POINT 4™

MARK 3

COMPUTER SYSTEM MANUAL

POINT 4 DATA CORPORATION

POINT 4™

MARK 2/3

COMPUTER SYSTEM MANUAL

POINT DATA CORPORATION

Revision D

NOTICE

Every attempt has been made to make this manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 863-1111

REVISION RECORD

PUBLICATION NUMBER: HM-081-0019

Revision	Description	Date
01	Draft Version to coincide with delivery of first MARK 3	05/15/81
A	Complete revision incorporating corrections and new material; PIB interface is now covered in the MARK 3 Peripherals Interface Manual	06/09/82
B	Update package incorporating information relevant to the MARK 3B and correction to illustration on page 2-24	12/10/82
(C) WLSS. M'S	Update package incorporating information relevant to the MARK 2/3 Tabletops and the Lowboy configurations	02/20/84
D	Update package incorporating information relevant to the MARK 2 with floppy disk drive	12/01/84

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed. The effective revision for each page is shown below.

Page	Rev	Page	Rev	Page	Rev
Cover	-	5-26, 5-27	A		
Title	D	5-28, 5-29	B		
ii thru iv	D	5-30	A		
v thru viii	C	5-31	B		
ix	A	5-32, 5-33	A		
x, xi	C	5-34	C		
1-1	C	6-1 thru 6-5	A		
1-2 thru 1-4	D	Appendix Title	-		
1-5 thru 1-11	C	A-1	A		
1-12	D	B-1	A		
1-13, 1-14	C	C-1	C		
2-1 thru 2-3	C	D-1	C		
2-4	A	E-1 thru E-4	A		
2-5	C	E-5	C		
2-6	D	E-6 thru E-10	A		
2-7, 2-8	C	Comment Sheet	D		
2-9	B	Mailer	-		
2-10 thru 2-12	C	Back Cover	-		
2-13, 2-14	B				
2-15	C				
2-16	B				
2-17, 2-18	C				
2-19	B				
2-20	C				
2-21	B				
2-22 thru 2-25	C				
3-1 thru 3-5	A				
3-6 thru 3-8	D				
3-9, 3-10	B				
3-11 thru 3-20	A				
4-1 thru 4-3	B				
4-4	A				
4-5	B				
5-1 thru 5-24	A				
5-25	B				

PREFACE

This manual describes the POINT 4 MARK 3 Computer System, a 16-bit, general-purpose minicomputer with a versatile instruction set.

The introduction describes standard features of the POINT 4 MARK 3 and includes detailed information on equipment and performance characteristics, and system architecture. Optional features are described in a separate section.

The MARK 3 system offers a choice of two Peripheral Interface Boards (PIB). The PIB is available in two versions. Both versions contain a built-in multiplexer and a disc drive interface. Additionally, the MARK 3 PIB provides a tape drive interface; the MARK 3B PIB provides a floppy disc drive interface.

Step-by-step directions for installation, upgrade, and detailed operating procedures are provided. Sections describing the input/output interface and the standard instruction set are included.

The appendices provide cabling information, summary information of POINT 4 MARK 3 commands and instructions, and programming examples.

Related manuals include:

Title	Pub. Number
POINT 4 MARK 3 Peripherals Interface Manual	HM-081-0026
POINT 4 MARK 3 Diagnostics Manual	

CONTENTS

APPENDICES

FIGURES

TABLES

Section 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

The POINT 4 MARK 3 Computer is a 16-bit general-purpose minicomputer with a versatile instruction set. The POINT 4 MARK 3 employs a novel design architecture (Microprogrammed Sequencer*) to achieve the simplicity and flexibility of a microprogrammed design with the speed of a hard-wired logic design. In addition, the design allows direct addressing of up to 128K bytes of MOS random access memory. These features make the POINT 4 MARK 3 Computer well suited to OEM applications in business data systems, and control systems. See Figure 1-1 for a photograph of the POINT 4 MARK 3 Computer chassis.

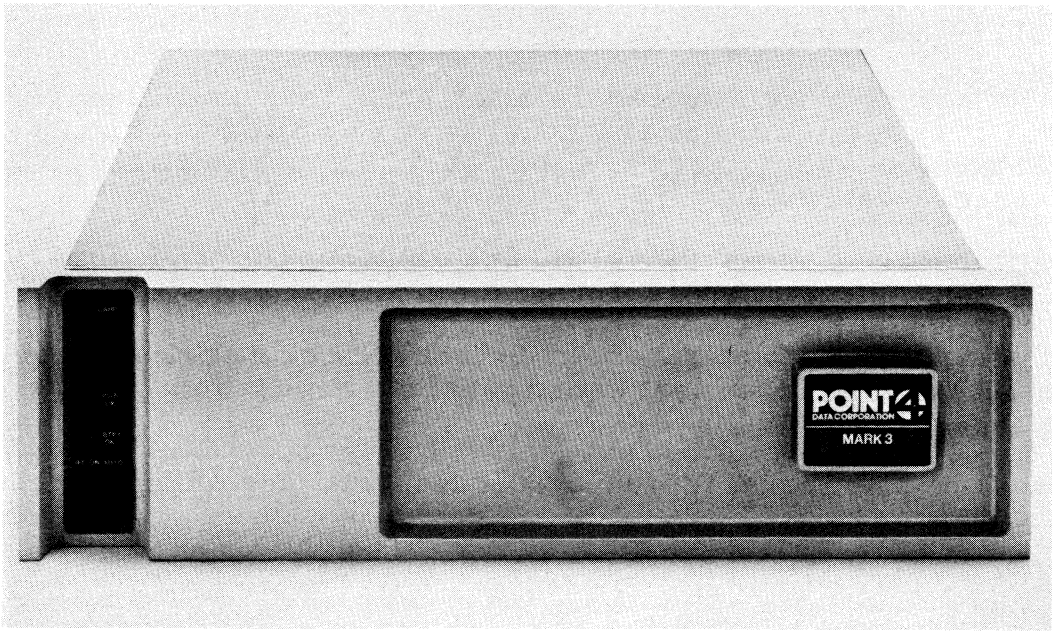


Figure 1-1. POINT 4 MARK 3 Computer Chassis

*Patent No. 4,370,729

1.1.1 POINT 4 MARK 3 COMPUTER SYSTEM CONFIGURATIONS

A POINT 4 MARK 3 Computer System is available in three different configurations. All configurations use the MARK 3 CPU board with the appropriate PROMs and all allow optional memory (up to 128K-bytes) and port (up to seven ports) expansion.

On a POINT 4 MARK 3 Computer System, the CPU and the PIB boards may be mounted in any order. However, mounting the CPU board in the top slot and the PIB in the second slot is the most convenient arrangement.

On a POINT 4 MARK 2 Computer System, the PIB must be mounted in the top slot.

1.1.1.1 Rack-mounted Configuration

Separate components are shipped for mounting in a standard computer rack. This is the most flexible of the MARK 3 Systems. It is available with either a MARK 3 or a MARK 3B PIB and can interface the maximum number of disc, tape, or floppy disc drives. Refer to Section 1.1.2 for a list of specific features. Unpacking and installation procedures are supplied in this manual.

1.1.1.2 Lowboy Configuration

The MARK 3 Computer System is available in a preassembled lowboy configuration consisting of the computer chassis and a rack-mounted peripheral subsystem in a movable and easily accessible cabinet.

The computer chassis contains the CPU board, the power supply, and a MARK 3 Peripherals Interface Board.

The peripherals consist of an 8-inch Winchester (SMD) disc drive, a 1/4-inch (QIC-02) streaming tape drive, and the peripheral power supply unit.

Unpacking and installation procedures are described in the MARK 3 Lowboy Tech Memo.

1.1.1.3 Tabletop Configurations

The POINT 4 MARK 3 CPU board with appropriate PROMs is the basis for two types of tabletop models: the MARK 3T and the MARK 2.

1. The MARK 3 Tabletop is housed in two stacked cabinets. The top cabinet houses the computer chassis which contains the CPU, the MARK 3 PIB, and the power supply.

   The bottom cabinet contains an 8-inch Winchester (SMD) drive, a 1/4-inch (QIC-02 interface) streaming tape drive, and a power supply unit.

   Unpacking, installation, and the use of the MARK 3T configuration is described in the POINT 4 MARK 3 Tabletop Computer User Guide.

2. The MARK 2 is housed in a single cabinet which contains the CPU, the MARK 2 PIB, power supply units, ST506 disc drive, and either a 1/4-inch (QIC-02) streaming tape drive or a 5-1/4-inch floppy disc drive. The MARK 2 PIB must be mounted in the top slot.

   Unpacking, installation, and the use of the MARK 2 System are described in the POINT 4 MARK 2 Computer User Guide.

1.1.2 FEATURES

The POINT 4 MARK 3 Computer includes the following features:

- CPU and 64K bytes of RAM on the same board
- CPU has special MANIP and Microcode PROMs specific to the peripheral interface
- Depending on the model, one of the following Peripheral Interface Boards (PIBs) may be included in the configuration:

  MARK 3 PIB with built-in:
  Multiplexer
  Disc Interface (SMD)
  Tape Interface (QIC-02)

  MARK 3B PIB with built-in:
  Multiplexer
  Disc Interface (SMD)
  Floppy Disc Interface (Shugart compatible)

  MARK 2 PIB with built-in:
  Multiplexer
  Disc Interface (ST506)
  Tape Interface (QIC-02)
  Floppy Disc Interface

  See the MARK 2/3 Peripherals Interface Manual.

- Four asynchronous ports, jumper selectable to 9600 baud
- Handles SMD/CMD/LMD/FMD/MMD disc drives supporting the sector-mark interface signal
- Depending on the model, interfaces up to two disc drives
- Depending on the model, interfaces to a streaming tape or floppy disc drive
- Disc transfer rates to 1.25 megabytes per second
- Tape transfer rates to 90 kilobytes per second (at 90 inches per second)
- Floppy disc transfer rates to 500 kilobits per second (for double density)
- Virtual control panel
- Internal power supply for CPU and PIB boards
- Optional 64K-byte Memory Expansion Board for a total of 128K bytes
- Optional Port Expansion Board providing three additional ports for a total of seven ports
- 100-pin backplane connector (special POINT 4-designed bus)

1.3 SYSTEM ARCHITECTURE

The POINT 4 MARK 3 architecture has been streamlined to create a system with a minimum of signal interfacing between boards and with a maximum speed of instruction execution. A combination of central processor logic and up to 64K bytes of RAM on a single circuit board eliminates time delays due to long memory access bus paths. The combination of all I/O control functions on a Peripheral Interface Board provides maximum efficiency and minimum backplane signal travel. An integrated power supply provides power for all POINT 4 MARK 3 functions. Figure 1-2 illustrates a typical configuration of a POINT 4 MARK 3 Computer System.

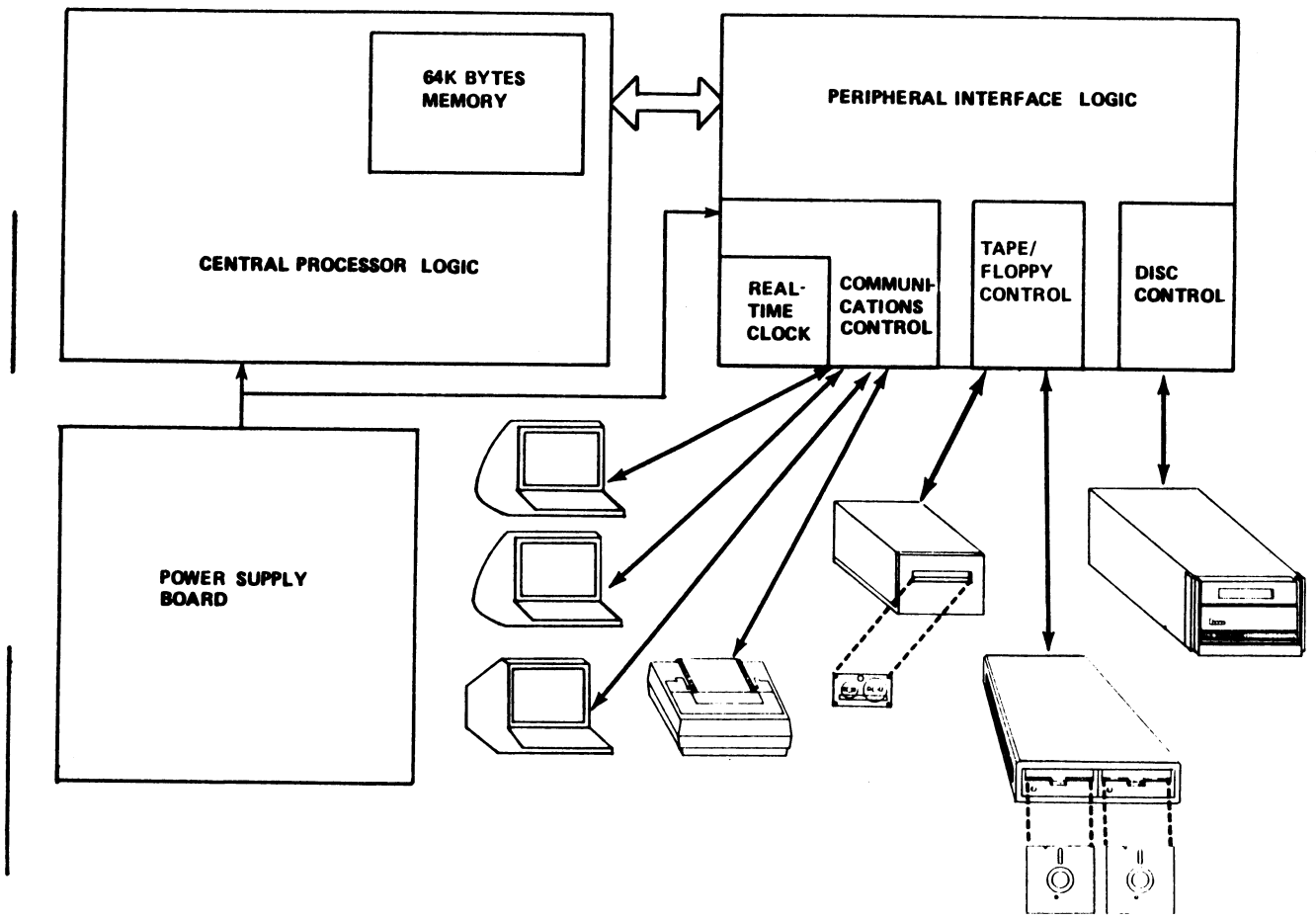


Figure 1-2. Typical POINT 4 MARK 3 Configuration

1.3.1 SYSTEM FUNCTIONAL UNITS

The POINT 4 MARK 3 is comprised of five functional units:

- CPU and Memory Board
- Peripheral Interface Board
- Power Supply Board
- Chassis and front panel
- Mini-panel

Figure 1-3 is a block diagram of the basic units of the computer system showing the functions each unit performs.

1.3.1.1 Central Processor and Memory Board

The Central Processor board contains all basic elements of the CPU:

- 2903 Microprocessor Bit Slice (4 bits wide) - contains four general-purpose accumulators, plus 12 special-purpose registers and arithmetic/logic functions
- Microprogrammed Control Store (512 words x 64 bits) - contains firmware to implement the software instruction set, as well as the DMA multiplexer, disc controller and tape or floppy disc controller functions for the PIB board
- Instruction Register
- Main Data Bus
- Program Counter
- Effective Address Register
- Timing Control
- Input/Output Control
- 64K Bytes of Random Access Memory (RAM)
- MANIP Proms - contain a program to implement Virtual Control Panel Features, and the Self Test

Figure 1-4 is a block diagram of the POINT 4 MARK 3 System, showing logic to handle each of the above functions.

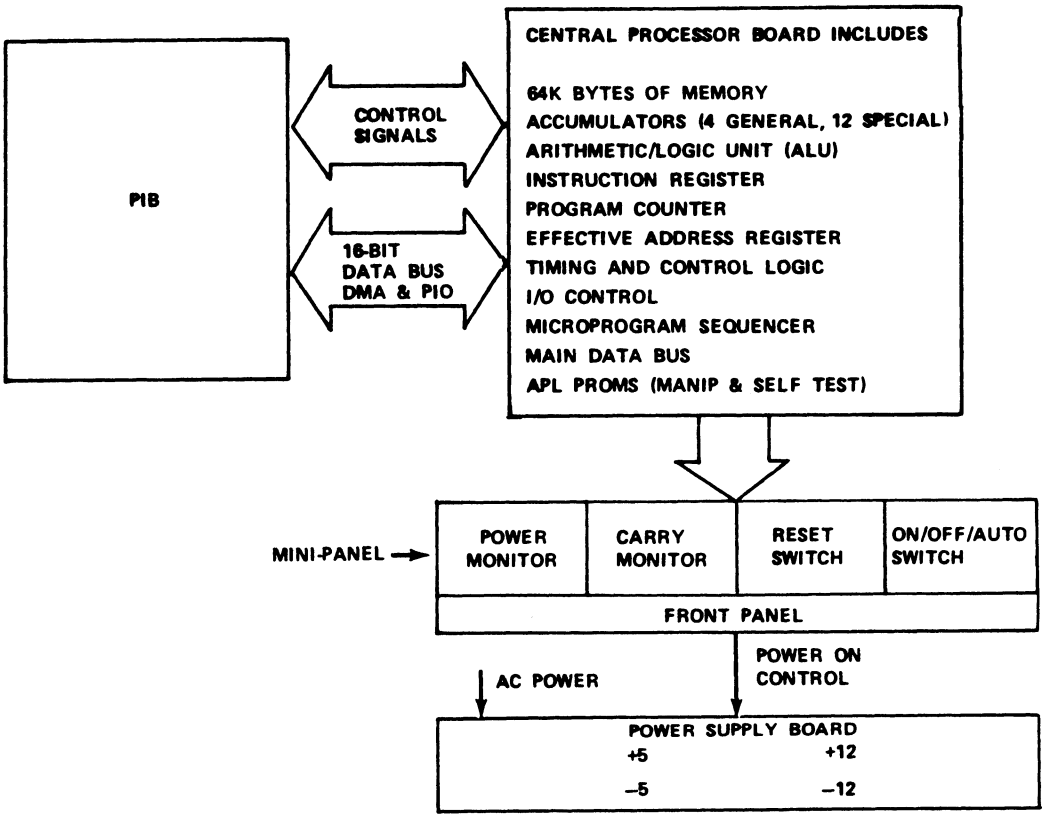


Figure 1-3. POINT 4 MARK 3 Computer System Block Diagram

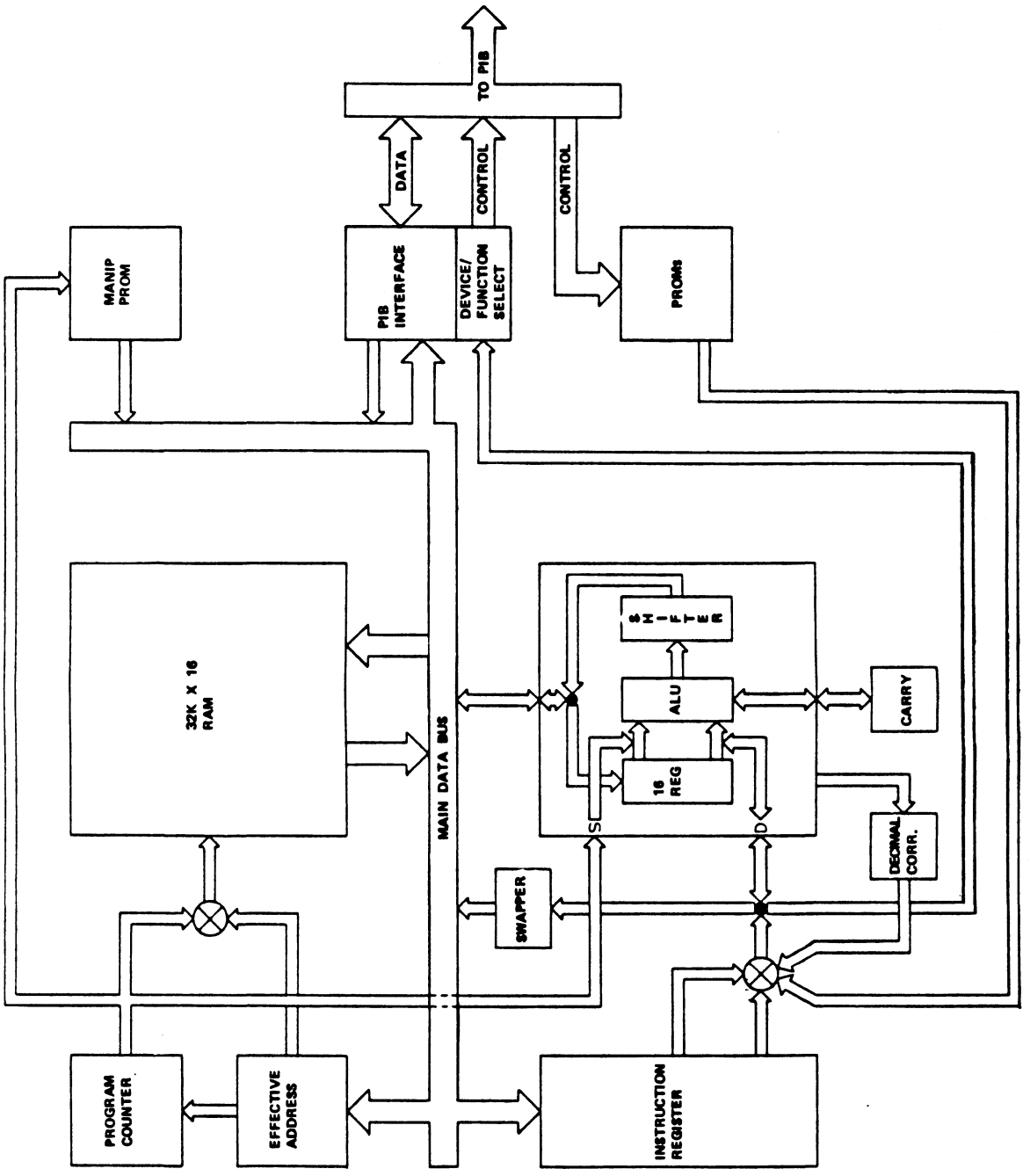


Figure 1-4. POINT 4 MARK 3 CPU/Memory Block Diagram

1.3.1.2 Peripheral Interface Board

The Peripheral Interface Board (PIB) contains all elements necessary for control of peripheral devices connected to the POINT 4 MARK 3. Functions included are:

GENERAL CONTROL FUNCTIONS

- Command Decoding
- Data Transmission/Reception
- Device Decoding
- Status Recording
- CRC Generation

MULTIPLEXER CONTROL FUNCTIONS

- Real-Time Clock
- Baud Rate Selection
- Data Synchronization & Format Manipulation
- Port Selection

DISC CONTROL FUNCTIONS

- Disc Selection
- DMA Bus Control
- Word Count
- Tag Line Control

TAPE CONTROL FUNCTIONS*

- Tape Drive Selection
- Read/Write Operations
- Tape Positioning
- Status Reporting
- Error Processing/Recovery

FLOPPY DISC CONTROL FUNCTIONS**

- Floppy Disc Drive Selection
- Side Selection
- Address Mark Detection
- Status Reporting
- FM and MFM Encode and Decode Logic

The Peripheral Interface Board is described in a separate manual which provides detail of all aspects of peripheral device control (see the MARK 3 Peripherals Interface Manual).

