Mainframe Assembly
SPC-16/85

Part Number	Description	Size, Features	Model Numbers, Notes	
31D01635A11	Memory Board	4K x 16	Model 1685-0100	[1] [3]
31D01594A31	Memory Board	8K x 16	Model 1685-0200	[2]
31D01783A01	Memory Board	16K x 16		
31D01628A51	MIO Board Assembly	w/o TTY Controller		[2]
31D01628A71	MIO Board Assembly	w/TTY Controller	Model 1685-0001	[2]
31D01452A01	MIB Assembly	4K Memory Boards only		[3]
31D01648A01	MIB Assembly			[3]
13C00189A91	Panel, Dummy Exp. MIB			
31D01650A01	MIB Assembly	Expanded Memory		[3]
31D01446A01	Console Board Assy			
31D01333A11	Arithmetic Board			
31D01331A31	Macro Control Board	w/FG/BG & MULT/ DIV	Model 1685-0004	
31D01331A11	Macro Control Board	w/FG/BG	Model 1685-0002	
31D01331A21	Macro Control Board	w/MULT/DIV		
31D01331A01	Macro Control Board	w/o FG/BG or MULT/DIV		
31D01811A51	Timing Control Board	w/o Fail Safe Group		
31D01811A21	Timing Control Board	w/Fail Safe Group	Model 1685-0008	
11C00026A01	Chassis, Mainframe			
09D00014A01	Console Assembly		Model 1685-0500	
01D00086A	Computer Mainframe Assy			

Notes:

[1] A 31D01470A01 Memory Board may be used in place of 31D01635A11 in a 4K MIB (31D01452A01).

[2] A 31D01450A11 or A31 MIO Board may be used in place of 31D01628A51 or A71, respectively, in a 4K MIB (31D01452A01).

[3] See Table C-7 for MIB-Memory compatibility.

APPENDIX D
SPC-16 CONSOLE

The controls and indicators, which are located on the front panel of the computer, allow manual data entry, examination of register and memory contents and initiation of automatic operation.

A key-operated console lock switch on the front panel provides a safety feature to prevent inadvertent operation of console controls when the system is on-line and operating.

The console is illustrated in Figure D-1. The console panel contains indicators (lighted/unlighted) and switches (up = 0 or not set, down = 1, or set). The ENTER ⑧ STEP ⑩ , SYSTEMS RESET ⑫ , and CONSOLE INTERRUPT ⑭ switches return to the up (off) position after being manually depressed. All other switches may be manually set either up or down.

The number codes (① through ㉒) are referred to in parts of Section 5.

① REGISTER DISPLAY indicators (0 through 15) indicate contents of the System Console Display register (K-register). The K-register is loaded under program control by the DSPL instruction in the RUN mode; in the IDLE mode, it is loaded with the contents of the selected register (see Register Select switches below).

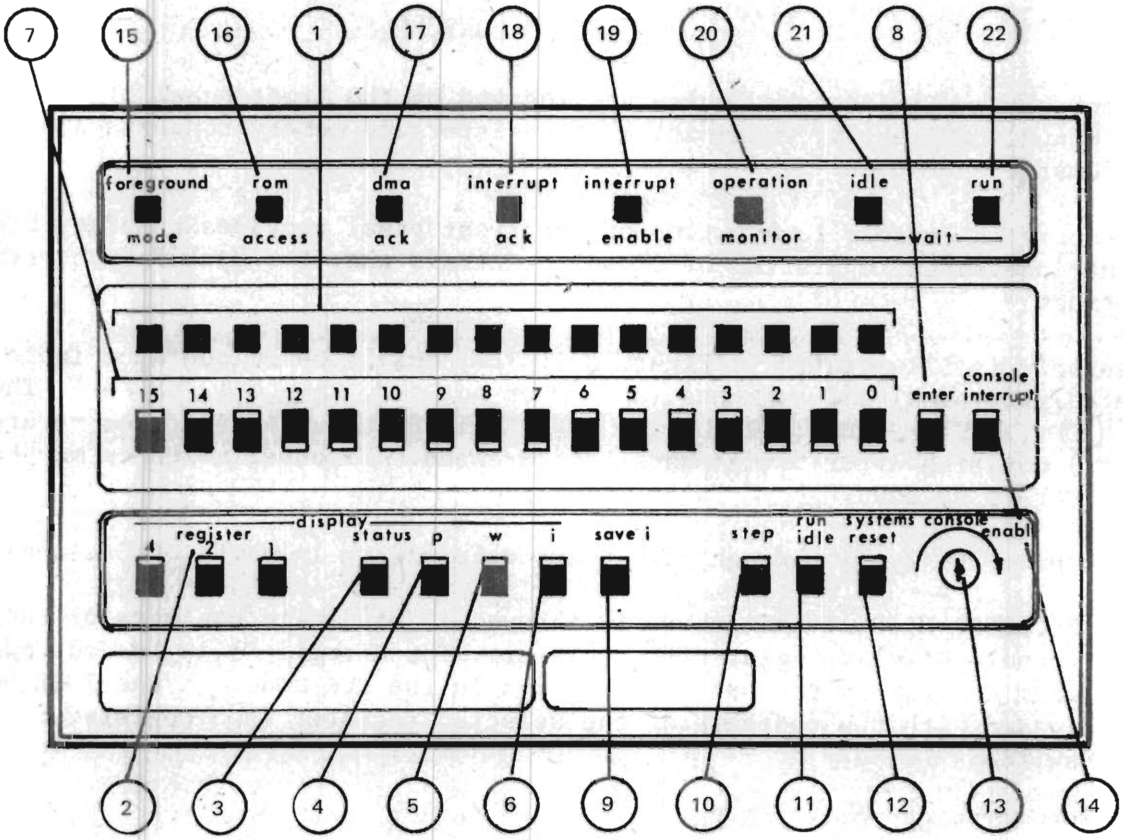
REGISTER SELECT SWITCHES ② through ⑥

These switches are used to select a register for display or data entry. They have a left-to-right priority; for example, if switch 6 is on (down), the other Register Select switches are disabled and the I-register is selected.

② GENERAL PURPOSE REGISTER select switches are used to select one of the eight operational registers to receive data subsequently keyed into the Data Entry switches ⑦ . The other Register Select switches ③ , ④ , ⑤ and ⑥ must be in the up (off) position in order to select a register with switches ② . After keying in a register code in the IDLE mode, the register contents are displayed on the console indicator lights ① .

The register codes are:

Register	Select Switches
A	000
X	001
Y	010
Z	011
B	100
C	101
D	110
E	111



① 16 REGISTER DISPLAY indicators
② 3 GENERAL PURPOSE REGISTER select switches
③ STATUS Register select switch
④ P-Register select switch
⑤ W-Register select switch
⑥ I-Register select switch
⑦ 16 DATA Entry switches
⑧ ENTER switch
⑨ SAVE I switch
⑩ STEP switch
⑪ RUN/IDLE switch
⑫ SYSTEMS RESET switch
⑬ CONSOLE KEY LOCK

⑭ CONSOLE INTERRUPT switch
⑮ FOREGROUND MODE indicator

⑯ READ ONLY MEMORY ACCESS indicator
⑰ DIRECT MEMORY ACCESS/AUTOMATIC DATA CHANNEL ACKNOWLEDGE indicator
⑱ INTERRUPT ACKNOWLEDGE indicator
⑲ INTERRUPT ENABLE indicator
⑳ OPERATIONS MONITOR ALARM indicator
㉑ IDLE indicator
㉒ RUN indicator

Figure D-1. The SPC-16 Computer Console

③ STATUS REGISTER select switch places the contents of the indicators and the shift counter (S-register) into the Console Display register. The Register Display indicators ① will show the status as follows:

Shift Counter	→ $DSPL_{0-3}$	
Link	→ $DSPL_4$	
Overflow	→ $DSPL_5$	
Plus	→ $DSPL_6$	
Zero	→ $DSPL_7$	
Foreground	→ $DSPL_8$	(same as ⑮)
0	→ $DSPL_{9-15}$	

The indicators may be changed by the Data Entry switches ⑦ .

④ "P" REGISTER select switch displays the contents of the Program Counter register on the Register Display indicators ① , bits 0-14. The interrupt status (same as ⑲) is displayed in bit 15. When manually stopping an executing program, the current instruction is completed before execution stops. Since the program counter is incremented by the current instruction, the P-register will contain the address of the next instruction to be executed.

The program counter may be changed by setting the Data Entry switches ⑦ . The interrupt status enable (bit 15) is not affected when the program counter is changed.

⑤ "W" REGISTER select switch displays the contents of the W-register. The W-register contains the effective operand address for the last instruction executed. The Working register may be changed by the Data Entry switches.

⑥ "I" REGISTER select switch displays the contents of the Instruction register.

When manually stopping an executing program, execution of the current instruction is completed and then stops, so that the Instruction register will contain the instruction just completed. The Instruction register may be changed by the Data Entry switches.

⑦ DATA ENTRY switches (0 through 15) are used to enter data or instructions into the selected register. These switches may be tested as sense switches or used as data input switches during program execution. Data switches in down position = 1, in up position = 0.

⑧ ENTER switch. In the IDLE mode, this switch replaces the contents of the selected register with the contents of the Data Entry switches.

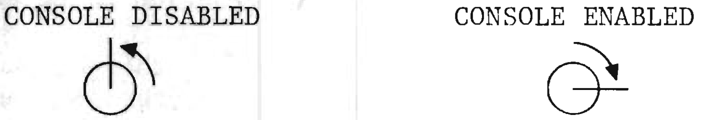
⑨ SAVE I switch preserves the contents of the Instruction register for repeated execution (e.g., referencing memory locations using the same load/store instruction) in the IDLE mode. SAVE I is effective only in the IDLE mode. The primary function of this switch is explained in Section 4.2.2.