*available on the MARK 3 PIB
**available on the MARK 3B PIB

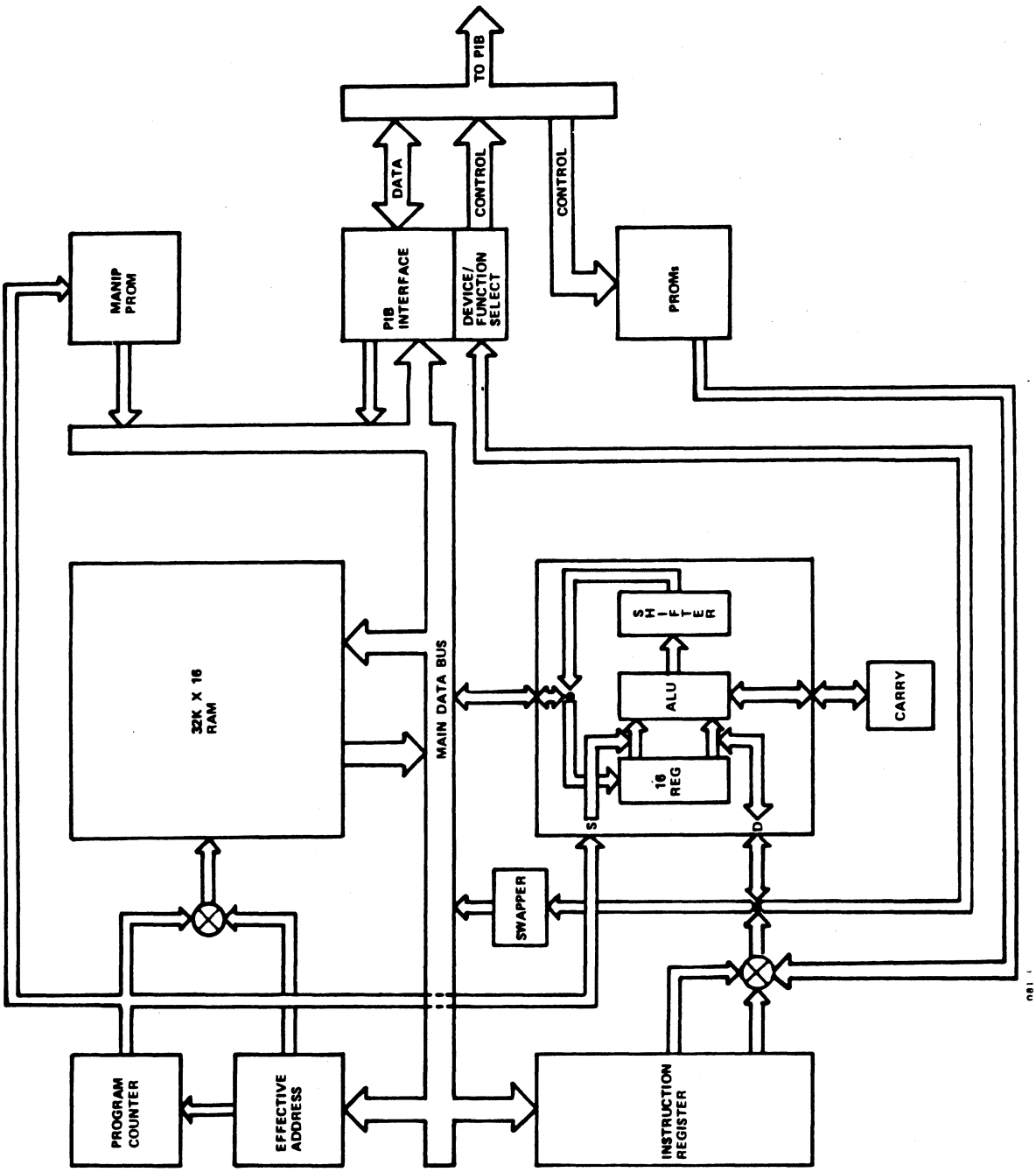


Figure 1-4. POINT 4 MARK 3 CPU/Memory Block Diagram

1.3.1.2 Peripheral Interface Board

A Peripheral Interface Board (PIB) contains the elements necessary for control of peripheral devices. Functions included are:

GENERAL CONTROL FUNCTIONS (all PIBs)

- Command Decoding
- Data Transmission/Reception
- Device Decoding
- Status Recording
- CRC Generation

MULTIPLEXER CONTROL FUNCTIONS (all PIBs)

- Real-Time Clock
- Baud Rate Selection (Hardware strappable)
- Data Synchronization & Format Manipulation
- Port Selection

DISC CONTROL FUNCTIONS (MARK 3 and MARK 3B PIBs only)

- Disc Selection
- DMA Bus Control
- Word Count
- Tag Line Control

ST506 DISC CONTROL FUNCTIONS (MARK 2 PIB only)

- DMA Bus Control
- Sector Buffer
- Error Detection and Correction
- Modulated Frequency Modulation (MFM) Encode and Decode Logic

TAPE CONTROL FUNCTIONS (MARK 2 and MARK 3 PIBs only)

- Tape Drive Selection
- Read/Write Operations
- Tape Positioning
- Status Reporting
- Error Processing/Recovery

FLOPPY DISC CONTROL FUNCTIONS (MARK 2 and MARK 3B PIBs only)

- Floppy Disc Drive Selection
- Side Selection
- Address Mark Detection
- Status Reporting
- FM and MFM Encode and Decode Logic (MARK 3B PIB only)

The Peripheral Interface Boards are described in a separate manual which provides detail of all aspects of peripheral device control (see the MARK 2/3 Peripherals Interface Manual).

Section 2
INSTALLATION

2.1 ENVIRONMENTAL REQUIREMENTS

Careful consideration must be given to the placement of the POINT 4 MARK 3 prior to installation to ensure that all power and environmental requirements are met. Necessary preinstallation considerations are discussed in the following subsections.

2.1.1 POWER REQUIREMENTS

The POINT 4 MARK 3 requires a power source of 117 VAC, +10%/-15%; 47 to 63 Hz with 5 amperes current draw; or a 234 VAC, +10%/-15%; 47 to 63 Hz power source with 2.5 amperes current draw. In addition to power requirements for the POINT 4 MARK 3, the power resources and electrical outlets needed to handle all peripheral devices must be considered.

2.1.2 TEMPERATURE REQUIREMENTS

Adequate environmental controls are needed to maintain the POINT 4 MARK 3 within the maximum operating range of 0 to 50 degrees C (32 to 122 degrees F), and relative humidity range of 10 to 90 percent, noncondensing.

2.1.3 ENCLOSURE REQUIREMENTS

The POINT 4 MARK 3 is packaged in a 3-slot chassis, measuring 5.25 inches high, 19 inches wide, and 17.5 inches deep. The CPU/Memory PCB, Peripheral Interface Board, and Power Supply Board have the same dimensions: 14.5 inches x 12 inches.

2.2 UNPACKING INSTRUCTIONS

The POINT 4 MARK 3 receives a complete test and inspection prior to shipment. Inspect each unit for completeness and shipping damage prior to installation. Inspect each carton for any evidence of damage due to dropping, puncturing, or crushing. If damage is evident, contact the carrier and the POINT 4 Data Corporation Sales Representative for further instructions.

2.2.1 UNPACKING THE CARTONS

The POINT 4 MARK 3 is packaged in a double-walled corrugated carton. Styrofoam packing inserts surround the chassis. Figure 2-1 illustrates the processor chassis packaging.

2.2.2 CONTAINER CONTENTS

Check each item removed from the carton against the packing slip. Inspect all items including cable connectors for damage. If items are damaged or broken, contact the POINT 4 Data Corporation Sales Representative.

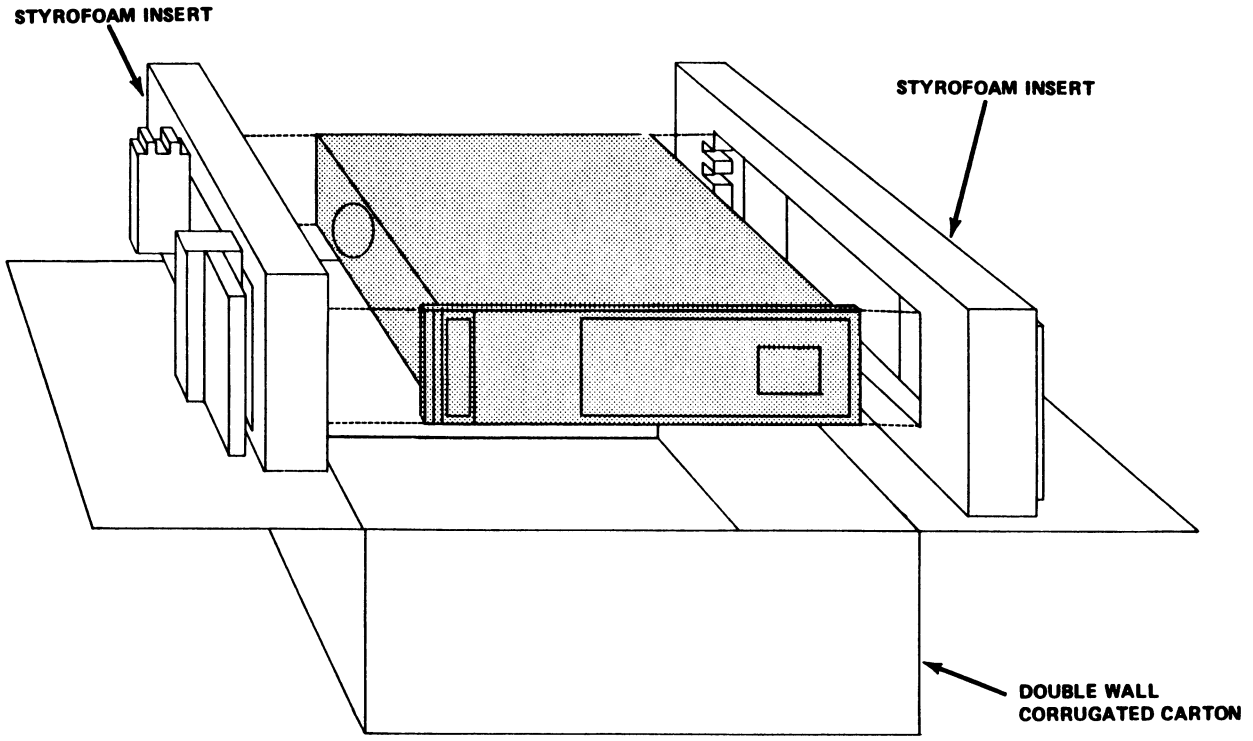


Figure 2-1. POINT 4 MARK 3 Processor Chassis Packaging

2.3 MOUNTING INSTRUCTIONS

The processor is designed to be mounted in a standard 19-inch equipment rack. Although the exact procedure may vary, mount the chassis according to the following general procedure.

2.3.1 FRONT PANEL

The POINT 4 MARK 3 front panel snaps off for easy removal. No screws or hinges hold it in place. Removing the front panel will reveal mounting slots on each side of the chassis (see Figure 2-2).

2.3.2 CHASSIS

Once the front and back of the chassis have been sufficiently supported, flanges are available to secure the chassis to the cabinet.

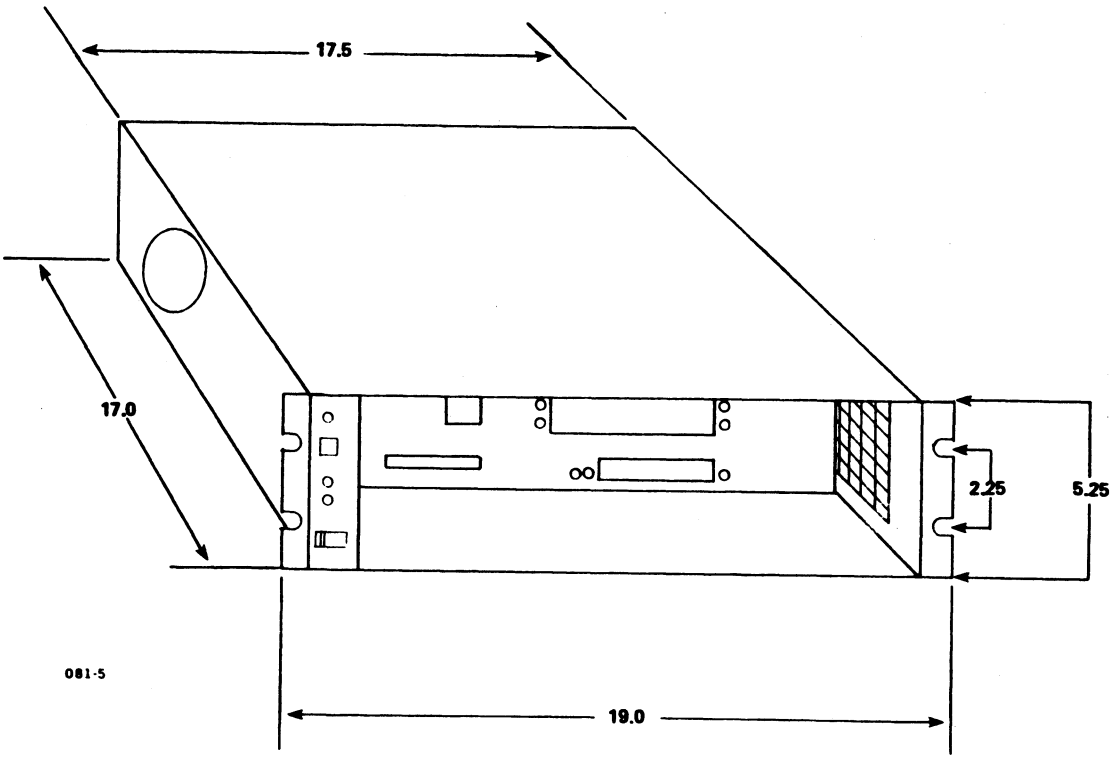


Figure 2-2. POINT 4 MARK 3 Chassis Mounting Slots

2.4 MARK 3 INSTALLATION PROCEDURE

This section details the installation procedure for the MARK 3 system. It covers board checkout, connecting MUX cables, and powering up the system. See Section 2.6 for cabling diagrams and information. To upgrade an existing system, see Section 2.5.

Installation procedures for a MARK 3 Lowboy and the Tabletop models are described in the documentation that accompanies those models.

If the MARK 3 to be installed has the extended capacity of 128KB memory and/or 7 ports, the CPU and/or the PIB board will have a piggyback expansion board. This does not change the installation procedure.

Before beginning with the installation procedure, make room on a table or other solid surface for the chassis and at least two boards.

2.4.1 CHECKING THE BOARDS

The first phase in the installation process involves checking for damage and making sure everything is connected properly. An improper connection may cause damage when power is turned on.

The CPU board is usually mounted in the top slot and the PIB in the second slot because this is the most convenient arrangement. The user may mount the boards in any order.

1. On the rear of the chassis, remove the eight screws holding the rear panels in place (see Figure 2-3).

2. Remove both panels and set them aside. Three boards will be visible: in the first (top) slot is the CPU/Memory Board, in the second slot the Peripheral Interface Board (PIB), and in the third slot (bottom) the Power Supply Board (see Figure 2-4).

3. Pull out the white tabs at the sides of each board.

4. Check the CPU Board:

   a. Back out the CPU board (top slot) and remove any packing material surrounding the board.

   b. Press down on each prom and any component using connectors to make sure it is seated properly. (Do not remove any, just press down on them.) Inspect the board for damage.

   c. If the CPU board has a Memory Expansion Board, make sure it is seated properly.

   d. Carefully reinstall the CPU board into slot 1, component side up, making sure that the card edge connectors slide smoothly into the backplane sockets.

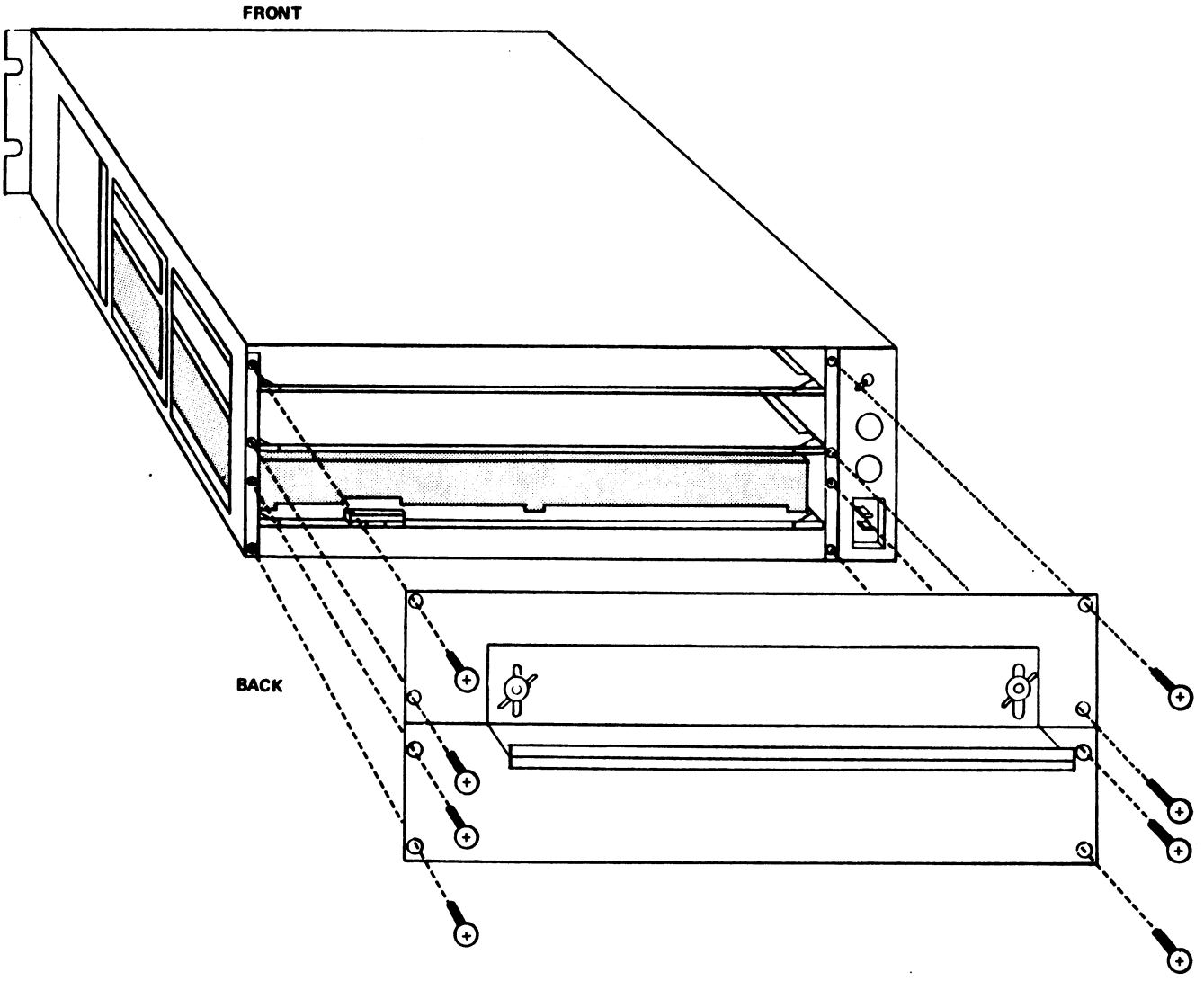


Figure 2-3. POINT 4 MARK 3 Chassis with Rear Panel Open Showing Mounting Slots

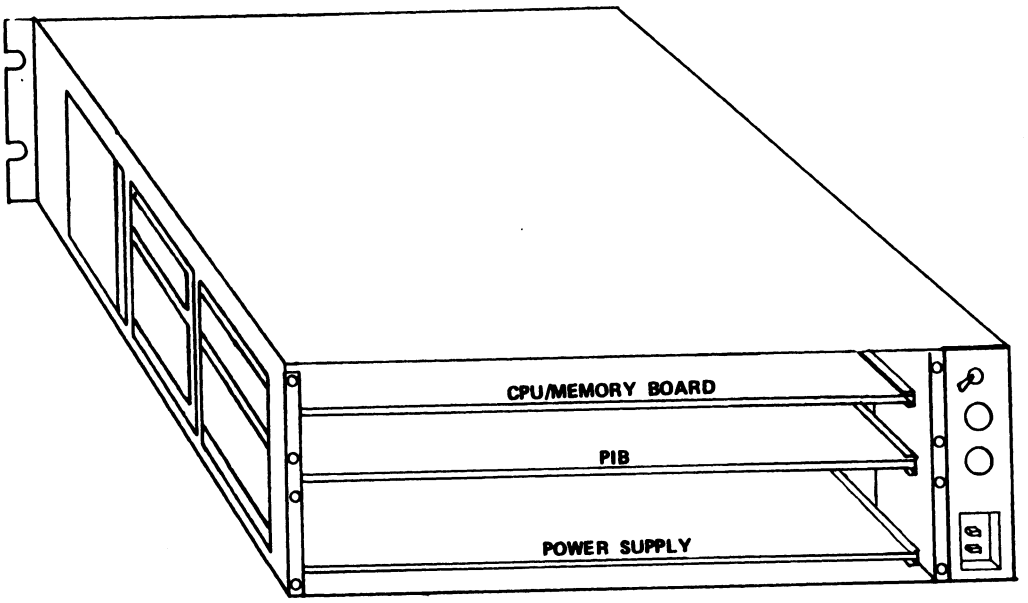


Figure 2-4. POINT 4 MARK 3 Board Configuration

5. Check the Power Supply Board:

   a. Remove any packing material surrounding the Power Supply Board.

   b. Push in on the Power Supply Board to make sure it is connected properly.

6. Check the PIB Board:

   a. Back-out the PIB board and remove any packing material surrounding it.

   b. Press down on the header plugs to make sure they are seated properly.

   c. If the PIB board has a Port Expansion board, make sure it is seated properly.

   d. Slide the board back into slot 2, component side up, leaving approximately two or three inches extended.

2.4.2 CONNECTING MUX CABLES

The next phase of the installation procedure is to connect the cables. For cabling information, see Section 2.6.

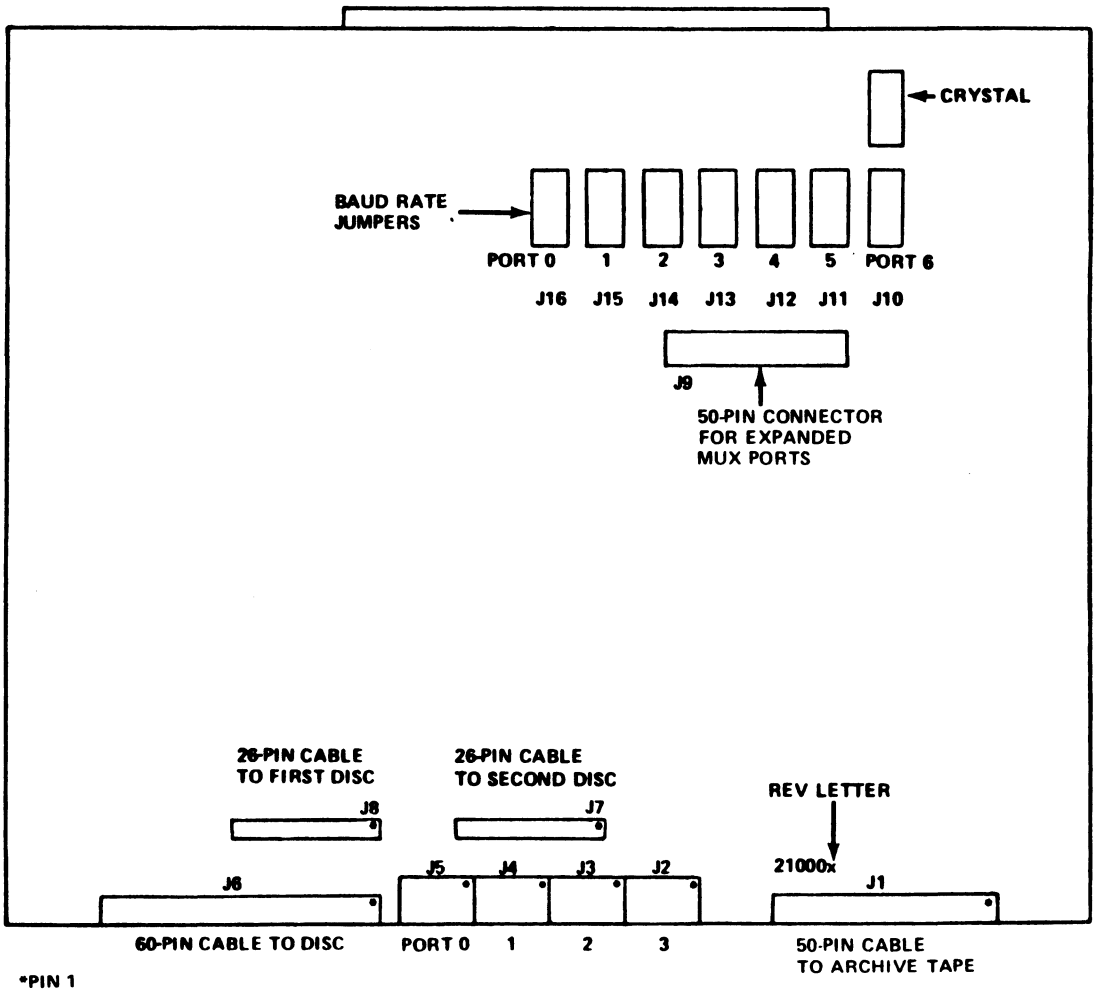
The MUX ports on either PIB are numbered 0-3. The expansion board ports are numbered 4-6. Port 0 is used as the master port; under IRIS, the printer is initially set up for Port 3.

NOTE

The printer may be connected to a port other than port 3 by changing the software (for an IRIS R7 system, see the IRIS R7.5 Release Notes; for an IRIS R8 system, see the IRIS Installation and Configuration Manual).