⑩ STEP switch. When this switch is depressed, the computer fetches an instruction from the location specified by the P-register, executes the instruction and increments the P-register. In the IDLE mode, only a single instruction cycle executes.(If the SAVE I switch is down, the current contents of the I-register are executed and preserved; the P-register is still incremented however. In the RUN mode automatic program execution is initiated by pressing STEP.

⑪ RUN/IDLE switch is used to place the computer in either the RUN mode (switch up) or the IDLE mode (switch down).

⑫ SYSTEMS RESET switch is used to initialize the computer and peripheral controllers. This switch sets foreground mode, if foreground/background option is installed, (shown by FOREGROUND MODE indicator ⑮) and disables all interrupts.

⑬ CONSOLE ENABLE Key Lock. This is a two position lock, as follows:



Turning the key counterclockwise leaves the console disabled (i.e., changing any of the console switches has no effect). Turning clockwise will enable the console switches. The console may be enabled/disabled in either the RUN or IDLE mode.

It was General Automation's intention to design the SPC-16 console to be able to be used as a "Systems-Operating-Console". In line with this the program had complete control over the console-display-register (K) and, it could "sense" the console data switches under program control for manual input commands. The console lock-out feature was built in as part of the fail-safe features required for an on-line real-time system in which only qualified operators could affect the operation of the system. Rather than "lock-out" the ability to sense the console switches, General Automation provided the means whereby either the "locked-out" or the "unlocked-out" approach could be used in sensing the console switches.

The "unlocked-out" approach means that the program senses the switches at will for manual input commands and anybody can change the switches at any time to implement these commands, independently of whether or not the console is locked.

The "locked-out" approach is accomplished by a software philosophy that requires a console-interrupt prior to sensing the console data switches for manual input commands, i.e., the console-interrupt is dedicated to initiating the sensing of the console data switches. Since the console-interrupt cannot be initiated unless the console is unlocked, then the operation is a "locked-out" one.

⑭ CONSOLE INTERRUPT switch. Pressing this switch with the console unlocked causes a console-interrupt request.

⑮ FOREGROUND MODE indicator indicates which set of eight programmable registers is being used. The indicator is on for foreground, off for background. If the foreground/background option is not installed, this indicator is always off.

⑯ READ ONLY MEMORY ACCESS indicator indicates when Read Only Memory is being accessed for instructions or data.

⑰ DIRECT MEMORY ACCESS/AUTOMATIC DATA CHANNEL ACKNOWLEDGE indicator indicates that a DMA cycle "steal" is taking place.

⑱ INTERRUPT ACKNOWLEDGE indicator indicates when an interrupt is being serviced. Manually stepping through an interrupt will result in the address of the branch vector being displayed in the indicators. Exit must be made by pressing STEP (to execute the interrupt) or SYSTEMS RESET (to reinitialize the system).

⑲ INTERRUPT ENABLE indicator indicates the system interrupt status:

On = ENABLED Off = DISABLED

⑳ OPERATIONS MONITOR ALARM indicator indicates when the alarm has been activated.

㉑ IDLE indicator indicates when the computer is in the IDLE mode.

㉒ RUN indicator indicates when the computer is in the RUN mode.

NOTE

When executing a program in the RUN mode, and a WAIT instruction is encountered, both the RUN and IDLE indicators come on.

See Section 6.2 for a brief discussion of the dynamic SAVI switch located on the rear of the console board.

INDEX

INDEX (Continued)

INDEX (Continued)

INDEX (Continued)

INDEX (Continued)

INDEX (Continued)

INDEX (Continued)

COMMENT SHEET

FROM:

NAME: ______________________________

BUSINESS ADDRESS: ______________________________

____

____

Does this publication meet your requirements? Yes □ No □

If no, please explain. ______________________________

____

____

____

____

Do you wish a reply? Yes □ No □

COMMENTS: (Describe any errors, suggested ideas, additions, or deletions etc. Please include page number.) All comments and suggestions become the property of General Automation, Inc.

____

____

____

____

____

____

____

____

No Postage Stamp Necessary If Mailed In the U.S.A.
(See Other Side)
Fold On Dotted Lines And Staple

YOUR COMMENTS, PLEASE . . .

This publication serves as a reference for systems analysts, programmers and operators of General Automation systems. Your answers to the questions on the back of this form help us produce better publications for your use.

FOLD

CUT ALONG LINE

FIRST CLASS
PERMIT NO. 423
ANAHEIM, CALIF.

BUSINESS REPLY MAIL
No Postage Necessary if Mailed in the United States

Postage Will Be Paid By . . .

GENERAL AUTOMATION, INC.
1055 South East Street
Anaheim, California 92803

Attention: Technical Publications

FOLD

General Automation, Inc.
1055 South East Street
Anaheim, California 92803

ADDITIONAL COMMENTS:

GENERAL AUTOMATION, INC.
1055 South East Street, Anaheim, California 92805 (714) 778-4800