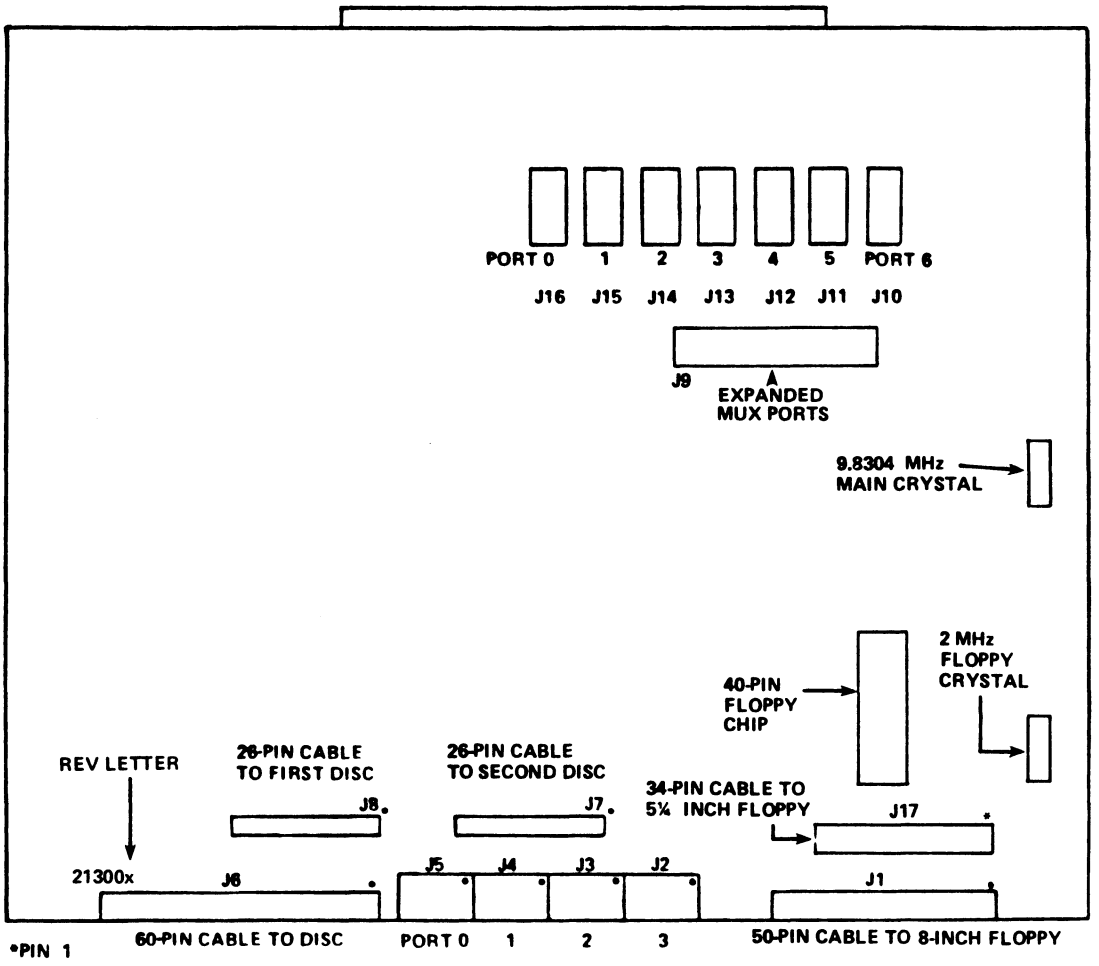
Cable connector locations for the MARK 3 PIB (with Tape Drive Interface) are shown in Figure 2-5, and for the MARK 3B PIB (with Floppy Disc Drive Interface) are shown in Figure 2-6.

1. To connect the master port CRT, plug the cable into connector J5. This is the leftmost of the four white Molex connectors.



STREAMER TAPE CABLE	J1
PRINTER OR CRT CABLES	J2, J3, J4
MASTER CRT CABLE	J5
DISC CABLE A	J6
DISC CABLE B – NO. 2	J7
DISC CABLE B – NO. 1	J8

Figure 2-5. Cable Connector Positions on a POINT 4 MARK 3 Peripheral Interface Board (with Tape Drive Interface)



EXTERNAL CABLES WHICH PLUG INTO THE PIB BOARD:

8-INCH FLOPPY	J1
PRINTER OR CRT CABLES	J2, J3, J4
MASTER CRT CABLE	J5
DISC CABLE A	J6
DISC CABLE B – NO. 2	J7
DISC CABLE B – NO. 1	J8
5¼-INCH FLOPPY CABLE	J17

Figure 2-6. Cable Connector Positions on a POINT 4 MARK 3B Peripheral Interface Board (with Floppy Disc Drive Interface)

2. To connect the disc drive:

   a. Plug Cable "A" into connector J6 (60-pin, left side of PIB) with pin 1 of cable to the right edge of the connector. Make sure connector is not shifted to right or left.

   b. Plug Cable "B" into connector J8 (26-pin, behind J6) with pin 1 of cable to the right edge of the connector.

   c. If a second drive is to be connected, plug the second "B" cable into J7 (to the right of J8).

3. If a streamer tape drive is to be connected (requires MARK 3 PIB), plug the 50-pin cable into J1 (right side of PIB) with pin 1 to the right edge.

4. If the system will have one or more floppy disc drives (requires MARK 3B PIB), the board will be set up for either 8-inch or 5-1/4-inch drive(s):

   a. To connect 8-inch floppy disc drive(s), plug the 50-pin cable into J1 (right side of PIB) with pin 1 to the right edge.

   b. To connect 5-1/4-inch floppy disc drive(s), plug the 34-pin cable in J17 (above J1 right side of PIB) with pin 1 to the right edge.

5. If drives are to be daisy-chained, connect Daisy-chain cable according to instructions provided by the drive vendor.

6. To connect the printer, plug its cable into J2 (white, 6-pin Molex).

7. The remaining ports are normally used for CRT terminals. If a modem is used on the system, refer to section 2.6 for cabling considerations.

NOTE

All MUX ports are jumpered for 9600 baud. They may be rejumpered if another rate is desired. See the MARK 3 Peripherals Interface Manual, Section 2.2, for baud rate selection information.

8. Push the PIB board back into the chassis. Make sure the card edge connectors slide smoothly into the backplane sockets.

9. Reinstall the two back panels.

2.4.3 POWERING UP THE MARK 3 SYSTEM

It is necessary to make sure that all connectors are seated properly before the electrical current is turned on. An improper connection may result in damage (see Section 2.4.1).

The following steps should be followed when first applying power to the POINT 4 MARK 3 power supply and processor:

1. Make sure the ON/OFF switch is turned to the OFF position.

2. Plug the AC cord into the rear right of the chassis.

3. Check that the grey cable connecting the Mini-panel to the backplane is plugged in.

4. Turn the power switch at the rear of the chassis to the ON position. The POWER OK indicator should illuminate indicating that all voltages are in tolerance.

5. Turn the power-control switch on the Mini-panel to ON.

6. The carry light comes on for approximately 1.5 seconds, goes off for .5 seconds, comes on again and remains illuminated.

7. The Master CRT displays OK, indicating that the system has run a successful CPU self-test program.

8. Run Self-Test (see Section 3.5).

9. POINT 4 provides diagnostics on disc, cassette tape or floppy disc. Load the diagnostics from the appropriate media and then run the program (see the POINT 4 MARK 3 Diagnostics Manual).

10. If the disc drive has been turned on, has a system disc, and is in a Ready state, the MARK 3 will read block 0 from the disc and idle at location 377, waiting to be overwritten by DMA (standard IPL).

11. If the disc drive is not in a Ready state:

    a. Press the RESET button on the Mini-Panel, which accesses the MANIP program. The CRT displays the contents of the program counter and accumulators.

    b. Press P (program load from disc) on the CRT keyboard, followed by a <RETURN>. The MARK 3 will read block 0 from the disc and idle at location 377, waiting to be overwritten by DMA (standard IPL).

12. To IPL the system from a floppy, press F on the CRT keyboard, followed by <RETURN>. The MARK 3 will read block 0 from the floppy and idle at location 377, waiting to be overwritten by DMA (standard IPL).

2.5 UPGRADING AN EXISTING MARK 3

A MARK 3 system may be upgraded to 128KB memory and/or seven MUX ports with expansion boards. If the existing CPU and PIB boards meet certain minimum specifications, expansion boards may be installed in the field.

Requirements and procedures given in this section are intended as guidelines. For specific information refer to the literature supplied with the expansion boards.

2.5.1 UPGRADING A MARK 3 TO 128KB MEMORY

If an existing 64KB MARK 3 system meets minimum requirements, it may be field-upgraded to 128KB memory by installing a Memory Expansion Board. These requirements and the installation procedure are outlined in the following sections.

2.5.1.1 Memory Expansion Requirements

Before the Memory Expansion Board is added, the following requirements must be met:

1. CPU board revision:

   Any A or B Revision CPU board that has been brought up to ECO #465 can be field-upgraded. If the board has a jumper from 3H pin 6 to 2F pin 1 (part of ECO #465), the ECO is installed. If not, the board should be returned to POINT 4 for installation of all ECOs through #465. Call POINT 4 for a Return Authorization Number.

   Any Revision C (or later) board can be field upgraded.

   NOTE

   REV letter is located on extractor end of board "CPU 20000 _".

2. Required CPU board parts (see Figure 2-7 for locations):

   a. Header plug at P3 (located between i.c. 10E and 14E). If there is no plug, one of the following may be installed:

Manufacturer	Part Number
POINT 4	500350
Berg	65610-150

   b. Correct Microcode and MANIP/Self-Test proms (refer to the instructions supplied with the Memory Expansion Board).

   c. Memory Expansion Board, POINT 4 P/N 20103.

3. For IRIS users, the operating system must be R7.5 or later.

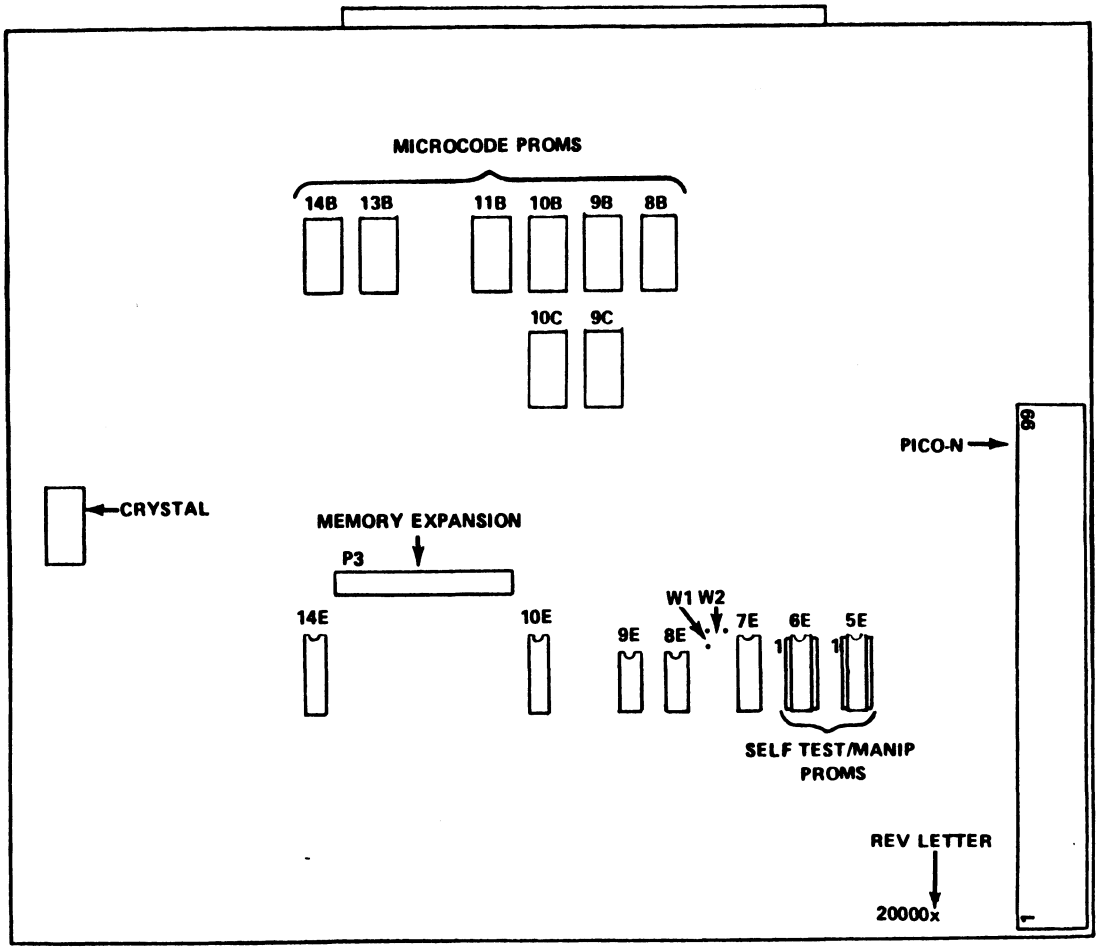


Figure 2-7. POINT 4 MARK 3 CPU Board

2.5.1.2 Memory Expansion Board Installation Procedure

To install the Memory Expansion Board, perform the following steps:

1. Test the system as follows:

   a. Run Self-Test on the CPU (see Section 3.5).

   b. Run Diagnostics on all devices on the PIB (see the POINT 4 MARK 3 Diagnostics Manual).

2. Remove the CPU board from the chassis.

3. Install header plug at P3, if necessary.

4. Move the W1 jumper (located between 7E and 8E) to W2 position.

NOTE

W1/W2 jumper must be set to one or the other, never to both.

- W1 position is for 64KB maximum
- W2 position is for 128KB maximum

5. Install the correct MANIP and Microcode proms (see instructions included with the Memory Expansion Board).

6. Plug Memory Expansion Board into 50-pin connector on CPU board and secure it with two screws (see Figure 2-8).

7. Make sure all plugable devices are fully inserted.

8. Reinstall CPU board in chassis.

9. Test the system as in step 1.

10. For IRIS users, modifications to the CONFIG file will be required to take advantage of additional memory space.

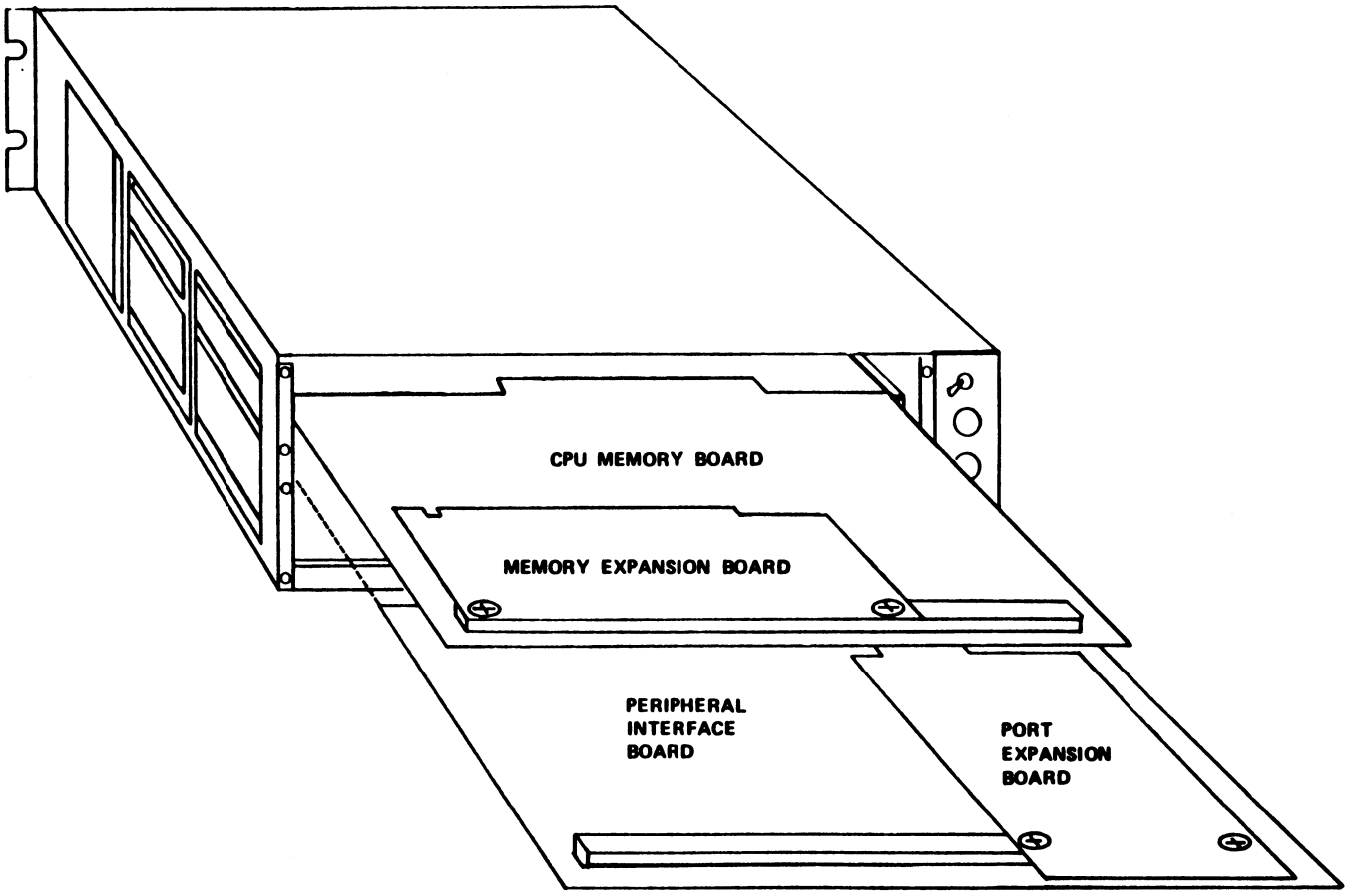


Figure 2-8. Installation of MARK 3 Memory Expansion and Port Expansion Boards

2.5.2 UPGRADING MARK 3 TO SEVEN PORTS

If an existing MARK 3 system meets minimum requirements, it may be field-upgraded to seven ports by installing the Port Expansion Board. These requirements and the installation procedure are discussed in the following sections.

2.5.2.1 MARK 3 Port Expansion Requirements

Before the Port Expansion Board is added, the following requirements must be met:

1. CPU Board requirements (see Figure 2-6 for locations):

   a. Rev A and B CPU boards must have ECO #465 installed. If the board has a jumper from 3H Pin 6 to 2F Pin 1 (part of ECO #465), the ECO is installed. If not, the board should be returned to POINT 4 for installation of all ECOs through #465. Call POINT 4 for a Return Authorization Number.

      Rev C or later CPU board does not require any ECOs to use expansion boards.

      NOTE

      Rev letter is located on extractor end of board "CPU 20000 _".

   b. Correct Microcode proms (refer to the instructions supplied with the Port Expansion Board for details).

2. MARK 3 PIB Board requirements (see Figure 2-5 for locations):

   a. Revision A MARK 3 PIB boards require a factory modification before the Port Expansion Board can be installed. Call POINT 4 for a Return Authorization Number.

      Any Revision B (or later) MARK 3 PIB board can be field-upgraded.

      NOTE

      Rev B or later MARK 3 PIB boards have rev letter marked on board, behind J1, following part number "PIB 21000 _".

   b. Connector at J9 - if missing, one must be installed.

Manufacturer	Part Number
POINT 4	500350
Berg	65610-150

c. DIP headers or push-on plugs to set baud rate on expansion ports.

d. Port Expansion Board, POINT 4 P/N 21100.

3. A MARK 3B PIB does not require any modifications for upgrade.

4. Software requirements:

   a. For IRIS users, this option requires version R7.5 (or later) and proper MUX driver.

2.5.2.2 Port Expansion Board Installation Procedure

To install the Port Expansion Board, perform the following steps:

1. Test the system as follows:

   a. Run Self-Test on the CPU (see Section 3.5).

   b. Run Diagnostics on all I/O devices on the PIB (see the POINT 4 MARK 3 Diagnostics Manual).

2. Remove CPU board from chassis:

   a. Install correct Microcode proms in CPU board.

   b. Make sure that all plugable devices are fully inserted.

   c. Reinstall CPU board in chassis and test CPU board.

3. Remove PIB board from chassis:

   a. Install header plug at J9, if necessary.

   b. Plug Port Expansion Board into 50-pin connector on PIB and secure with two screws (see Figure 2-8).

   c. Install baud rate jumpers at locations J10, J11, J12.

   d. Make sure that all plugable devices are fully inserted.

   e. Reinstall PIB board in chassis and attach cables (see Section 2.6).

4. Retest system as in step 1.

2.6 PIB CABLING

The MUX ports on Revisions A through C of the MARK 3 PIB board have two status inputs and no control output. MUX ports on Revision D (or later) MARK 3 PIBs, MARK 3B PIBs, and all Port Expansion Boards have one status input and one control output. The new output bit is particularly useful for modem control (DTR).

Cables for Rev A-C MARK 3 PIBs have pins 2 and 5 jumpered together. The jumper should be removed for new-style ports (Rev D (or later) and for all MARK 3B PIB and Port Expansion Board ports).

Table 2-1 is a guide for cable, port type, and device usage. Cables are for standard configuration; nonstandard devices may have different cabling requirements.

Figures 2-9 through 2-11 illustrate standard cable wiring for the MARK 3.

The PIB or Expansion Board end of the MUX cable uses a 6-pin connector:

	Molex P/N	POINT 4 P/N
Connector, 6-pin	09-50-3061	500106
Pin, crimp type (one pin per wire required)	08-50-0106	724005

The other end of the cable uses a standard RS-232 DB-25 connector.

See Appendix A for cable length considerations.

TABLE 2-1. MARK 3 CABLE USAGE

	MARK 3 PIB Rev A-C Ports 0-3	MARK 3 PIB (Rev D or later) and MARK 3B PIB Ports 0-3	All Port Expansion Ports 4-6
CRT	1 or 2	1 or 2	1 or 2
Printer	1 or 2	2	2
Modem	*	3	3

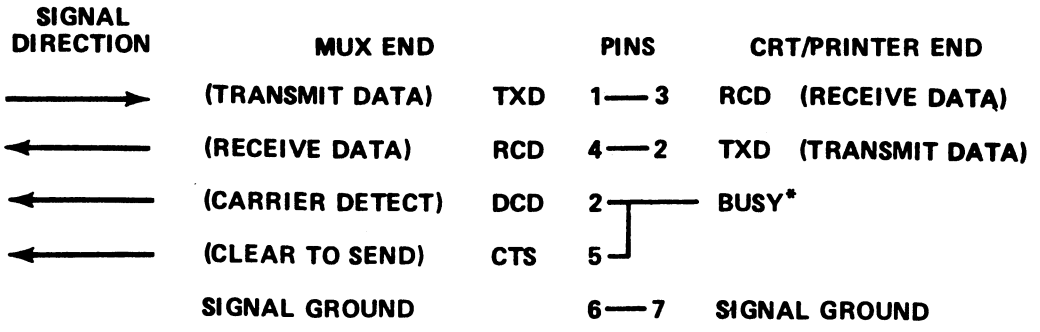
where

1 - Asynchronous CRT/printer cable for Ports 0-3 on Revision A thru C MARK 3 PIBs (see Figure 2-9).

2 - Asynchronous CRT/printer cable for Ports 0-3 on Revision D (or later) MARK 3 PIBs; MARK 3B PIBs; and Ports 4-6 on all Port Expansion Boards (see Figure 2-10).

3 - Asynchronous modem cable for Ports 0-3 on Revision D (or later) MARK 3 PIBs; MARK 3B PIBs; and Ports 4-6 on all Port Expansion Boards (see Figure 2-11).

*A modem may function on Rev A-C MARK 3 PIBs if the modem does not require a control line from the computer (no DTR required); in that case, use cable 3.



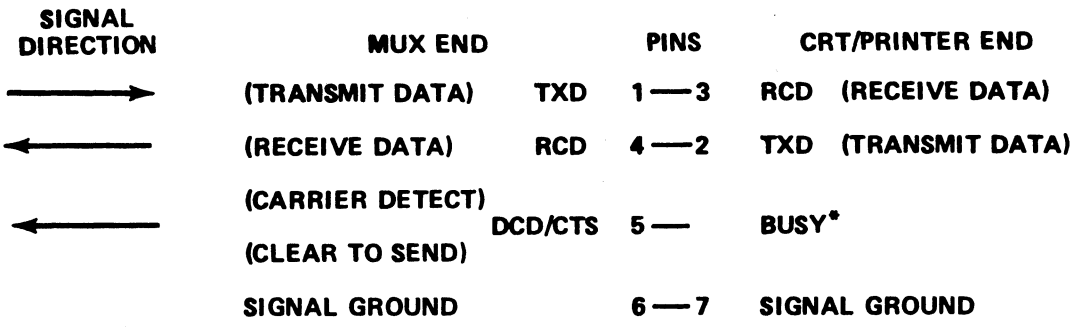
*BUSY:

CRTs – NOT REQUIRED
PRINTERS – NORMALLY PIN 19 OR 20
MAY ALSO BE 11 OR 14
(CONSULT YOUR PRINTER MANUAL)
PRINTER BUSY MUST BE:
NORMAL = HIGH
BUSY = LOW

NOTE

THIS CABLE SHOULD ALSO WORK WITH CRTs ON REV D OR LATER MARK 3 PIBs FOR ALL SEVEN PORTS; IT WILL NOT WORK WITH PRINTERS ON REV D OR LATER MARK 3 PIBs.

Figure 2-9. Asynchronous CRT/Printer Cable For Ports 0-3, MARK 3 PIB Revisions A thru C



*BUSY:
CRTs – NOT REQUIRED
PRINTERS – NORMALLY PIN 19 OR 20
MAY ALSO BE 11 OR 14
(CONSULT YOUR PRINTER MANUAL)
PRINTER BUSY MUST BE:
NORMAL = HIGH
BUSY = LOW

Figure 2-10. Asynchronous CRT/Printer Cable For Ports 0-3, MARK 3 PIB Revision D (and later) Ports 0-3, MARK 3B PIB Ports 4-6, All Port Expansion Boards

SIGNAL DIRECTION	MUX END		PINS		MODEM END
→	(TRANSMIT DATA)	TXD	1 — 2	TXD	(TRANSMIT DATA)
←	(RECEIVE DATA)	RCD	4 — 3	RCD	(RECEIVE DATA)
→	(REQUEST TO SEND) (DATA TERMINAL READY)	RTS/DTR	2 — 20	DTR	(DATA TERMINAL READY)
←	(CARRIER DETECT) (CLEAR TO SEND)	DCD/CTS	5 — 8	DCD	(CARRIER DETECT)
	SIGNAL GROUND		6 — 7		SIGNAL GROUND

NOTE

IF A MODEM IS REQUIRED ON A REV A–C MARK 3 PIB, AND EXPANSION PORTS ARE NOT AVAILABLE, THE MODEM USED MUST NOT REQUIRE ANY CONTROL LINE FROM THE COMPUTER. IT MUST BE JUMPERED OR SWITCHED TO BE READY CONSTANTLY (NO DTR REQUIRED).

Figure 2-11. Asynchronous Modem Cable For Ports 0-3, MARK 3 PIB Revision D (and later) Ports 0-3, MARK 3B PIB Ports 4-6, All Port Expansion Boards

Section 3
OPERATING PROCEDURES

3.1 INTRODUCTION

The POINT 4 MARK 3 Computer has two means of control:

- Mini-Panel on MARK 3 chassis
- Virtual Control Panel via Master Terminal

This section describes the Mini-Panel controls and indicators, and provides procedures for performing common types of operations through the Virtual Control Panel. In addition, instructions and procedures for Processor/CTU interface are included.

This section also contains instructions for diagnostic checks.

3.2 MINI-PANEL

The POINT 4 MARK 3 chassis houses essential controls and indicators for basic processor operation. The controls and indicators of the Mini-panel are located on the left-hand side of the chassis (see Figure 1-1). There are three types of operating functions on the Mini-panel: power controls and indicators, processor operation monitoring, and program execution controls. Figure 3-1 is an illustration of the Mini-panel controls and indicators.

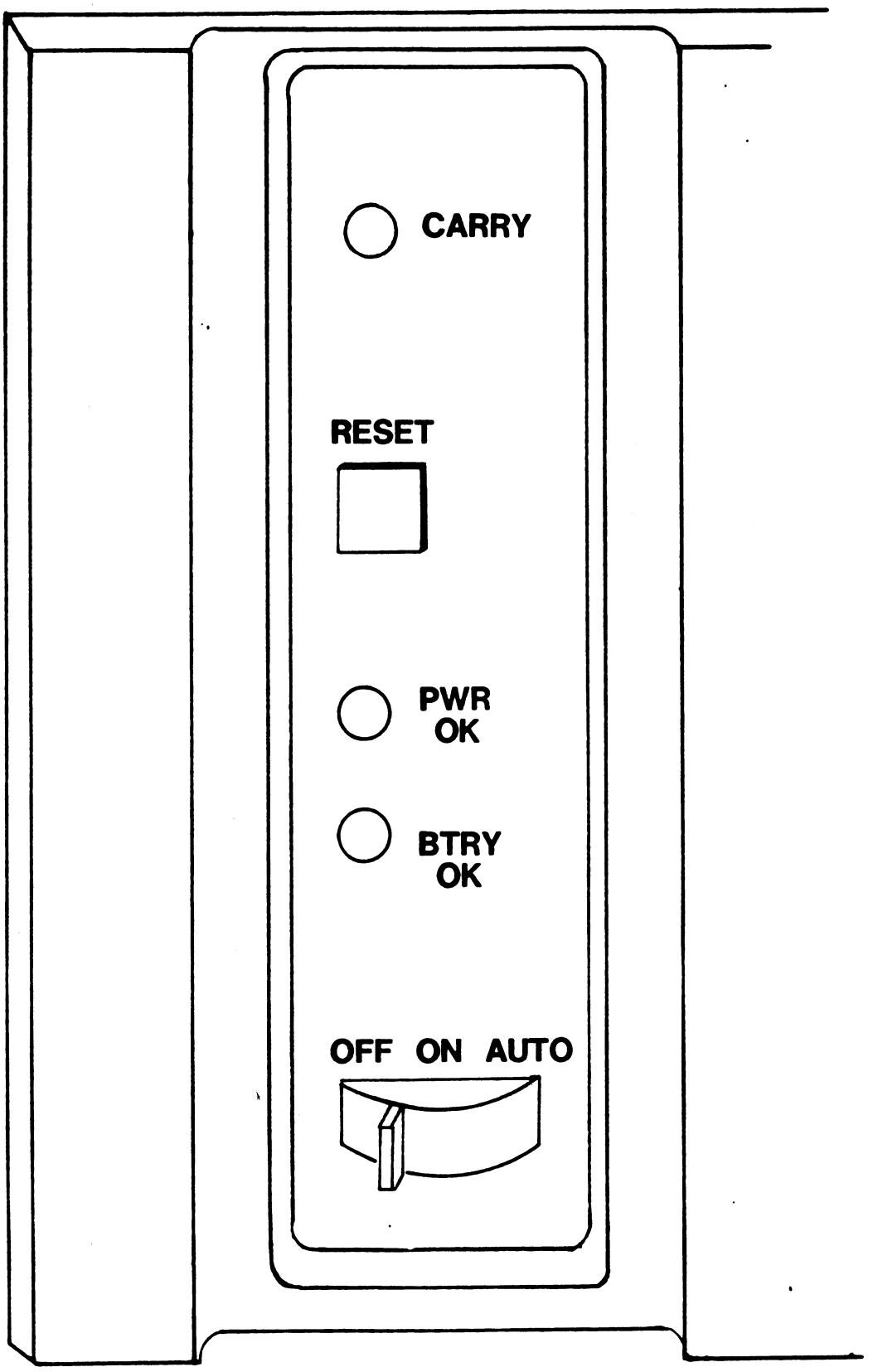


Figure 3-1. POINT 4 MARK 3 Mini-Panel

3.2.1 POWER CONTROLS AND INDICATORS (POWER SWITCH/PWR OK LIGHT)

The Mini-panel contains power controls and indicators. Power ON is controlled by a three-position switch. Table 3-1 lists the three functions of the power control switch.

A Light Emitting Diode (LED) indicator illuminates to indicate that all DC voltages are in tolerance. Table 3-2 shows the interpretations of the PWR OK LED.

See Figure 3-1 for an illustration of the three-position switch and the PWR OK indicator.

TABLE 3-1. POWER CONTROL SWITCH FUNCTIONS

Switch Setting	Function
ON	Turns on power to the processor and places the Mini-panel in the Panel-On Mode. In this mode all controls and indicators on the Mini-panel are enabled. One pass of Self-test is executed, the carry light illuminates once and goes off, and the message OK is displayed on the port 0 terminal. If a disc drive is connected and operational, an automatic IPL (initial program load) is performed when the power control switch is set to ON.
AUTO	Same as ON.
OFF	Turns off the power supply and thus all processor and Mini-panel functions.

TABLE 3-2. POWER OK LED INTERPRETATIONS

PWR OK	Interpretation
OFF	Power supply not connected to AC.
OFF	This condition (if power control switch is set to ON or AUTO position and AC plug is connected) indicates that one of the power supply voltages is out of tolerance.
ON	All power supply voltages are in tolerance and available to the processor chassis.

3.2.2 PROCESSOR OPERATION MONITORING (CARRY LIGHT)

In addition to the power monitoring indicators discussed above, the Mini-panel has an LED indicator for monitoring the carry state of processor operation. This LED is enabled in the Panel-On Mode (power control switch set to ON or AUTO) and disabled in the Panel-Off Mode (power control switch set to OFF). It indicates the current state of the processor carry flag. The LED illuminates when the carry flag is set to 1 (one). See Figure 3-1 for location of the carry indicator.

3.2.3 PROGRAM EXECUTION CONTROL (RESET SWITCH)

A pushbutton switch is available to reset program execution in the processor. This switch is enabled in the Panel-On Mode (power control switch set to ON or AUTO position) and disabled in the Panel-Off Mode (power control switch set to OFF position).

Pressing RESET loads the contents of an octal debugger/manipulator/self-test PROM into the top 1000 (octal) words of memory. The debugger/manipulator is used for access to accumulators and memory, allowing examination and deposit of data for operation monitoring and control. It optionally allows loading of system software from disc. See Section 3.3 for debugger/manipulator program commands and Section 3.5 for self-test diagnostic capabilities.

See Figure 3-1 for location of the RESET switch.

3.2.4 BATTERY MONITORING INDICATOR (BTRY OK LIGHT)

A battery monitoring indicator is present on the Mini-panel. This indicator is nonfunctional since battery backup is not available on the POINT 4 MARK 3.

3.3 VIRTUAL CONTROL PANEL

The POINT 4 MARK 3 has the ability to perform many front panel operations plus some system monitoring functions from a master terminal. This feature is designed for use by computer operators and programmers to debug system problems and to manipulate the contents of registers and memory. The feature is implemented in a stand-alone program called MANIP which is loaded into RAM from a PROM when the RESET switch is pressed or when a "HALT" instruction is executed.

3.3.1 MANIP PROGRAM

MANIP is a simple but powerful position-independent memory manipulator and debug package. MANIP occupies only 1000 (octal) words of memory.* All operations are executed by typing one letter followed by octal parameters as required (except colon (:) which is also preceded by an octal parameter) and ending with a <RETURN>.

Table 3-3 lists the functions provided by MANIP (the number in the right column indicates the number of parameters required for that particular function).

MANIP normally occupies the memory locations 77000 through 77777. Location 77000 is reserved for saving the initial value of the program counter (PC), that is, the value of PC where the CPU was executing before MANIP was started. MANIP may be moved at any time by use of its MOVE (M) instruction.

The carry light flashes while MANIP is waiting for an input character to be entered. This is a signal that MANIP is active and will respond to input.

If an error is made while entering control information, two choices are available for correcting it:

1. Press <ESC> (or any other control character except <RETURN>) to delete the type-in and enable a new type-in.

2. If the error was made in entering an octal value, type a few zeros followed by the correct octal number. MANIP will only use the last six octal digits entered for the octal word.

---

*For those who are familiar with POINT 4's IRIS Operating System, MANIP is comparable to DBUG. The main differences are that MANIP does not have (1) symbolic capability, (2) breakpoints or trace, (3) disc read or write, (4) virtual addressing, and (5) <CTRL-H>/<CTRL-A> (backspace) capability. MANIP occupies only 1000 (octal) words of memory, while DBUG occupies 3000 (octal) words of memory.

TABLE 3-3. SUMMARY OF MANIP COMMAND FUNCTIONS

Code	Function	Parameters Required
A	Display initial PC, accumulators and carry flip-flop	(0)
C[1]	Change accumulator or carry flip-flop	(2)
D	Dump (octal, word or byte)	(1)
H	Reads block 0 from 20MB or 45MB tape on 20MB or 45MB drive respectively	(0)
H46	Reads block 0 from 20MB tape on 45MB Archive drive	(0)
H111	Reads block 0 from 20MB tape on 45MB Cipher drive	(0)
F	Reads block 0 from floppy	(0)
J	Jump with accumulators and carry restored	(1)
K	Store a constant in a block of memory	(3)
M	Move a block in memory	(3)
P	Reads block 0 from disc	(0)
R	Read 3 blocks from CTU	(0)
T	Run Self-test program	(0)
:	Examine or deposit into a specified location	(1:1)

[1]On MARK 3 Computers with Rev 6A MANIP PROMs and on all MARK 2 Computers, the C command has been replaced by the extensions of the H command. If a C command is used on those computers, MANIP responds with a backslash (\).

3.3.2 MANIP COMMAND DESCRIPTIONS

A MANIP command consists of a single letter which is the command identifier and parameters which specify memory addresses and data input. All parameters must be entered in octal form. The letters x, y, and z are used to represent octal parameters. Press <RETURN> after entering any command. Table 3-4 lists each MANIP command and its definition.

TABLE 3-4. MANIP COMMANDS

Command & Parameters	Definition
A	Causes the initial value of PC (program counter) saved in first location of MANIP, the contents of accumulators A0, A1, A2, A3, and the carry flip-flop as they were at the time MANIP was entered to be typed on the master terminal.
Cx,y	Change accumulator or carry flip-flop: • If x is 0, 1, 2, or 3, then y is stored as saved value for accumulator x (A0, A1, A2, A3, respectively). • If x is 4, then saved value of the carry flip-flop is set equal to the LSB of y • Parameter Description x - 1 octal digit 0-4 y - 1 word octal
Dx	Dump memory in octal, beginning at location x. Eight words are typed per line, with the address of the first word at the beginning of each line. • Parameter Description x - octal number representing a 16-bit memory address
F	Reads block 0 from floppy disc and idles at 377 waiting to be overwritten by DMA from floppy disc (requires Floppy MANIP prom - mutually exclusive with H and R commands).

Table continues on next page.

TABLE 3-4. MANIP COMMANDS (Cont)

Command & Parameters	Definition
H	Reads block 0 from 20MB tape on 20MB drive or 45MB tape on 45MB drive and idles at 377 waiting to be overwritten by DMA from tape (requires Archive MANIP prom - mutually exclusive with F and R commands).
H46	Reads 20MB tape on 45MB Archive tape drive.
H111	Reads 20MB tape on 45MB Cipher tape drive.
Jx	Jump to location x after storing accumulators and carry values. • Parameter Description x - octal number representing 16-bit memory address
Kx,y,z	Store the octal constant z in locations x through y, inclusive. • Parameter Description x - octal number representing 16-bit beginning memory address y - octal number representing 16-bit ending memory address z - octal number representing constant
Mx,y,z	Move block in memory. Locations x through y, inclusive, are moved to area starting at location z. • Source and destination areas may overlap in either direction without bad effects. • May be used to move MANIP itself as long as destination area does not overlap source area. • Parameter Description x - octal number representing 16-bit beginning memory address y - octal number representing 16-bit ending memory address z - octal number representing 16-bit beginning memory address of new location
P	Reads block 0 from disc and idles at 377 waiting to be overwritten by DMA from disc.

TABLE 3-4. MANIP COMMANDS (Cont)

Command & Parameters	Definition
R	Reads a 600-word bootstrap (blocks 2, 3 and 4) from the cassette tape. Bootstrap will run automatically. See Section 3.4.2 for description of CTU commands in MANIP which may be used in CTU mode. (Requires CTU MANIP proms - mutually exclusive with H and F commands.)
T	Run the Self-test program. Sucessful completion results in OK being displayed on port 0 terminal. Self-test then moves itself in memory and repeats the above. Main memory will be overwritten. See the POINT 4 MARK 3 Diagnostics Manual for further details.
x:y	Octal value y is stored at location x, and next cell is opened. • Parameter Description x - octal number representing 16-bit memory address y - 1 to 6 digits representing an octal value If y is omitted, the current content of location x is displayed. A new y may then be entered, or next cell opened without change.

3.4 PROCESSOR/CTU INTERFACE

This section describes commands used to transfer stand-alone programs such as diagnostics between the CTU and the POINT 4 MARK 3. There are sixteen basic CTU commands which can be enabled from the master terminal. Access to CTU operation is through POINT 4 MARK 3's virtual control program MANIP. Entering an R command and a <RETURN> causes a CTU Boot. Once in CTU operating mode, all MANIP commands except F, H, T and R are enabled plus a subset of the CTU commands. Section 3.4.2 describes CTU commands enabled in MANIP.

Optionally, it may be desirable to access the CTU from the IRIS Operating System DBUG program. The DBUG program must be used to write to the CTU. All sixteen CTU commands are enabled under DBUG. Section 3.4.3 describes the commands enabled under DBUG.

3.4.1 CTU COMMANDS

A CTU command consists of a single-character control code (<CTRL> and an ASCII character), a block number for the starting block, an additional block count and a <RETURN>. There are sixteen functions which can be specified by control codes from an ASCII keyboard.

For ASCII Code Chart, see Appendix B.

3.4.1.1 Command Functions

The control code of a CTU command is a single nonprinting character entered while holding down the <CTRL> key on the keyboard. The CTU will echo two printable characters, a caret for the control key and the ASCII letter representing the command for ease of command verification. The CTU command functions are listed in Table 3-5.

TABLE 3-5. CTU COMMAND FUNCTIONS

Function	Control Code	CTU ASCII Echo
Read Blocks	<CTRL-R>	^R
Write Blocks	<CTRL-W>	^W
Seek Block	<CTRL-S>	^S
Enquiry	<CTRL-E>	^E
Verify Block	<CTRL-V>	^V
Write Buffer	<CTRL-B>	^B
Access Buffer	<CTRL-A>	^A
Fill Buffer	<CTRL-F>	^F
Put in Buffer	<CTRL-P>	^P
List Directory	<CTRL-D>	^D
Open File	<CTRL-O>	^O
Kill File	<CTRL-K>	^K
Rewind Drive	<CTRL-Z>	^Z
Select Track	<CTRL-T>	^T
Initialize Track	<CTRL-I>	^I
Cancel Command	<CTRL-X>	^X

3.4.1.2 Command Format

All CTU commands are structured as follows:

COMMAND [BLOCK NO.] [,ADD'L BLOCK COUNT] <RETURN>

NOTE

Fields enclosed in brackets are optional.

Field functions are described as follows:

COMMAND
This is a one-character control code specifying the function to be performed. See Subsection 3.4.1.1 for a complete listing of control codes and their functions.

[BLOCK NO.]
This is an optional DECIMAL block address specification. Zero (0) is the first block address. The maximum block address depends upon tape length (typically 999).

NOTE

CTU blocks contain only 128 words; IRIS Operating System blocks contain 256 words.

[ADDITIONAL BLOCK COUNT]
This is an optional DECIMAL field which specifies the number of blocks to be operated upon in addition to the block specified in the [BLOCK NO.] field. A specification of zero (0) for this field instructs the CTU to operate only on the block specified in the [BLOCK NO.] field. The maximum count is 255.

<RETURN>
An ASCII carriage return character is the execute instruction for the CTU. If the CTU receives a <CTRL-X> before a <RETURN>, the CTU cancels (does not execute) the preceding command string specified. A new command can follow the <RETURN>.

3.4.1.3 CTU Error Conditions

The CTU reports error conditions by presentation of an error code. An error condition is given in the following format:

BELL, Error Code, BELL, <RETURN>, Line Feed

The error codes and their descriptions are shown in Table 3-6.

TABLE 3-6. CTU ERROR CODES

Error	Description
P	A write or erase operation was attempted on a write protected tape.
M	Tape motion failure. This error occurs either as a result of a jam or mechanical malfunction, or as a result of incorrect tape positioning due to operator handling. In the case of incorrect tape positioning, the CTU does not know the tape location and thus runs into the stops.
R	Read error. The operator should retry the command. An excessive number of read errors usually indicates noise interference, faulty system ground, a defective tape or a CTU hardware malfunction.
U	Unknown name. An attempt was made to delete a filename not found in the directory.
?	Syntax error in command string.
F	Track Directory is full (126) names. An old filename must be killed before a new filename may be entered.

3.4.2 CTU COMMANDS ENABLED IN MANIP

A group of nine CTU commands may be used from the Virtual Control Panel (MANIP) mode. These commands allow limited access to CTU files. A more complete set of CTU commands is enabled from the IRIS Operating System DBUG program.

To enter CTU mode from MANIP use the MANIP R command to read blocks 2, 3 and 4 from the CTU tape. This overlays a portion of MANIP with the CTU control commands.

All MANIP CTU commands consist of a control character <CTRL> and an ASCII character, followed optionally by one or more parameters, and terminated by a <RETURN>. The only exception is <CTRL-X> which cancels any partially entered command immediately. Data is stored on tape in blocks of 256 bytes (128 words) each. Table 3-7 lists the CTU commands enabled in MANIP after MANIP is overlaid. All numeric parameters (x,y) are in DECIMAL, origin 0.

The Read CTU command transfers data into memory starting at address 0. To start the transfer at some other address, precede the CTU command with:

```
Memory address (octal) : <RETURN>
```

MANIP will then display the content of the chosen location, followed by a colon. This allows examination of the word before starting the tape transfer. To proceed, type the <CTRL-R>, followed by its parameters (if any) and a <RETURN>.

See Section 3.4.4 for CTU operating procedures.

TABLE 3-7. CTU COMMANDS IN MANIP

Control Character/ Parameters	Description
<CTRL-D>	List directory (index) from tape, if tape is so formatted.
<CTRL-E>	Enquire (error status).
<CTRL-O>file	Open the named file, if it is in the directory.
<CTRL-O>file,x,y	Create a directory entry for the named file starting at block x and containing y+1 blocks of 128 words each.
<CTRL-R>	Read the open file from tape into memory.
<CTRL-R>x,y	Read from tape into memory; read y+1 blocks starting at block x.
<CTRL-S>x	Seek to block x on tape. <CTRL-S>999 will wind the tape all the way forward.
<CTRL-T>n	Select track n (0 or 1).
<CTRL-X>	Cancel partially entered command.
<CTRL-Z>	Rewind tape to starting position.
<CTRL-K>file	Kill the named file, i.e., erase its name from the directory.

NOTE

<ESC> exits CTU mode and reverts to normal MANIP commands, but does not cancel any partial command that may already have been transmitted to the CTU. Use <CTRL-X> to cancel a partial command.

MANIP allows reading from CTU into RAM (CPU main memory), but does not allow writing onto tape. To write to the CTU, obtain a cassette which contains DBUG, read it into memory by means of the MANIP <CTRL-R> command, then Jump into DBUG and use its CTU write capabilities (see Section 3.4.3).

3.4.3 CTU COMMANDS ENABLED IN DBUG

DBUG is a position-independent debug package of the IRIS Operating System. When using the Virtual Control Panel on the POINT 4 MARK 3 it is necessary to use CTU commands enabled in DBUG to write to the CTU. The write procedure from MANIP requires a cassette containing DBUG which must be read into memory using MANIP. A jump into DBUG allows use of DBUG CTU commands to write to the CTU. CTU commands enabled in DBUG may also be used in other CTU transfer procedures.

All CTU commands consist of a control character, followed optionally by one or more parameters, and terminated by a <RETURN>. The only exception is <CTRL-X> which cancels any partially entered command immediately. Data is stored on tape in blocks of 256 bytes (128 words) each. Table 3-8 lists the CTU commands enabled in DBUG. All numeric parameters (x,y) are in DECIMAL, origin 0.

TABLE 3-8. CTU COMMANDS ENABLED IN DBUG

Control Character/ Parameters	Description
<CTRL-A>x,y	Access CTU buffer, i.e., transfer buffer into memory. Transfers y bytes, starting at byte x. Default = 256 bytes starting at byte 0.
<CTRL-B>x	Write CTU buffer onto tape, at block x.
<CTRL-D>	List directory (index) from tape, if tape is so formatted.
<CTRL-E>	Enquire (error status).
<CTRL-F>	Fill CTU buffer from memory (128 words).
<CTRL-I>x	Initialize (format) selected track to x+1 blocks of 128 words each. Maximum = 999 for 1000 blocks.
<CTRL-K>file	Kill the named file; i.e., erase its name from the directory.
<CTRL-O>file	Open the named file, if it is in the directory.

TABLE 3-8. CTU COMMANDS ENABLED IN DBUG (Cont)

Control Character/ Parameters	Description
<CTRL-O>file,x,y	Create a directory entry for the named file (max. 5 char.), starting at block x and containing y+1 blocks of 128 words each.
<CTRL-P>x,y	Put into CTU buffer from memory, transferring y bytes beginning at byte x in the buffer. Default = 256 bytes starting at byte 0.
<CTRL-R>	Read the open file from tape into memory.
<CTRL-R>x,y	Read from tape into memory; read y+1 blocks starting at block x.
<CTRL-S>x	Seek to block x on tape.
<CTRL-T>n	Select track n (0 or 1).
<CTRL-V>	Verify; i.e., read from tape into CTU buffer, checking checksum.
<CTRL-W>	Write from memory onto tape into the open file, if any.
<CTRL-W>x,y	Write from memory onto tape, writing y+1 blocks starting at block x.
<CTRL-X>	Cancel partially entered command.
<CTRL-Z>	Rewind tape to starting position.

NOTE

<ESC> exits CTU mode and reverts to normal DBUG commands, but does not cancel any partial command that may already have been transmitted to the CTU. Use <CTRL-X> to cancel a partial command.

All commands that transfer data into or out of main memory default to an initial address of 0. To start the transfer at some other address, precede the CTU command with:

```
Memory address (octal) : <RETURN>
```

DBUG will display the content of the chosen location, followed by a colon. This allows examination of the word before starting the tape transfer. To proceed, type the CTU control character (e.g., <CTRL-R> or <CTRL-W>), followed by its parameters and a <RETURN>.

Table 3-9 is a quick-reference guide to the DBUG CTU commands used for data transfer from a source to a destination.

TABLE 3-9. SUMMARY OF DATA TRANSFER COMMANDS

Source	Destination	Command
Tape	Memory	<CTRL-R>
Memory	Tape	<CTRL-W>
Tape	Buffer	<CTRL-V>
Buffer	Tape	<CTRL-B>
Buffer	Memory	<CTRL-A>
Memory	Buffer	<CTRL-F> complete buffer <CTRL-P> selected byte(s) only

3.4.4 CTU OPERATING PROCEDURES

The POINT 4 MARK 3 provides for CTU handling through the Virtual Control Panel program MANIP. Place the Boot Cassette in the CTU and press RESET. Use MANIP's "R" command and a <RETURN> to load a secondary boot routine from the CTU. This routine overlays MANIP from 77000 through 77577 (600 words). The boot occupies blocks 2, 3 and 4 (3 blocks) of the tape.

The Boot Cassette, when loaded as described above, may be used to write any portion of core to cassette. For example

```
20: <RETURN> <CTRL-W>100,50
```

will write block 100 (decimal) plus 50 (decimal) additional blocks starting at location 20 (octal) for a total of 51 CTU blocks.

NOTE

One CTU block contains 256 bytes (128 words). Therefore, one disc block (512 bytes) is equal to two CTU blocks.

For detailed instructions on use of standard CTU utility programs and on transferring files from the POINT 4 MARK 5 to the POINT 4 MARK 3, see the CTUTILITY documentation available from POINT 4 Data Corporation.

3.5 DIAGNOSTIC CHECKS

3.5.1 DIAGNOSTIC CAPABILITIES

The POINT 4 MARK 3 has a comprehensive built-in diagnostic program, contained in a PROM (Programmable Read-Only Memory).

The Self-Test diagnostic contains the following tests:

1. Compare Instruction Test
2. ALU and Data Bus Test
3. ALU Source Operand Test
4. Exhaustive ALU Instruction Test
5. Page 0 and Base 3 Addressing Modes Test
6. Relative, Base 2, and Indirect Addressing Modes Test
7. Limited I/O Instruction Test
8. Worst-Case Memory Test of all Memory Locations

3.5.2 SELF-TEST OPERATING PROCEDURES

General procedures for initiating the POINT 4 MARK 3 Self-Test follow. For details on tests executed, expected results, and error interpretation, see the POINT 4 MARK 3 Diagnostics Manual.

A T (Test) Command from the Virtual Control Panel, followed by a <RETURN> initiates the MARK 3 Self-Test program (see Section 3.3.2). Successful completion results in OK displayed on the Master Terminal. Upon completion of the Self-Test procedure, Self-Test will randomly move itself in memory and begin the procedure again. Thereafter, each time the Self-Test moves itself in memory and repeats, an OK is displayed. The Self-Test program continues re-execution until RESET is pressed.

3.5.3 SELF-TEST ERRORS

If Self-Test encounters an error, it types out the PC address and contents of the accumulators when the error occurred.

If no terminal is being used, the carry light provides information about Self-Test. An irregularly flashing carry light indicates Self-Test is running. A failed Self-Test is indicated by a regularly flashing pattern, or a steady on or steady off condition of the carry light.

Refer to the POINT 4 MARK 3 Diagnostics Manual for detailed information on diagnostic programs, program listings, and error interpretation.

Section 4
INPUT/OUTPUT INTERFACES

4.1 INPUT/OUTPUT BUS

Input/output bus signals connect the processor logic to peripheral device logic. These signals, used in both programming I/O transfers and data channel transfers, interface the processor Main Data Bus to peripheral device controllers. Logic to implement I/O transfer instructions is present in all device controllers. Data channel transfer logic is present only in those controllers that control devices using the data channel. Device-end control logic for these functions may vary widely, depending on the requirements of the particular device. This section describes the POINT 4 MARK 3 I/O bus and control signals.

4.1.1 INPUT/OUTPUT INTERFACE SIGNALS

Signals on the input/output bus can be grouped into the following signal classifications:

- Bidirectional Data Bus (16 lines): Used for transfer of all data and address words between the CPU and a peripheral device, for both programmed I/O and data channel transfers.
- Device Codes (6 lines): Codes used to designate the peripheral device involved in an input/output instruction.
- Device Control Signals (2 lines): Signals generated by the CPU in response to input/output instructions; used to initialize and control I/O devices.
- Interrupt Input Signals (3 lines): Signals used to designate which controller will be serviced.
- Disc Status Flags (5 lines): Signals used to carry disc status information to the processor.

Figure 4-1 is a diagram of I/O signals across the I/O bus. Table 4-1 lists these signals by classification (group) and signal name, indicates direction, and describes each signal function.

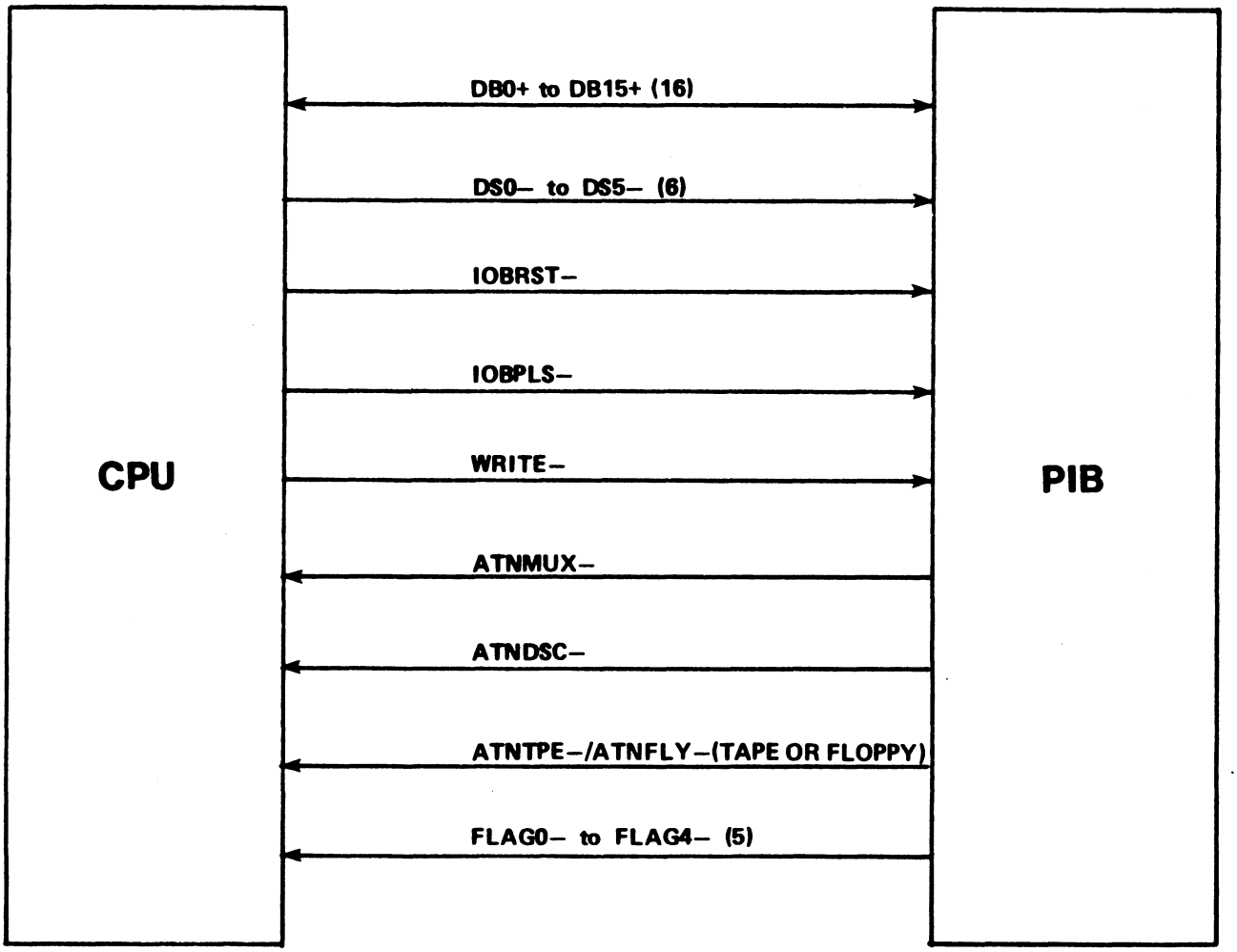


Figure 4-1. Input/Output Signals

TABLE 4-1. INPUT/OUTPUT SIGNALS

Signal Group	Signal*	Direction	Description
Data Bus	DB0+ to DB15+	Bidirectional	All data and addresses are written from CPU to PIB or read from PIB to CPU via these lines. DB0+ is the MSB.
Device Code	DS0- to DS5-	From CPU	Processor-generated signals specifying device code (bits 10-15 of I/O instruction) to PIB. The PIB interprets DS0, DS1 and DS2 as device codes and DS3, DS4 and DS5 as function codes to condition the specified device. DS0- is the MSB.
Device Control Signals	IOBRST-	From CPU	SYSTEM RESET: Processor or Mini-panel-generated signal used to reset all interfaces to peripheral devices on the system.
	IOBPLS-	From CPU	PULSE: Processor-generated signal used as a strobe to load registers during an output transfer.
	WRITE-	From CPU	WRITE/READ: Generated by processor to indicate direction of transfer. A negative signal indicates current I/O cycle is in Write mode; a positive signal indicates current cycle is in Read mode.
Interrupt Input Signals	ATNMUX-	From Device	ATTENTION: Used by multiplexer to signal need for interrupt servicing.
	ATNDSC-	From Device	ATTENTION: Used by disc controller to signal need for interrupt servicing.
	ATNTPE-	From Device	ATTENTION: Used by tape controller to signal need for interrupt servicing (mutually exclusive with ATNFLY-).
	ATNFLY-	From Device	ATTENTION: Used by floppy disc controller to signal need for interrupt servicing (mutually exclusive with ATNTPE-).
Disc Status Flags	FLAG0- to FLAG4-	From Device	FLAGS: Used by firmware to transfer status information: FLAG0 - End (floppy) FLAG1 - Index detected FLAG2 - Sector detected FLAG3 - Full or empty FLAG4 - Gap

*Signal names ending with "+" are active high; those ending with "-" are active low.

4.1.2 BACKPLANE PIN SIGNAL CONNECTORS

All signal connections between the processor and the PIB board take place via 100-pin backplane connectors. Figure 4-2 shows the connector pin layout for all I/O signals. The labelled pins refer to the I/O control signals, data transfer signals and the power lines used by the CPU and PIB boards.

BOTTOM			TOP
GND	2	1	GND
GND	4	3	GND
+5V	6	5	+5V
+5V	8	7	+5V
DS1–	10	9	DS2–
DS5–	12	11	DS0–
DS3–	14	13	DS4–
DB15+	16	15	WRITE–
DB13+	18	17	DB14+
DB11+	20	19	DB12+
DB9+	22	21	DB10+
DB7+	24	23	DB8+
DB5+	26	25	DB6+
DB3+	28	27	DB4+
DB1+	30	29	DB2+
GND	32	31	DB0+
IOBRST–	34	33	GND
GND	36	35	GND
GND	38	37	IOBPLS–
FLAG4–	40	39	GND
FLAG2–	42	41	FLAG3–
FLAG0–	44	43	FLAG1–
ATNMUX–	46	45	ATNDSC–
GND	48	47	ATNTPE–/ATNFLY–
	50	49	GND
	52	51	
	54	53	
GND	56	55	
AUTO–	58	57	GND
GND	60	59	CL–
PWRGON–	62	61	GND
GND	64	63	PWRF–
XRESET–	66	65	GND
–5V	68	67	XRESET+
–5V	70	69	–5V
GND	72	71	GND
–12V	74	73	–12V
GND	76	75	–12V
+12V	78	77	GND
GND	80	79	+12V
+12VBU	82	81	GND
GND	84	83	+12VBU
–5VBU	86	85	GND
GND	88	87	–5VBU
+5VBU	90	89	GND
+5VBU	92	91	+5VBU
GND	94	93	GND
+5V	96	95	+5V
+5V	98	97	+5V
FRM GND	100	99	FRM GND

Figure 4-2. Backplane I/O Signals

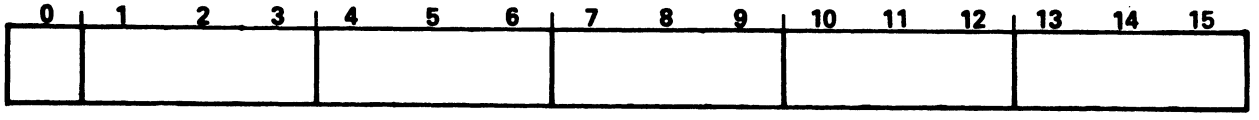
Section 5
STANDARD INSTRUCTION SET

5.1 INTRODUCTION

This section explains the function and use of POINT 4 MARK 3 instructions. Included is a discussion of two's-complement notation, addressing modes, and the individual instructions in the memory reference, arithmetic/logical, and input/output instruction groups. Input/output instructions and interrupt handling instructions are presented, with details given for special code-77 (CPU) instructions.

5.2 OCTAL REPRESENTATION AND TWO'S COMPLEMENT NOTATION

The computer uses 16-bit binary words for program instructions and data. The bits are numbered 0 through 15 with bit 0 the most significant bit (MSB) and bit 15 the least significant bit (LSB). For convenience, binary words are represented in 6-digit octal form. Each octal digit represents three bits and can have values between 0 and 7, except the most significant digit which represents a single bit and has a maximum value of 1. The POINT 4 MARK 3 16-bit binary word format is shown in Figure 5-1.



082-23

Figure 5-1. POINT 4 MARK 3 16-bit Binary Word Format

The reader is presumed to be familiar with binary and octal notations. For a simple review, the following example shows the correspondence between decimal, binary and octal representation:

Decimal	Binary	Octal	
0	0000000000000000	000000	
1	0000000000000001	000001	
2	0000000000000010	000002	
8	0000000000001000	000010	
64	0000000001000000	000100	
5407	0001010100011111	012437	
32,767	0111111111111111	077777	(15 bit max.)
65,535	1111111111111111	177777	(16 bit max.)

The computer represents negative numbers in two's-complement form. Signed positive and negative numbers are used both as 16-bit operands and as 8-bit address displacements in memory reference instructions. A review of two's complement arithmetic follows.

In two's-complement arithmetic, positive and negative values are distinguished by a 0 or 1 in the leftmost bit position (sign bit). Positive numbers have a sign bit of 0, with the numerical value expressed in ordinary binary form by the remaining bits. Negative numbers have a sign bit value of 1 and the numerical value expressed in two's-complement form. The two's complement is found by taking the one's complement or logical complement of the number including the sign bit (changing all 0's to 1's and all 1's to 0's) and adding 1.

The number zero is represented by 0's in all bit positions. There is only one representation for zero, since the two's complement of zero is also zero. Zero is a nonnegative value. For this reason also, there is one more negative number than there are nonzero positive numbers.

The range of signed, 8-bit fields is as follows:

	Binary Representation	Octal Value
Largest positive	01 111 111	+177
	01 111 110	+176
	.	
	.	
	.	
	00 000 001	+1
	00 000 000	0
	11 111 111	-1
	11 111 110	-2
	.	
	.	
	.	
	10 000 001	-177
Most negative	10 000 000	-200

5.3 INSTRUCTION TYPES

From the programmer's point of view, the POINT 4 MARK 3 Computer is comprised of four accumulators, 64K bytes of memory and an input/output bus. The instructions control and manipulate the data flowing between these elements.

Instruction words can be classified into one of the following three categories:

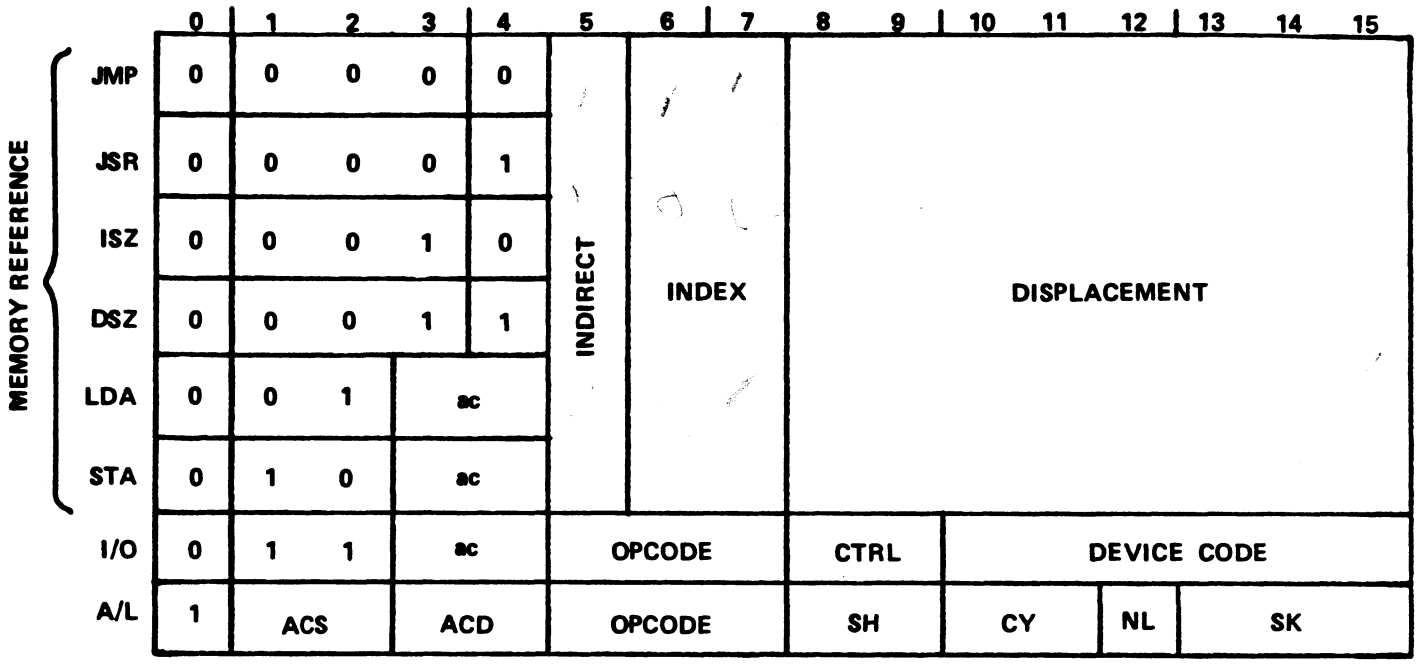
1. Memory Reference Instructions are instructions that reference a memory location. These include:

   LDA - Load an accumulator from memory
   STA - Store an accumulator into memory
   JMP - Jump to another location in memory
   JSR - Jump to a subroutine in memory
   ISZ - Increment memory and skip if zero
   DSZ - Decrement memory and skip if zero

2. Arithmetic/Logic Instructions are instructions that specify a particular arithmetic or logical operation to be performed on one or two operands stored in the accumulators, and allow for testing the result for skip conditions.

3. Input/Output Instructions are instructions for input/output operations to a specific peripheral device.

Figure 5-2 is an overview of the formats for each type of instruction. Each of these three classes is discussed in detail in the succeeding subsections.



ac = ACCUMULATOR
CTRL = CONTROL PULSE
ACS = SOURCE ACCUMULATOR
ACD = DESTINATION ACCUMULATOR
SH = SHIFT CONTROL
CY = CARRY PRESELECTION
NL = NO-LOAD
SK = SKIP CONDITION

Figure 5-2. POINT 4 MARK 3 Instruction Format Summary

5.4 MEMORY REFERENCE INSTRUCTIONS

Six memory reference instructions are used to move data between memory locations and accumulators, to transfer program control to a new location, and to modify and test memory words. The memory reference instructions fall into three general categories, as follows:

1. Move Data Instructions: LDA, STA
2. Jump Instructions: JMP, JSR
3. Modify Memory Instructions: ISZ, DSZ

Before describing the function of each instruction in this group it is necessary to describe the way in which they address memory.

5.4.1 MEMORY ADDRESSING

Each memory reference instruction uses one of several addressing modes to determine an effective memory address, E. The processor accesses the location specified by the effective memory address and uses the contents as the operand of the instruction.

The Jump instructions (JMP, JSR) and the Modify Memory instructions (ISZ, DSZ) both use the binary format shown in Figure 5-3.

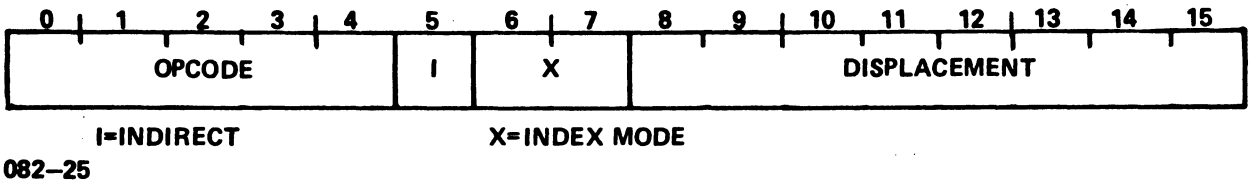


Figure 5-3. Jump and Modify Memory Instruction Binary Word Format

Bits 0-4 of the instruction word are the OPCODE field. Bit 5 is the indirect or I field; bits 6 and 7 are the Index Mode or X field; and bits 8-15 are the displacement or D field.

The Move Data instructions (LDA, STA) use the binary format shown in Figure 5-4.

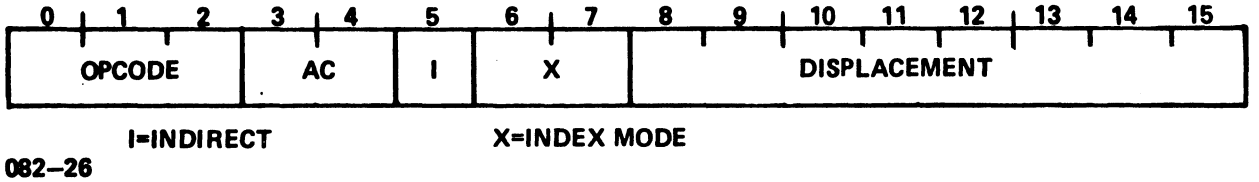


Figure 5-4. Move Data Instruction Binary Word Format

Table 5-1 defines the memory reference instructions, indicating bits, their fields and field definitions.

All addresses, both direct and indirect, are entered into the Effective Address Register. When this register contains the effective address E, the instruction specified in bits 0 through 4 of the command is executed.

TABLE 5-1. MEMORY REFERENCE INSTRUCTIONS

Bits	Field	Definition
0-2	OPCODE	Determines which instruction is performed. For the Move Data instructions (LDA, STA), bits 0, 1 and 2 make up the OPCODE field. For the Jump instructions (JMP, JSR), and Modify Memory instructions (ISZ, DSZ), bits 0-4 make up the OPCODE field.
3,4	Accumulator (ac)	The ac field for the Move Data instructions (LDA, STA) specifies one of the four general accumulators. The specified accumulator will either have data stored in it or data transferred from it.
5	Indirect (I)	Determines whether the X and D fields specify the effective address (E) directly or whether indirect addressing is to be used.
6,7	Index Mode (X)	Defines one of the four addressing modes. Each addressing mode may be thought of as a page of 256 words which the instruction can address directly.
8-15	Displacement (D)	Specifies the word addressed on the selected page.

5.4.1.1 Indexing Mode

The X field selects one of the indexing modes shown in Table 5-2.

TABLE 5-2. INDEXING MODES

Bits 6-7	Definition
00	Page Zero - Page zero is defined as the first 256 memory locations (addresses in the range from 000000 to 000377 octal). The effective memory address in page zero addressing is equal to the value of the D field which is an unsigned binary integer that can have values from 000 octal to 377 octal.
01	Relative Addressing - In the relative addressing mode the address placed in the Effective Address Register is equal to the address in the Program Counter (PC) plus the value of the displacement in the D field. In this case the displacement D is a signed binary integer. Bit 8 is the sign (0 = positive; 1 = negative) and the integer may have any value in the range from -200 to +177 octal (decimal -128 to +127). The address in PC can be visualized as the center of a 256-word page and any address between the bottom (128 words below the PC) and top (127 words above PC) of the page can be specified by the displacement D.
10 or 11	Base Register Addressing - In the base register addressing mode the address placed in the Effective Address Register is equal to the address in accumulator register A2 (code 10) or A3 (code 11) plus the value of the displacement in the D field. In this case the displacement D is a signed binary integer. Bit 8 is the sign (0 = positive; 1 = negative) and the integer may have any value in the range from -200 octal to +177 octal (decimal -128 to +127). The address in A2 or A3 can be visualized as the center of a 256-word page and any address between the bottom (128 words below A2 or A3) and top (127 words above A2 or A3) of the page can be specified by the displacement D.

5.4.1.2 Indirect Addressing Operations

When the I field (bit 5) of the Memory Reference Instruction contains a 1, an indirect addressing sequence is required. In this case, the address in the Effective Address Register (determined by the X and D fields) is the memory address from which a second address word is to be fetched.

5.4.2 TYPES OF MEMORY REFERENCE INSTRUCTIONS

When the Effective Address Register contains the effective address E, one of two groups of memory reference instructions is performed as determined by the operation codes. Refer to Section 5.4.1 for basic memory reference instruction formats and field definitions. Refer to Appendix C (Von Neumann Map of the POINT 4 MARK 3 Command Structure) for octal formats of each instruction, and to Appendix D (POINT 4 MARK 3 Instruction Reference Chart) for octal-to-symbolic conversion of memory reference instructions.

5.4.2.1 Move Data Instructions

When the code in bits 1 and 2 of the OPCODE field is not 00, and the effective address (E) is in the Effective Address Register, two operations are performed, as shown in Table 5-3.

TABLE 5-3. MOVE DATA INSTRUCTIONS

Bits 1-2	OPCODE	Definition
01	LDA	Load Accumulator Instruction - The contents of memory location E are stored in the accumulator specified by the ac field (bits 3 and 4). The contents of E are unaffected; the original contents of the accumulator are lost.
10	STA	Store Accumulator Instruction - The data in the accumulator specified by the ac field is transferred to memory location E. The contents of the accumulator are unaffected; the original contents of E are lost.

5.4.2.2 Jump and Modify Memory Instructions

When bits 0, 1 and 2 are all zero, and the effective address (E) is in the Effective Address Register, one of four operations is performed. The operation is specified by the code in bits 3 and 4 of the OPCODE extension field. These jump and modify memory instructions are shown in Table 5-4.

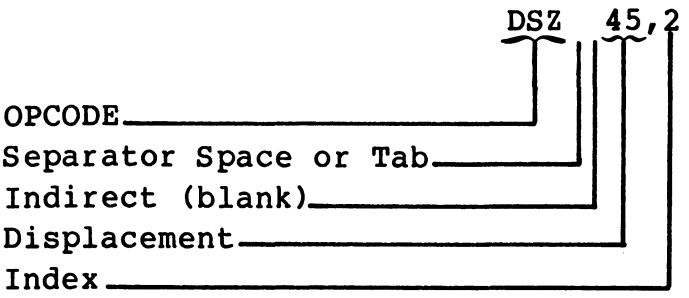
TABLE 5-4. JUMP AND MODIFY MEMORY INSTRUCTIONS

Bits 1-4	OPCODE	Definition
0000	JMP	Jump Instruction - The effective address E is transferred from the Effective Address Register to the Program Counter (PC). The next instruction is then fetched from jump address E and sequential execution is continued from there.
0001	JSR	Jump to Subroutine Instruction - After the effective address E has been calculated the address in PC is incremented and the incremented value is stored in accumulator A3. Then the effective address E is transferred from the Effective Address Register to the Program Counter (PC). The next instruction is then fetched from jump address E. Execution of another JMP or JSR instruction that specifies A3 will cause the program to return to the address in A3 plus or minus any desired displacement D.
0010	ISZ	Increment and Skip if Zero - The contents of effective address E are fetched, incremented and written back into address E. If the incremented value is equal to zero, PC is incremented by one to skip the next instruction.
0011	DSZ	Decrement and Skip if Zero - The contents of the location specified by effective address E are decremented and written back into address E. If the decremented value is equal to zero, PC is incremented by one to skip the next instruction.

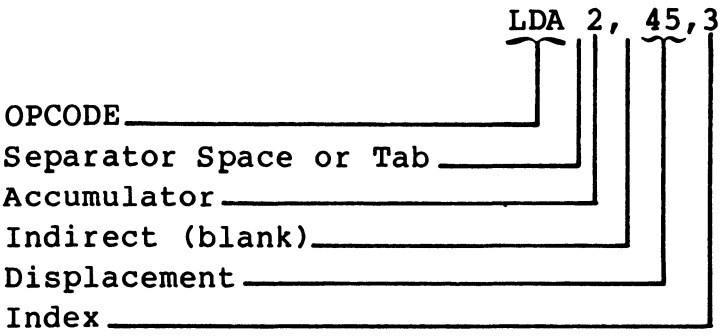
5.4.2.3 Assembler Language Conventions and Addressing Examples

The assembler language memory reference instruction consists of the instruction OPCODE mnemonic (STA, LDA, JMP, etc.) followed by symbols that specify the accumulator, the addressing mode and the memory address. The assembler program translates these statements into binary code which the processor executes. Table 5-5 shows the programming conventions for memory reference instructions.

The format for modify memory and jump instructions requires the instruction mnemonic and a memory address, including indirect addressing displacement, and indexing indicator. The assembly language instruction will be formatted as follows:



The move data instructions LDA and STA also require that an accumulator (A0-A3) be specified. For example



Fields that are not specified will be assembled containing 0s. An "@" symbol denotes indirect addressing and places a 1 in bit 5 of the instruction. An example of indirect page-zero (X Field = 00) addressing is as follows:

```
LDA 1,@20
```

Relative addressing is formatted as follows:

```
LDA 0,. +15
```

The symbol "." indicates X = 01 (relative addressing) and thus "." represents the current value of the program counter.

TABLE 5-5. ASSEMBLER LANGUAGE CONVENTIONS FOR MEMORY REFERENCE INSTRUCTIONS

Instruction Function	OPCODE Mnemonic	Separator Space or Tab	Accumulator Number	,	Memory Address			
					Indirect	Displcmt	,	Index
Load Accumulator	LDA		ac*	,	blank or @	Displacement	,	blank or 1** or 2 or 3
Store Accumulator	STA							
Jump	JMP		none					
Jump Subroutine	JSR							
Increment and Skip if Zero	ISZ							
Decrement and Skip if Zero	DSZ							

*ac = 0, 1, 2, 3 representing A0, A1, A2, A3

**Instead of "displacement,1" the following sequence may be used: ".±displacement"

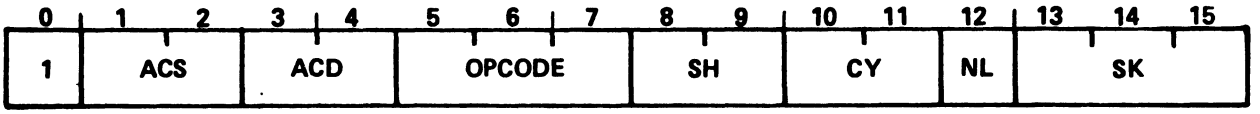
5.5 ARITHMETIC AND LOGICAL INSTRUCTION GROUP

The eight arithmetic/logical instructions perform binary addition subtraction and logical functions on 16-bit operands. These instructions are:

- Arithmetic: ADD, ADC, INC, SUB, NEG
- Logical: MOV, COM, AND

All Arithmetic and Logic instructions contain a 1 in bit 0 and have their basic Arithmetic/Logical Unit (ALU) function specified by bits 5-7 as shown in Figure 5-5. The fields of the Arithmetic/Logical instruction format are as follows:

- Source Accumulator (ACS)
- Destination Accumulator (ACD)
- OPCODE
- Shifter/Swapper (SH)
- Carry Preselect (CY)
- No-Load (NL)
- Skip Condition Tester (SK)



082-27

Figure 5-5. Arithmetic/Logical Instruction Format

5.5.1 ARITHMETIC AND LOGICAL PROCESSING

The organization of the arithmetic/logical processing unit must be described before discussion of the eight arithmetic and logical instructions and their auxiliary control fields. Subsystem organization is shown in Figure 5-6, and described in the following subsections.

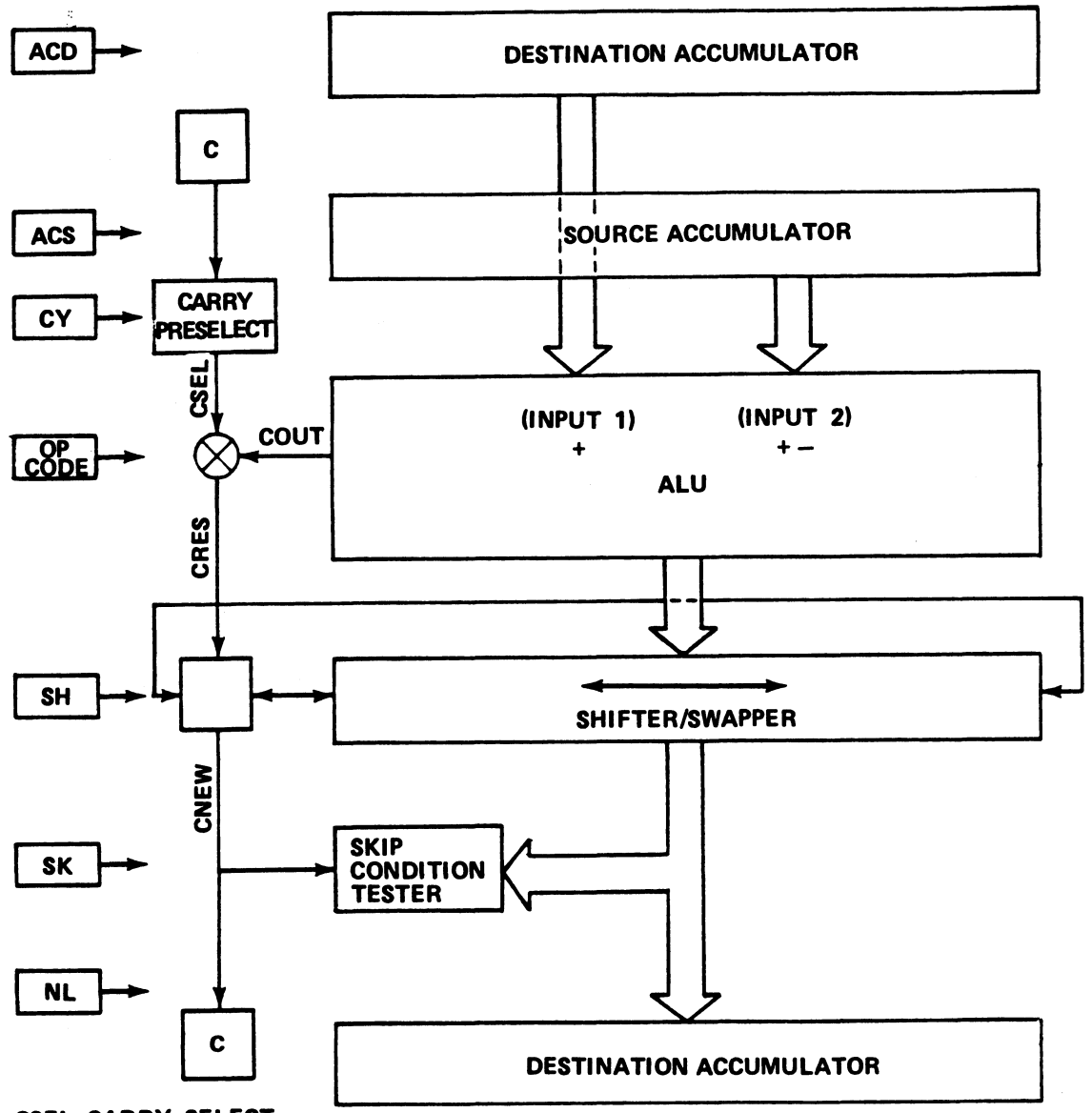


Figure 5-6. Arithmetic/Logical Operations

5.5.1.1 Arithmetic/Logical Operations

The heart of the subsystem is the Arithmetic/Logic Unit (ALU) which performs the actual addition, subtraction or logical operation. It has provision for two inputs:

Input 1: Comes from the accumulator selected by the ACD field and is used only in the operations which require two operands (ADD, SUB, ADC and AND).

Input 2: Comes from the accumulator selected by the ACS field and is used in all operations.

The ALU performs the arithmetic or logical operation specified by the OPCODE field (bits 5-7). The result of this operation may cause a carry-out to occur from the most significant bit of the ALU. In the case of an operation which adds unsigned integers, a carry-out is equivalent to overflow; however this is not always true. See Section 5.5.1.2 for a more complete discussion of carry and overflow operation.

If the result of the arithmetic or logical operation involves a carry-out (COUT) the carry preselected by the CY field of the instruction (CSEL) is complemented. The resulting carry (CRES) together with the 16-bit operation result generated by the ALU is applied as a 17-bit operand to the Shifter where a shift-left, shift-right or swap may occur as determined by the SH field of the instruction. After shifting, the carry (CNEW) and the 16-bit operation result are loaded into the Carry Flag (C) and the destination accumulator (ACD) unless this is prevented by a 1 in the No-Load (NL) field. In either case they are tested for a skip condition (i.e., to determine if the next instruction should be skipped) as specified in the SK field of the instruction.

5.5.1.2 Overflow and Carry-Out Operations

The 16-bit numbers processed by the ALU may be thought of as unsigned integers between 0 and 64K or as signed integers between -32K and +32K.

Binary Number	Unsigned Interpretation		Signed Interpretation	
(in ALU)	Octal	Decimal	Octal	Decimal
1111111111111111	177777	64K-1	-00001	-1
1111111111111110	177776	64K-2	-00002	-2
.				
.				
.				
1000000000000001	100001	32K+1	-77777	-32K+1
1000000000000000	100000	32K	-100000	-32K
0111111111111111	077777	32K-1	+77777	32K-1
0111111111111110	077776	32K-2	+77776	32K-2
.				
.				
.				
0000000000000001	000001	1	+00001	1
0000000000000000	000000	0	00000	0

When working with either interpretation there is the possibility of an overflow (answer greater than the maximum number that can be represented) or underflow (less than the minimum). In general, the ALU will produce the correct result if no overflow or underflow occurs, and will produce 64K more than or less than the correct result if there is underflow or overflow, respectively.

The relationship between underflow/overflow and the carry-out from the ALU MSB is shown in the following paragraphs.

1. Unsigned integers:

   Decimal: 0 <= x < 64K
   Octal: 0 <= x <= 177777

   When ADDing two numbers, if the true result is less than 64K, the ALU will produce the correct result and no carry-out will result. If the true result is greater than or equal to 64K, the ALU will produce 64K less than the true result (i.e., the true result truncated to 16 bits) and a carry-out will result. Note that in these cases, a carry-out is synonymous with overflow and indicates that the ALU output is not the true result.

   SUBtraction is accomplished in the ALU by complementing the subtrahend and adding it to the minuend with a carry-in. Therefore, when SUBtracting one unsigned integer from another, if the true result is positive or zero, the ALU will produce the true result and will also produce a carry-out.

If the true result is negative, the ALU will produce the true result plus 64K (since all numbers are interpreted as positive), and no carry-out will result. Note that in these cases a carry-out is the opposite of underflow and indicates that the ALU output is the true result.

2. Signed Integers:

    Decimal: -32K <= x < 32K
    Octal: -100000 <= x <=77777

When ADDing two positive integers (or SUBtracting a negative integer from a positive one), if the true result is less than 32K, the ALU will produce the true result and no carry-out. If the true result is greater than or equal to 32K, the ALU output will appear negative (since the MSB = 1), will be 64K less than the true result, and no carry-out will occur. Note that in this case an overflow is not signalled by a carry-out.

When ADDing two integers with opposite signs (or SUBtracting two numbers having the same sign) the ALU will always produce the true result since the true result must be between -32K and +32K. A carry-out will occur if the result is positive and not if it is negative.

When ADDing two negative numbers (or SUBtracting a positive number from a negative one) if the true result is greater than or equal to -32K, the ALU will produce the true result. If the true result is less than -32K the ALU output will appear positive (MSB=0) and will be 64K greater than the true result. In either case, a carry-out will always occur.

These relationships are illustrated in Figure 5-7.

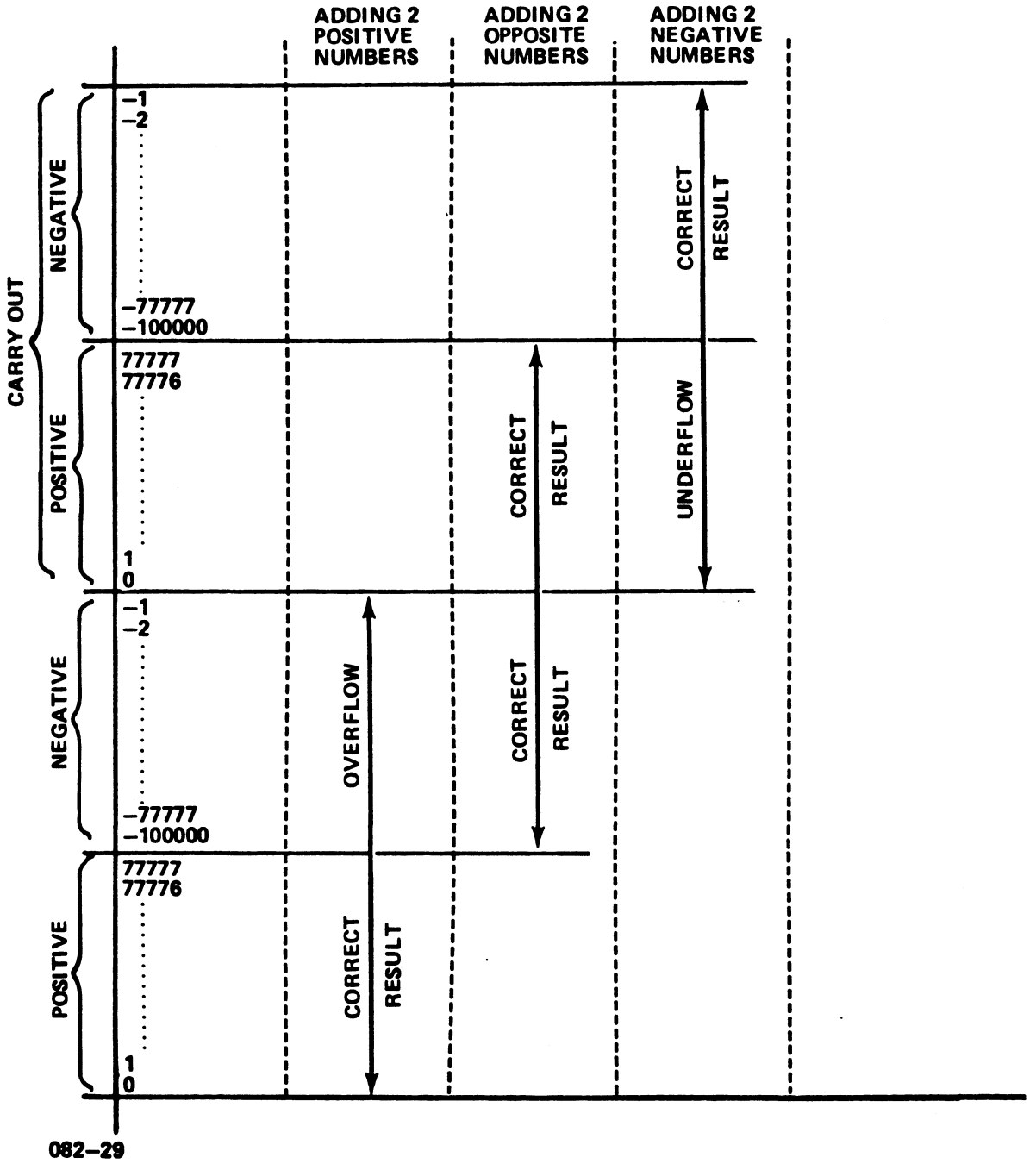


Figure 5-7. Overflow and Carry Operations Analysis for Signed Integers

5.5.2 ARITHMETIC/LOGIC FUNCTIONS

The OPCODE field (bits 5 through 7) defines one of eight arithmetic/logic operations to be performed by the 16-bit ALU as shown in Table 5-6.

TABLE 5-6. ARITHMETIC/LOGIC FUNCTIONS

Bits 5-7	OPCODE	Definition
000	COM	Complement - Complement the contents of ACS. Do not modify the preselected carry bit.
001	NEG	Negate - Produce the two's complement of the contents of ACS. If ACS=0, complement the preselected carry bit.
010	MOV	Move - Supply the unmodified contents of ACS. Do not modify the preselected carry bit.
011	INC	Increment - Add 1 to the contents of ACS. If the result is 0, complement the preselected carry bit.
100	ADC	Add Complement - Add the complement of ACS to ACD. Complement the preselected carry bit if ACS is less than ACD.*
101	SUB	Subtract - Subtract ACS from ACD. Complement the preselected carry bit if ACS is less than or equal to ACD.*
110	ADD	Add - Add the contents of ACS to the contents of ACD. If the unsigned sum is greater than or equal to two to the sixteenth power, complement the preselected carry bit.
111	AND	And - Logically AND the contents of ACS with the contents of ACD. Do not modify the preselected carry bit.
*Using a 16-bit unsigned integer interpretation.		

5.5.3 SECONDARY FUNCTIONS

The Shift (SH), Carry (CY), No-load (NL), and Skip (SK) fields specify secondary operations performed on the ALU result produced by the OPCODE field. These fields are discussed in the sections that follow.

5.5.3.1 Shift Field (SH)

The SH field (bits 8 and 9) determines the shifting action (if any) produced by the Shifter on the result of the calculation produced by the ALU, as shown in Table 5-7.

TABLE 5-7. SHIFT FIELD DEFINITIONS

Bits 8-9	Mnemonic	Definition
00	-	No Shift - Do not modify the ALU result. The carry resulting from the ALU operation is unaffected.
01	L	Left Rotate - Shift the result one place to the left, and insert the state of the carry resulting from the ALU (CRES) in the LSB (bit 15) position. Insert the out-shifted MSB (bit 0) into the carry bit (CNEW).
10	R	Right Rotate - Shift the result one place to the right, and insert the state of the carry resulting from the ALU (CRES) into the MSB (bit 0) position. Insert the out-shifted LSB (bit 15) into the carry bit (CNEW).
11	S	Swap - Swap the eight MSBs of the result with the eight LSBs. The carry resulting from the ALU is unaffected.

5.5.3.2 Carry Control Field (CY)

The CY field (bits 10 and 11) specifies the base to be supplied to the ALU for carry calculation, as shown in Table 5-8.

TABLE 5-8. CARRY CONTROL FIELD

Bits 10-11	Mnemonic	Definition
00	-	No change - The current state of the carry flag is supplied to the ALU as a base for carry calculation.
01	Z	Zero - The value 0 is supplied to the ALU as a base for carry calculation.
10	O	One - The value 1 is supplied to the ALU as a base for carry calculation.
11	C	Complement - The complement of the current state of the carry flag is supplied to the ALU as a base for carry calculation.

The three logical functions (MOV, COM, AND) supply the values listed above as the carry bit to the Shifter. The five arithmetic functions (ADD, ADC, INC, SUB, NEG) supply the complement of the base value if the ALU operation produces a carry-out of bit 0; otherwise they supply the value listed above.

5.5.3.3 No-Load Field (NL)

The NL field (bit 12) determines whether or not the output of the Shifter is stored in ACD and in Carry. If bit 12=0, the Shifter output is stored in ACD and in Carry. If bit 12=1, no storage action occurs.

5.5.3.4 Skip Control Field (SK)

The SK field determines the type of skip test to be performed on the Shifter output. If the selected skip test is affirmative, the next instruction is skipped. The skip tests that can be selected by the SK field (bits 13-15) are shown in Table 5-9.

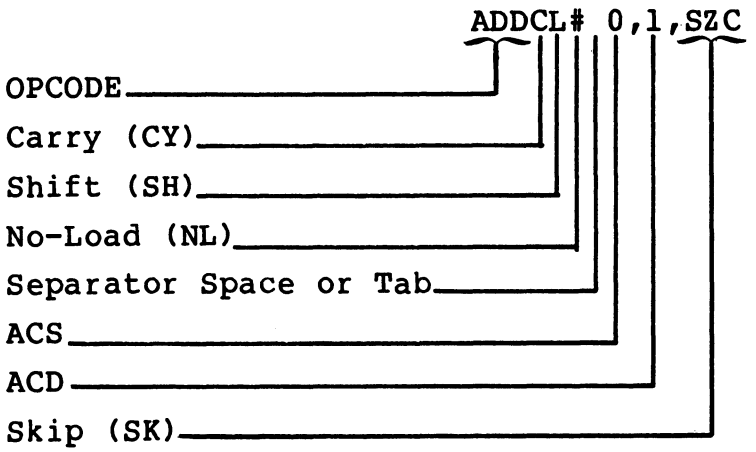
TABLE 5-9. SKIP CONTROL FIELD

Bits 13-15	Mnemonic	Definition
000	-	No skip test (never skip)
001	SKP	Skip unconditionally (no skip test required)
010	SZC	Skip if carry bit is zero
011	SNC	Skip if carry bit is nonzero
100	SZR	Skip if result is zero
101	SNR	Skip if result is nonzero
110	SEZ	Skip if either carry bit or result is zero
111	SBN	Skip if both carry bit and result are nonzero

5.5.4 ASSEMBLER LANGUAGE CONVENTIONS AND EXAMPLES

The assembler language arithmetic or logical instruction consists of the instruction OPCODE mnemonic (ADD, NEG, COM, etc.) followed by symbols that specify the carry indicator, the shift indicator, the load/no-load indicator, a source and a destination accumulator and the skip conditions. Table 5-10 shows the programming conventions for arithmetic and logical instructions.

The format is as follows:



The CY, SH, NL, and SK fields are specified by adding the appropriate mnemonic symbols. None of these four fields has to be specified, but their symbols must appear in the proper order and place if they are included. Those fields not specified will be assembled containing 0s. For example

```
ADDCL 0,1
```

performs the following operation: Add A0 to A1 and supply the complement of the Carry flag to the ALU. Shift the 17-bit output to the left, and store it in A1 and the Carry flag.

TABLE 5-10. ASSEMBLER LANGUAGE CONVENTIONS FOR ARITHMETIC AND LOGICAL INSTRUCTIONS

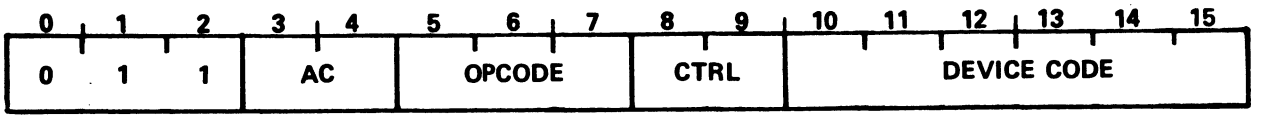
Instruction Function	OPCODE Mnemonic	Optional Secondary Functions			Separator Space or Tab	Accumulators				Optnl Skip SK*
		CY*	SH*	NL*		ACS	,	ACD	,	
Add	ADD									blank
Subtract	SUB									SKP
Move	MOV	none	none			0		0		SZC
Increment	INC	Z	L	none		1		1		SNC
Negate	NEG	O	R	#		2	,	2	,	SZR
Complement	COM	C	S			3		3		SNR
Add Complement	ADC									SEZ
Logical And	AND									SBN

*Elimination of a mnemonic symbol for these fields will cause the field to be assembled as all zeros.

5.6 INPUT/OUTPUT INSTRUCTION GROUP

The input/output instructions enable the processor to communicate with the peripheral devices on the system and also perform various operations within the processor. I/O instructions transfer data between accumulators and devices, start or reset device operation, or check the status of each device. Each I/O instruction contains a 6-bit device code field, which specifies the particular device for this data transfer. The system allows up to 63 peripheral devices, with each device assigned a unique code from 00 through 76 octal. The 77 octal code denotes a special class of instructions that controls certain CPU functions such as interrupt handling. Use of the 00 code is not recommended, since a device with that code would give a default response to an Interrupt Acknowledge instruction.

All instruction words in this category have the format shown in Figure 5-8.



082-30

Figure 5-8. Input/Output Instruction Format

An instruction in this class is designated by 011 in bits 0-2. The OPCODE and Control (CTRL) fields define the I/O operation to be performed. If a data transfer operation is involved, the ac field (bits 3 and 4) specifies the accumulator involved in the data transfer (otherwise it has no effect). Bits 10-15 select the device that is to respond to the instruction.

5.6.1 PROGRAMMED I/O INSTRUCTIONS

Programmed I/O instructions apply to all device codes except code 77 (CPU). I/O transfer instructions are:

DIA, DOA

The POINT 4 MARK 3 programmed input/output system provides for full 16-bit data transfer:

Input channel (DIA)
Output channel (DOA)

Each device interface contains a 6-bit address decoder (bits 10-15). When the processor executes an I/O instruction, it places the specified device code onto the Device Select lines of the I/O Bus. The appropriate device will recognize its own code and thus respond to the I/O instruction. All other devices ignore the instruction.

5.6.1.1 I/O Transfer Instructions

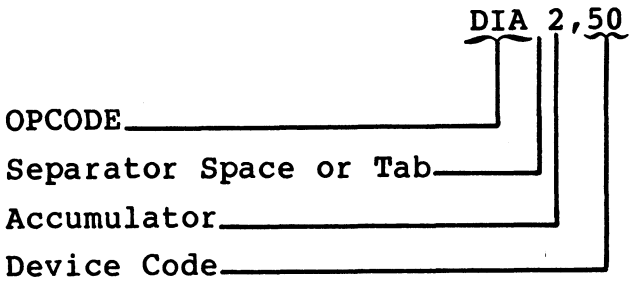
I/O Transfer instructions move data between the processor and the device interface. The OPCODE field (bits 5-7) of the instruction specifies the type of transfer to take place (Data In, Data Out, No Transfer, etc.). Bits 3 and 4 specify the accumulator that supplies or receives the data and bits 8 and 9 specify a control function which is not used in I/O transfer instructions. The type of transfer is determined by the code in the OPCODE field as illustrated in Table 5-11. The control (CTRL) field is not used in these instructions.

TABLE 5-11. I/O TRANSFER INSTRUCTIONS

Bits 5-7	OPCODE	Definition
000	NIO	No data transfer involved.
001	DIA	Transfers (reads) the contents of either the status or the data register in the specified device into the CPU accumulator indicated in the AC field. If the device code is even, the contents of the status register will be transferred. If the device code is odd, the contents of the data register will be transferred.
010	DOA	Transfers (writes) the contents of the accumulator indicated by the AC field to either the command or the data register in the specified device. If the device code is even, the contents of the accumulator will be transferred to the command register. If the device code is odd, the contents of the accumulator will be transferred to the data register. **NOTE** Valid device codes are: 10-17 = MUX ports 0-3 20-25 = MUX expansion ports 4-6 30-37 = Floppy 0-3 50-55 = Disc 0 & 1 60-62 = Tape 0-3

5.6.1.2 Assembler Language Conventions and Examples

An assembler language I/O Transfer Statement consists of the instruction mnemonic, an optional control function, an accumulator and octal device code. Table 5-12 shows the programming conventions for input/output transfer instructions. For example



This instruction transfers the contents of A2 to the IOCB pointer for device 50 (disc). If A2=0, DMA transfer will not be activated; otherwise, DMA transfer will be activated.

The device code may be represented by a device mnemonic. Thus,

```
DIA 2,DSC
```

is equivalent to the previous example because DSC represents device 50 (disc controller).

A No-I/O (NIO) will not specify an accumulator since no data transfer occurs:

```
NIO DSC
```

TABLE 5-12. ASSEMBLER LANGUAGE CONVENTIONS FOR INPUT/OUTPUT TRANSFER INSTRUCTIONS

Instruction Function	OPCODE Mnemonic	Device Function	Separator Space/Tab	Accumulator	,	Device Code
No Input/ Output	NIO	Not Used		None		
Data Input	DIA	Not Used		0 1	,	00-76 Octal
Data Output	DOA	Not Used		2 3		

5.6.2 SPECIAL CODE 77 (CPU) INSTRUCTIONS

Certain system functions and interrupt processing control are accomplished via I/O instructions with the octal code 77 in bits 10-15. These instructions do not directly address a particular peripheral device. The device code mnemonic is CPU.

CPU instructions have the same general format as I/O transfer instructions. The OPCODE field and Control field, however, are interpreted differently.

OPCODE Field:

- Addresses all I/O devices simultaneously for certain interrupt control functions.
- Does nondata transfer functions such as resetting all I/O devices and transferring control to the Virtual Control Panel program MANIP.

Control Pulse:

- Addresses the Multiplexer IOCB pointer
- Addresses the Disc Controller IOCB pointer
- Addresses the Tape Controller IOCB pointer
- Addresses the Floppy Disc Controller IOCB pointer

Table 5-12 gives details on special I/O instructions.

The CPU has two flags which can be tested by the I/O Skip instructions with device code 77:

- BUSY - ION set (Interrupts are enabled)
- DONE - Power-failure has been detected (causes interrupt if ION set; software will stop operations in order to prevent power failure while performing a disc transfer.)

5.6.2.1 Special Mnemonics for CPU Instructions

The assembler also recognizes several special mnemonics for CPU instructions. The regular instruction mnemonics and the special mnemonics along with a description of special CPU instructions are listed in Table 5-13.

TABLE 5-13. SPECIAL CPU I/O INSTRUCTIONS

Instruction	Special Mnemonic	Definition
NIO CPU	-	No Action
NIOS CPU	INTEN	Set the processor's Interrupt On (ION) flag. The processor will now respond to interrupt requests from devices, after execution of one more instruction.
NIOC CPU	INTDS	Clear the Interrupt On flag, so that the processor will not respond to interrupt requests.
DOB ac,CPU	MSKO ac	Set up the Interrupt Disable flags in all devices simultaneously, according to the mask code in accumulator ac. A MSKO with ac>1 and LSB (ac) = 1 disables interrupts from all devices. A MSKO with ac>1 and LSB (ac) = 0 enables all interrupts. The values ac=1 or ac=0 have special meanings: ac=1 disables all DMA ac=0 enables DMA to controllers
DICC ac,CPU	IORST	Clears the processor ION flag; clears any pending interrupts; clears DMA mode (MSKO with LSB of accumulator set to 1); clears MSKOUT (MSKO with LSB set to 0). Also sends IOBRST pulse to Peripheral Interface Board which clears all disc control registers, and all tape or floppy disc control registers, but does not change the IOCB pointers or the multiplexer communications controller setup.
DOC ac,CPU	HALT	Transfers control to the processor Virtual Control Panel, thus enabling use of MANIP commands for front panel operations.

Note that the special mnemonic does not allow the programmer to specify the S and C functions. For example

```
MSKO 3
```

when executed, sets the interrupt status based on the value in accumulator 3. To activate the disc controller it would be necessary to use

```
DOBC 3,CPU
```

The instruction IORST, however, assumes the C function. All I/O device flags are reset and the ION flag is cleared.

5.6.2.2 Control Field Uses

The control field is used in conjunction with DIB and DOB instructions with device code 77 to control interrupt polling and DMA transfers. Table 5-14 describes the functions of the control codes (S, C and P) when used with DIB and DOB instructions.

TABLE 5-14. CONTROL FIELD DEFINITIONS FOR I/O INSTRUCTIONS WITH DEVICE CODE 77

OPCODE Bits 5-7	OPCODE Mnemonic	Control Bits 8-9	Control Mnemonic	Definition
011	DIB	00	None	None
011	DIB	01	S	Reads pointer for multiplexer IOCB to determine if an interrupt is pending (MSB=1).
011	DIB	10	C	Reads pointer for disc controller IOCB to determine if an interrupt is pending (MSB=1).
011	DIB	11	P	Reads pointer for tape or floppy disc controller IOCB to determine if an interrupt is pending (MSB=1).
100	DOB	00	None	MSKO instruction - see Table 5-13.
100	DOB	01	S	Sets pointer to multiplexer IOCB but does not activate multiplexer. DOA instructions must be issued to those ports to be activated.
100	DOB	10	C	Sets pointer to disc controller IOCB and activates disc controller. The controller then performs automatic block transfer based on information in IOCB. If a DOBC is issued with accumulator set to zero, automatic block transfer is shut off and/or pending interrupt bit for disc controller is reset.
100	DOB	11	P	Sets pointer to tape or floppy IOCB and activates tape or floppy controller. The controller then performs automatic block transfer based on information in IOCB. If a DOBP is issued with accumulator set to zero, automatic block transfer is shut off and/or pending interrupt bit for tape or floppy controller is reset.

5.6.2.3 Skip Instructions

A value of 111 in bits 5-7 signifies a conditional skip instruction. The function field in this case indicates which processor flag (Interrupt On or Power Fail) will be tested, as shown in Table 5-15.

TABLE 5-15. I/O SKIP INSTRUCTIONS

Bits 8-9	Instruction	Definition
00	SKPBN CPU	Skip next instruction if Interrupt On is nonzero.
01	SKPBZ CPU	Skip next instruction if Interrupt On is zero.
10	SKPDN CPU	Skip next instruction if the Power Failure flag is nonzero.
11	SKPDZ CPU	Skip next instruction if the Power Failure flag is zero.

5.6.2.4 Assembler Language Conventions and Examples

CPU instructions are usually written using the special mnemonics shown in Section 5.6.2.1. However they may also be written in the same manner as I/O transfer instructions, specifying the instruction mnemonic, optional control function, optional accumulator, and a device code of 77 octal (mnemonic CPU). For example

```
NIOS CPU
```

sets the ION flag in the processor.

For additional programming examples, see Appendix E.

5.7 INSTRUCTION EXECUTION TIMES

One of the outstanding features of the POINT 4 MARK 3 is the reduction in instruction time over execution time in comparable mini-computers. Table 5-16 gives instruction execution times for the POINT 4 MARK 3.

These times are exclusive of two types of overhead:

1. A 600-nanosecond refresh cycle takes place once every 16 microseconds - this adds about 4% overhead.

2. Arithmetic/Logic instructions on RAM page boundaries (2 least significant digits of address = 76 or 77) take an extra 200 nanoseconds - this results in approximately 0.5% overhead.

TABLE 5-16. INSTRUCTION EXECUTION TIMES

Instruction Category	Instruction (Generic Types)	Execution Times (ns)
MEMORY REFERENCE	Load or Store Accumulator (LDA,STA)	1400
	Increment or Decrement if Zero (ISZ,DSZ)	1800
	Jump (JMP)	800
	Jump to Subroutine (JSR)	800
	For Indirect Addressing, add	600
ARITHMETIC/ LOGIC	Arithmetic/Logic Instructions (COM,NEG,MOV,INC,ADC,SUB,ADD,AND)	600
	For skip, add	0
	For swap, add	200
INPUT/ OUTPUT	Input (DIA)	2600
	Output (DOA)	2200
	I/O Skips - CPU only (SKPBN,SKPBZ,SKPDN,SKPDZ)	2000
DATA CHANNEL TRANSFERS	Disc Transfers:	
	Input (16 bits)	1200
	Output (16 bits)	1200
	Floppy Disc Transfers:	
	Input (16 bits)	4600
	Output (16 bits)	5000
	MUX Transfers (includes automatic vectoring, special character test, and buffer-end test):	
	Input (8 bits)	9600
	Output (8 bits)	8400
	Tape Transfers:	
	Input (8 bits)	5600
	Output (8 bits)	5600

Section 6
OPTIONAL FEATURES

6.1 INTRODUCTION

The 64K-byte Memory Expansion Board and the Port Expansion Board are options designed to significantly enhance the performance of the POINT 4 MARK 3. The memory expansion option increases memory capacity from 64K bytes to 128K bytes; the port expansion option provides three additional asynchronous ports for increased system capability. When these options are combined, dramatic increases in performance can be realized. Both options are piggy-back boards designed for easy installation.

6.2 64K-BYTE MEMORY EXPANSION BOARD

The Memory Expansion Board option provides 64K bytes of additional memory, significantly expanding the MARK 3's capabilities. The doubling of MARK 3 memory capacity allows for substantial increase in system performance. The Memory Expansion Board is a piggy-back, plug-in board positioned on top of the main CPU board via the 50-pin header connector on the CPU board. The expansion board has two connectors: its black connector sits flat on the board for normal operations; its green connector sits at right angles to the board for debugging purposes. The board is cut out to facilitate test connections.

See Figure 6-1 for an illustration of the 64K-byte Memory Expansion Board.

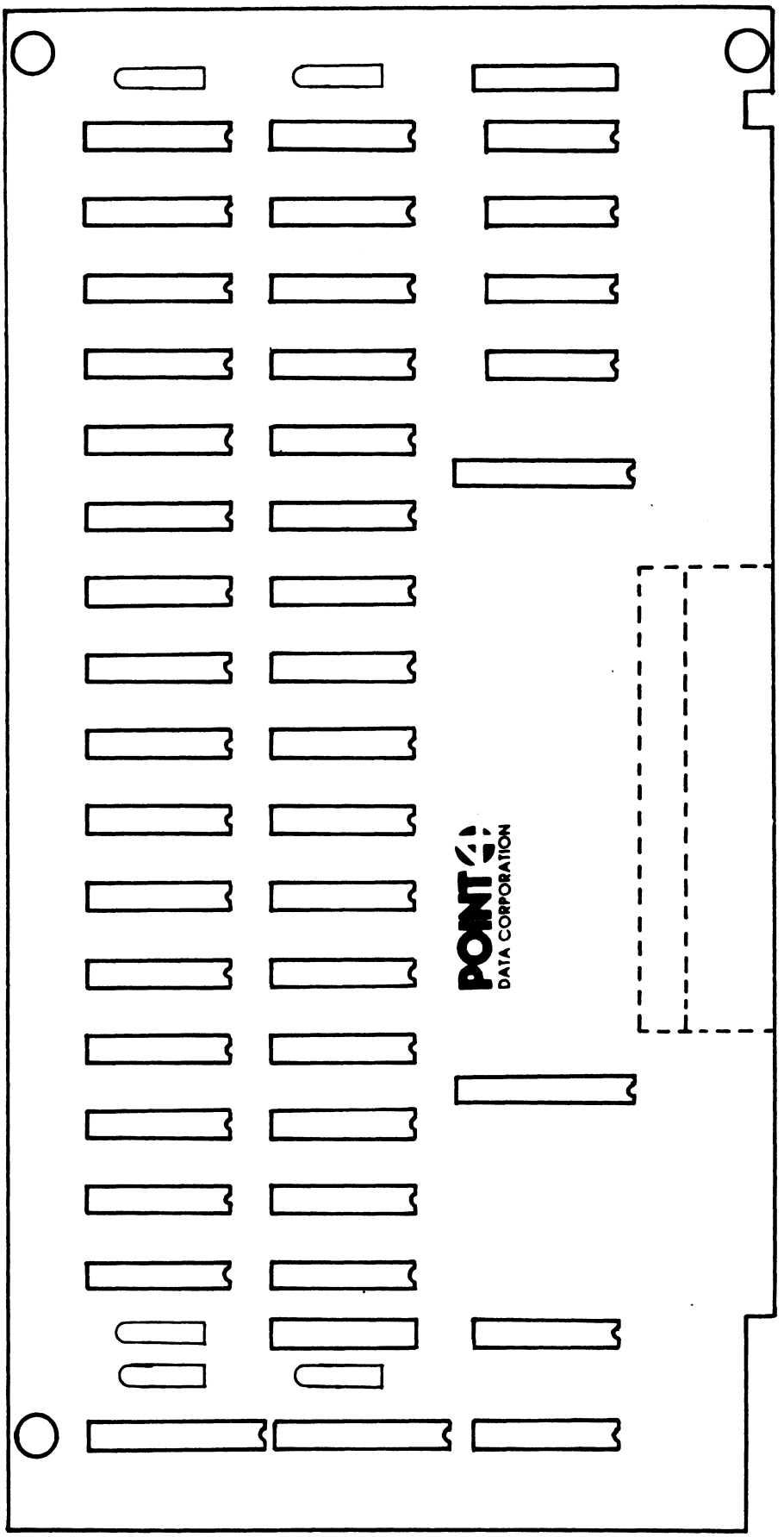


Figure 6-1. POINT 4 MARK 3 64K-byte Memory Expansion Board

6.3 PORT EXPANSION BOARD

This option adds three asynchronous ports to the system, significantly increasing MARK 3 capabilities, especially when added in conjunction with the memory expansion board. These three additional ports are functionally identical to the existing MARK 3 ports, allowing access from additional terminals. With the Port Expansion Board option, MARK 3 easily expands its capabilities to seven ports.

The Port Expansion Board is a piggy-back, plug-in board positioned on top of the Peripheral Interface Board (PIB) via the 50-pin header connector on the PIB. The expansion board has two connectors: its black connector sits flat on the board for normal operations; its green connector sits at right angles to the board for debugging purposes. The board is cut out to facilitate test connections.

See Figure 6-2 for an illustration of the Port Expansion Board.

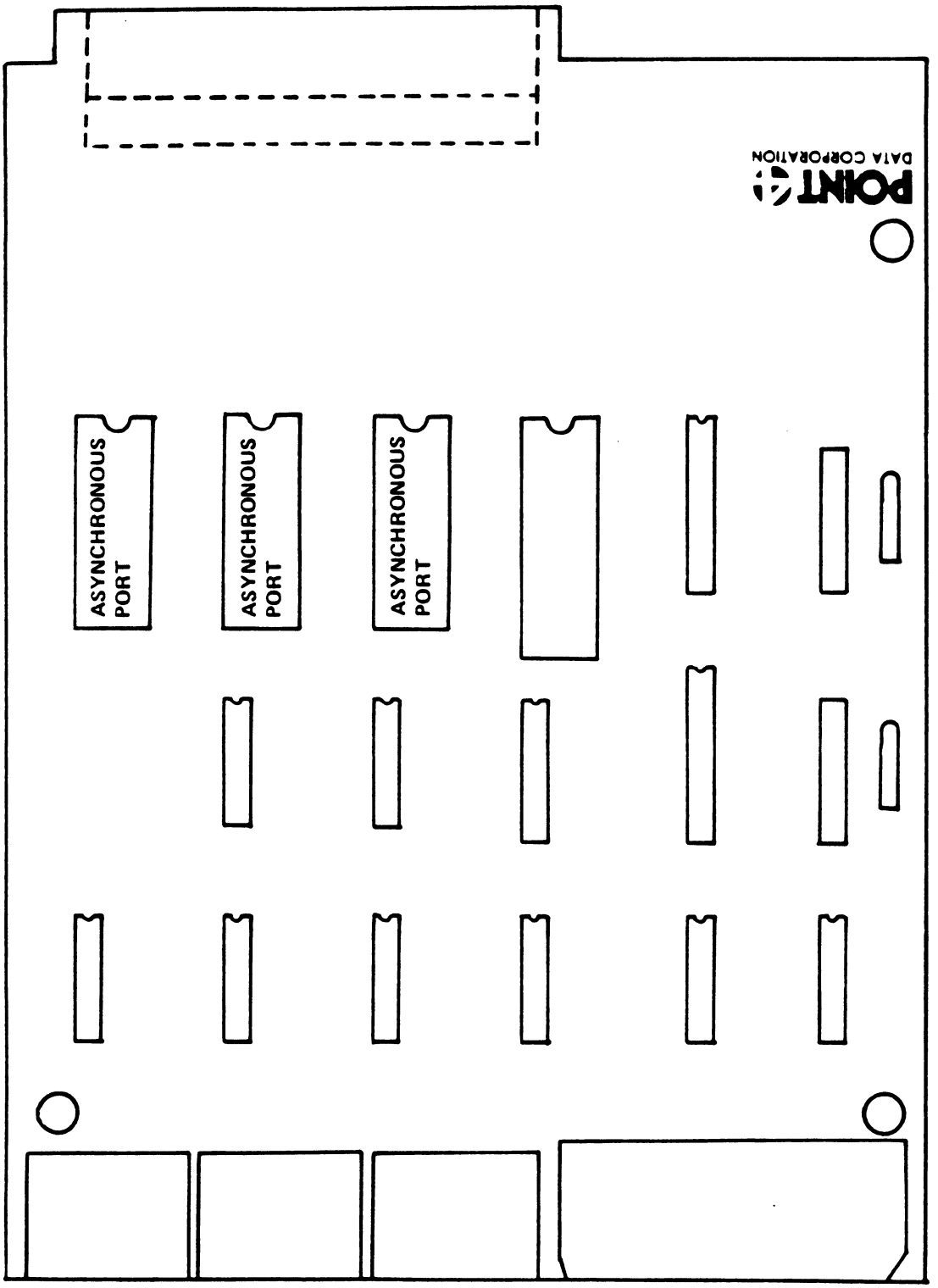


Figure 6-2. POINT 4 MARK 3 Port Expansion Board

APPENDICES

Appendix A
CABLE LENGTH CONSIDERATIONS

The following considerations apply to the specification of maximum cable length between the POINT 4 MARK 3 MUX and a CRT or other terminal, using the RS-232 interface.

The Electronics Industries Association (EIA), in Recommended Standard RS-232C, states, "The use of short cables (each less than approximately 50 feet...) is recommended; however, longer cables are permissible, provided that the resulting load capacitance...does not exceed 2500 picofarads." The 50-foot recommendation is generally viewed as extremely conservative.

In normal noise environments, the limiting factor determining maximum cable length is waveshape distortion due to resistance-capacitance effects. This factor is directly proportional to line length and baud rate. The following maximum line lengths represent safe engineering practice:

Baud Rate	Maximum Length
9600	500 feet
4800	1000 feet
2400	1/2 mile
1200	1 mile

The same relationship should continue for several miles, after which the limiting factor will become resistive voltage drop.

Ordinary unshielded wire may be used such as telephone company interior wiring. In foot-traffic areas, standard 3-wire a.c. line cord with heavy insulation makes a sturdy, inexpensive cable.

In noisy environments (cable runs in close proximity to fluorescent lights or air conditioning or elevator motors), the limiting factor becomes noise pickup and shielded wire should be used. The same line lengths may be employed.

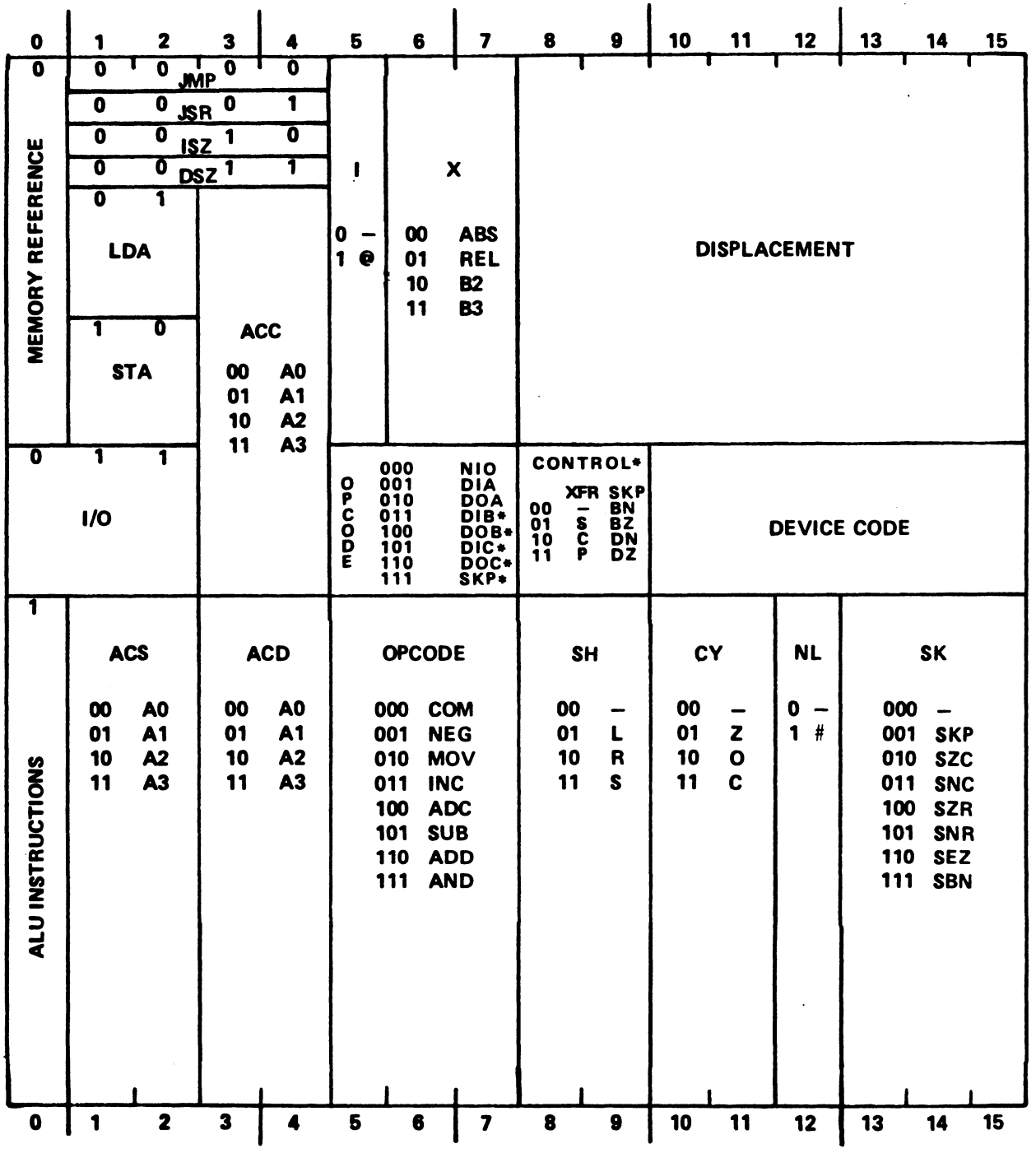
The only effect of excessive cable length will be the occasional incorrect transmission of a character. If this happens too frequently to be acceptable, a lower baud rate should be used.

Appendix B
ASCII CODE CHART

ASCII CODE in OCTAL

000	NUL	<CTRL-@>	040	BLANK	100	@	140	`
001	SOH	<CTRL-A>	041	!	101	A	141	a
002	STX	<CTRL-B>	042	"	102	B	142	b
003	ETX	<CTRL-C>	043	#	103	C	143	c
004	EOT	<CTRL-D>	044	$	104	D	144	d
005	ENO	<CTRL-E>	045	%	105	E	145	e
006	ACK	<CTRL-F>	046	&	106	F	146	f
007	BEL	<CTRL-G>	047	'	107	G	147	g
010	BKSP	<CTRL-H>	050	(	110	H	150	h
011	HTAB	<CTRL-I>	051	)	111	I	151	i
012	LF	<CTRL-J>	052	*	112	J	152	j
013	VTAB	<CTRL-K>	053	+	113	K	153	k
014	FF	<CTRL-L>	054	,	114	L	154	l
015	CR	<CTRL-M>	055	-	115	M	155	m
016	SO	<CTRL-N>	056	.	116	N	156	n
017	SI	<CTRL-O>	057	/	117	O	157	o
020	DLE	<CTRL-P>	060	0	120	P	160	p
021	XON	<CTRL-Q>	061	1	121	Q	161	q
022	AUXON	<CTRL-R>	062	2	122	R	162	r
023	XOFF	<CTRL-S>	063	3	123	S	163	s
024	AUXOFF	<CTRL-T>	064	4	124	T	164	t
025	NAK	<CTRL-U>	065	5	125	U	165	u
026	SYN	<CTRL-V>	066	6	126	V	166	v
027	ETB	<CTRL-W>	067	7	127	W	167	w
030	CAN	<CTRL-X>	070	8	130	X	170	x
031	ENDMD	<CTRL-Y>	071	9	131	Y	171	y
032	SUB	<CTRL-Z>	072	:	132	Z	172	z
033	ESC	<CTRL-[>	073	;	133	[	173	{
034	F SEP	<CTRL-\>	074	<	134	\	174	|
035	G SEP	<CTRL-]>	075	=	135	]	175	}
036	R SEP	<CTRL-^>	076	>	136	^	176	~
037	U SEP	<CTRL-_>	077	?	137	_	177	DEL

Appendix C
VON NEUMANN MAP OF POINT 4 MARK 3 COMMANDS



*DEVICE 77 (CPU) ONLY

082-37

Appendix D
POINT 4 MARK 3 INSTRUCTION CHART

ARITH/LOGIC		MEMORY REFERENCE		INPUT/OUTPUT	
100000	COM	0	JMP	60000	NIO
100400	NEG	4000	JSR	60400	DIA
101000	MOV			61000	DOA
101400	INC	10000	ISZ	61400	DIB *
102000	ADC	14000	DSZ	62000	DOB *
102400	SUB			62400	DIC *
103000	ADD	20000	LDA	63000	DOC *
103400	AND	40000	STA		
				ACCUMULATOR	
SOURCE		ACCUMULATOR		0	0
0	0	0	0	4000	1
20000	1	4000	1	10000	2
40000	2	10000	2	14000	3
60000	3	14000	3		
				I/O PULSE *	
DESTINATION		INDIRECT		100	S
0	0	2000	@	200	C
4000	1			300	P
10000	2	ADDRESS MODE			
14000	3	0	ABS	I/O SKIP *	
		400	REL	63400	SKPBN
SHIFT		1000	BASE2	63500	SKPBZ
100	L	1400	BASE3	63600	SKPDN
200	R			63700	SKPDZ
300	S	DISPLACEMENT			
		0–177	POS.	DEVICE CODE	
CARRY		200–377	NEG.	10	PZS
20	Z			11	PZD
40	O			12	P1S
60	C			13	P1D
		SPECIAL ARITHMETIC TESTS		14	P2S
NO–LOAD				15	P2D
10	#	101014	SKZ	16	P3S
		101015	SNZ	17	P3D
SKIP CONDITION		101112	SSP		
1	SKP	101113	SSN	SPECIAL CPU INSTRUCTIONS	
2	SZC	102032	SGE		
3	SNC	102033	SLS	60017	INTEN
4	SZR	102414	SEQ	60277	INTDS
5	SNR	102415	SNE	62077	MSKO
6	SEZ	102432	SGR	62677	IORST
7	SBN	102433	SLE	63077	HALT

*DEVICE CODE 77 (CPU) ONLY

Appendix E
PROGRAMMING EXAMPLES

E.1 NUMBER HANDLING

E.1.1 GENERATING NUMBERS

Six numbers can be generated with single instructions:

```
SUB     0,0 --> 0
SUBZL   0,0 --> 1
SUBZR   0,0 --> 100000
ADC     0,0 --> 177777  (= -1)
ADCZL   0,0 --> 177776  (= -2)
ADCZR   0,0 --> 77777
```

E.1.2 NUMBER TESTING

Twenty different sets of numbers can be tested with a single instruction. Figure E-1 shows the conditions under which each of the basic arithmetic test instructions will skip. Figure E-2 shows which instructions to use to test the 20 sets of numbers. The skip condition can be changed from a zero-test to a nonzero-test to obtain the complements of the 20 sets.

Instr.	SZR skips if:	SZC skips if:
MOVZ	0	all
MOVO	0	none
MOVZL	0,100000	>=0
MOVOL	none	>=0
MOVZR	0,1	even
MOVOR	none	even
COMZ	-1	0
COMO	-1	none
COMZL	-1,-2	<0
COMOL	none	<0
COMZR	-1,77777	odd
COMOR	none	odd
INCZ	-1	not -1
INCO	-1	-1
INCZL	77777	-1<=x<=77777
INCOL	-1	-1<=x=77777
INCZR	0	odd
INCOR	-1	odd
NEGZ	0	not 0
NGEO	0	-1
NEGZL	100000	-77777<=X<0
NEGOL	0	-77777<=x<0
NEGZR	-1	even
NEGOR	0	even
ADDZ	0,100000	>=0
ADDO	0,100000	<0
ADDZL	0	2d bit = 0
ADDOL	100000	2d bit = 0
ADDZR	0	all
ADDOR	100000	all

NOTE

For ADD instructions, ACS and ACD must be the same.

Figure E-1. Conditions Under Which Each Of The Basic Arithmetic Test Instructions Will Skip

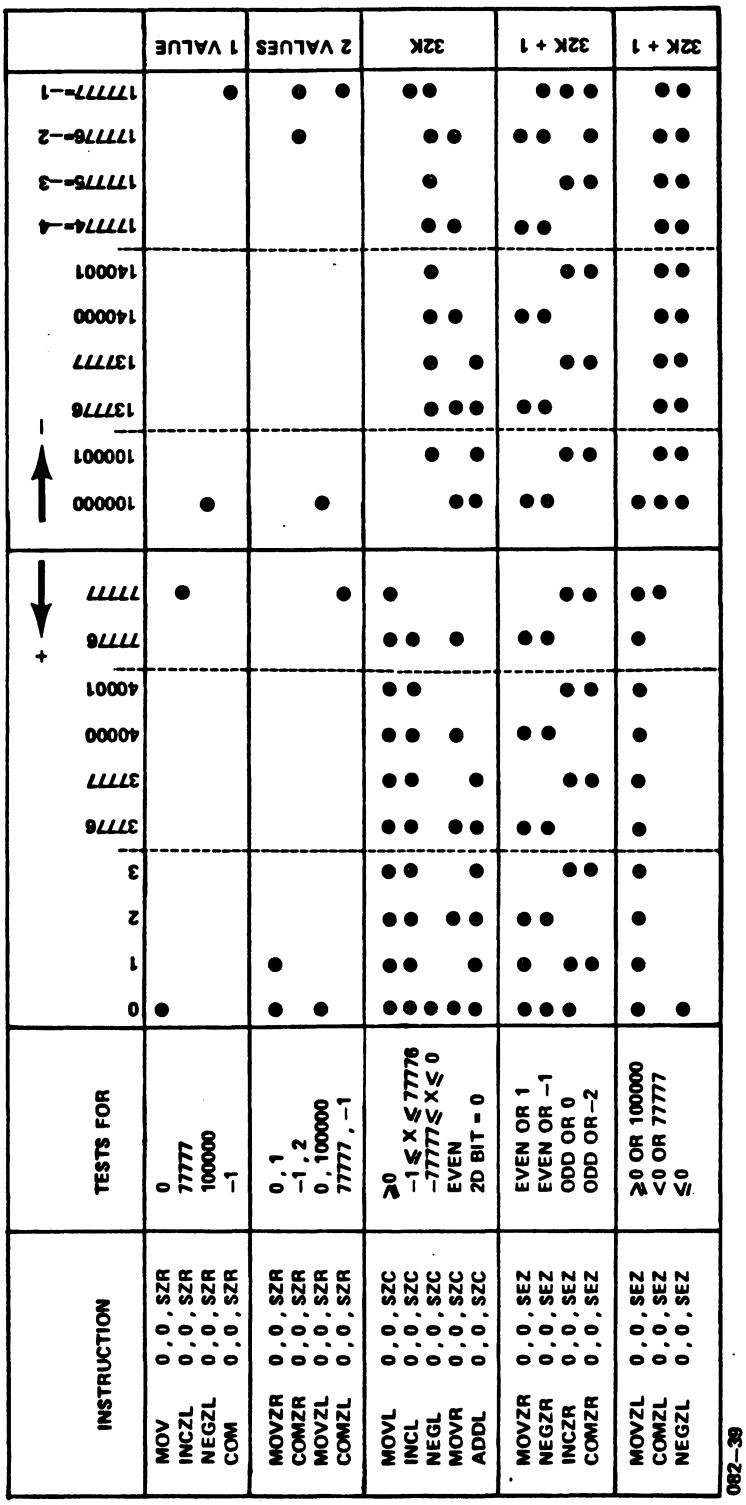


Figure E-2. The 20 Different Sets Of Numbers Which Can Be Tested With A Single Instruction

E.2 BIT TESTING

Any bit in a word can be tested with a maximum of three instructions, without requiring another accumulator. Three bit positions can be tested with just one instruction (bits 0, 1, and 15). Seven bit positions (bits 2, 3, 7, 8, 9, 10 and 14) require two instructions, and the other six require three. Figure E-3 shows which instructions to use to test for any bit in a word.

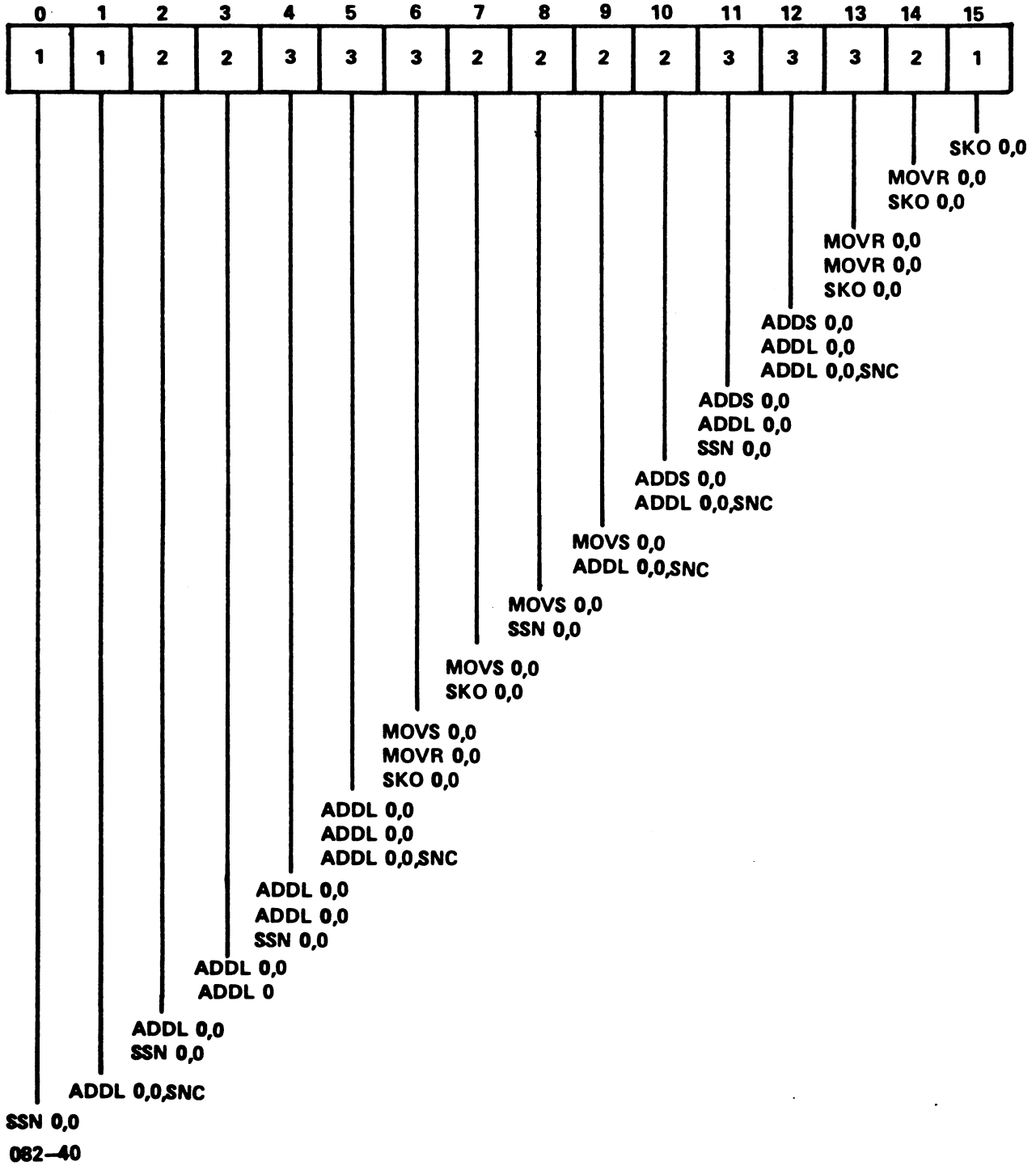


Figure E-3. How To Test For Any Bit In A Word

E.3 ACCUMULATOR HANDLING

E.3.1 TO "OR" TWO ACCUMULATORS

The following routine forms the inclusive-OR of A0 and A1 in A1. The routine uses the fact that an arithmetic ADD is equivalent to an OR if corresponding bits in the two operands are not both 1.

```
COM     0,0
AND     0,1  ;remove those bits where both are 1
ADC     0,1  ;then add original value
```

E.3.2 TO "EXCLUSIVE-OR" TWO ACCUMULATORS

This routine utilizes the fact that an arithmetic ADD is the same as an exclusive-OR except for the carry bits.

```
MOV     1,2
ANDZL   0,2  ;form the carry bits
ADD     0,1  ;add the original operands
SUB     2,1  ;remove the carry bits
```

This routine destroys the carry flag. To preserve the Carry at the expense of an additional instruction, use the following:

```
COM     1,2
AND     0,2  ;forms A0 * A1 (bar)
COM     0,0
AND     0,1  ;A0 (bar) * A1
ADD     2,1  ;A0 ⊕ A1 = A0 * A1 (bar) + A0 (bar) * A1
```

E.3.3 TO "DECREMENT" AN ACCUMULATOR

The following routine is used to decrement an accumulator:

```
NEG     0,0
COM     0,0
```

E.3.4 TO "COMPLEMENT" THE MOST SIGNIFICANT BIT

The following instruction complements the MSB and clears the carry bit:

```
ADDOR   0,0
```

E.4 PARITY GENERATION OR CHECKING

The two-instruction loop

```
ADD   0,0,SZR
JMP   .-1
```

will complement the original carry if A0 had odd parity, and leave it unchanged if A0 had even parity.

E.5 I/O PROGRAMMING FOR THE MASTER TERMINAL

The assembler listings in Figure E-4 provide examples of inputting and outputting to a Master Terminal (Teletype or CRT), using the standard Device Code 10/11-type controller. They also illustrate byte handling and interrupt handling conventions. For further information on I/O interfaces, see Section 4.

```
ASM  ,@$LPT1,M3PXB
AUG  4, 1982  10:52:11

; PROGRAMMING EXAMPLES

          1000        .LOC   1000

            10 PZS=   10     ;PORT ZERO STATUS INPUT OR CONTROL OUTPUT
            11 PZD=   11     ;PORT ZERO DATA INPUT OR OUTPUT


; INPUT/OUTPUT ROUTINES


; MASTER TERMINAL INPUT - ACCEPTS CHARACTER INTO A0

   1000  60410        DIA    0,PZS     ;READ PORT ZERO STATUS
   1001 101213        SKO    0,0       ;IS THERE ANY INPUT AVAILABLE ?
   1002    776        JMP    .-2       ;  NO, KEEP WAITING
   1003  60411        DIA    0,PZD     ;ACCEPT IT: CLEARS DONE BIT IN STATUS


; MASTER TERMINAL OUTPUT - ASSUMES OUTPUT CHARACTER IS IN A0

   1004  30405        LDA    2,C2      ;LOAD MASK BIT FOR OUTPUT REGISTER
   1005  64410        DIA    1,PZS     ;READ PORT ZERO STATUS REGISTER
   1006 133415        AND#   1,2,SNR   ;IS OUTPUT REGISTER EMPTY ?
   1007    776        JMP    .-2       ;  NO, WAIT TO FINISH PREV. OUTPUT
   1010  61011        DOA    0,PZD     ;  YES, OUTPUT THE CHARACTER

   1011      2 C2:    2      ;MASK BIT FOR OUTPUT REGISTER EMPTY
```

Figure E-4. Master Terminal I/O Programming Examples (1 of 4)

```
; SUBROUTINE TO TYPE ASCII TEXT

; INITIAL CONDITIONS: NONE
; CALLING SEQUENCE:
;     JSR    TYPE
;     (ASCII TEXT,
;     PACKED 2 CHARACTERS/WORD
;     WITH 0 BYTE TERMINATOR)
;     RETURNS HERE
; RETURN CONDITIONS: A0 = 0; A1, A2, A3 DESTROYED; C = 0

   1012  25400 TYPE: LDA    1,0,3    ;PICK UP 2 ASCII CHARACTERS
   1013 175420        INCZ   3,3      ;ADV. RETURN PNTR; C=BYTE CNTR
   1014  30775 TYPE2:LDA    2,C2     ;PICK UP MASK BIT
   1015  60410        DIA    0,PZS    ;READ PORT ZERO STATUS REGISTER
   1016 113415        AND#   0,2,SNR  ;IS OUTPUT REGISTER EMPTY ?
   1017    776        JMP    .-2      ;  NO, KEEP WAITING
   1020  20407        LDA    0,C377L  ;YES, PREPARE LEFT-BYTE MASK
   1021 123705        ANDS   1,0,SNR  ;EXTRACT A BYTE - IS IT 0 ?
   1022   1400        JMP    0,3      ;  YES, RETURN TO CALLER
   1023  61011        DOA    0,PZD    ;OUTPUT THE CHARACTER
   1024 125362        MOVCS  1,1,SZC  ;SWAP THE 2 ASCII CHAR.; CK. BYTE CNT
   1025    767        JMP    TYPE2    ;  TYPE THE SECOND CHARACTER
   1026    764        JMP    TYPE     ;  GET 2 MORE CHARACTERS TO TYPE

   1027 177400 C377L:177400           ;OCTAL 377 IN LEFT BYTE

; SUBROUTINE TO TYPE A NUMBER IN OCTAL FORM

; INITIAL CONDITIONS: A1 = NUMBER TO BE TYPED
; CALLING SEQUENCE:
;     JSR    TPOCT
;     RETURNS HERE
; RETURN CONDITIONS: A1 = 0; A0, A2, A3 DESTROYED; C = 1

   1030  20415 TPOCT:LDA    0,C.BIT  ;PREPARE FOR MSD = 1 BIT
   1031 101120 TPOC2:MOVZL  0,0      ;PRESET CARRY (MSB:1, OTHERS:0)
   1032 125105        MOVL   1,1,SNR  ;LEFT-SHIFT A BIT OUT OF A1 INTO C
   1033   1400        JMP    0,3      ;RETURN WHEN PUSHER BIT IS GONE
   1034 101103        MOVL   0,0,SNC  ;ASSEMBLE ASCII DIGIT; COMPLETE ?
   1035    775        JMP    .-3      ;  NO, GET MORE BITS
   1036  70410        DIA    2,PZS    ;READ PORT ZERO STATUS
   1037 151200        MOVR   2,2      ;POSITION OUTPUT REGISTER BIT TO LSB
   1040 151213        SKO    2,2      ;IS OUTPUT REGISTER EMPTY ?
   1041    775        JMP    .-3      ;  NO, WAIT FOR IT
   1042  61011        DOA    0,PZD    ;YES, OUTPUT THE ASCII DIGIT
   1043  20403        LDA    0,C.OCT  ;PREPARE FOR NEXT OCTAL DIGIT
   1044    765        JMP    TPOC2    ;CONTINUE THE LOOP

   1045 140014 C.BIT:140014 ;CONST. TO STRIP OFF 1 BIT & CNVT. TO ASCII
   1046  10003 C.OCT:010003 ;CONST. TO STRIP OFF 3 BITS & CNVT. TO ASCII
```

Figure E-4. Master Terminal I/O Programming Examples (2 of 4)

```
- PAGE 3 -

; BYTE MOVE SUBROUTINES

; ASSUMPTION: ALL BYTE ADDRESSES REFER TO LOWER 32K OF MEMORY


; GET A BYTE INTO A0 FROM BYTE ADDRESS GIVEN IN A1

; INITIAL CONDITIONS: A1 = BYTE ADDRESS
; CALLING SEQUENCE:
;     JSR    GETBY
;     RETURNS HERE
; RETURN CONDITIONS: A0 = DESIRED BYTE, A1 = UNCHANGED

    1047 131220 GETBY:MOVZR  1,2       ;CONVERT BYTE ADDRESS INTO WORD ADDRESS
    1050  21000        LDA    0,0,2    ;FETCH WORD CONTAINING DESIRED BYTE
    1051 101003        MOV    0,0,SNC  ;DO WE WANT LEFT BYTE ?
    1052 101300        MOVS   0,0      ;  YES, SWAP THE WORD
    1053  30403        LDA    2,C377   ;RIGHT BYTE MASK
    1054 143400        AND    2,0      ;MASK THE RIGHT BYTE
    1055   1400        JMP    0,3      ;RETURN

    1056    377 C377: 377


; PUT A BYTE FROM A0 INTO MEMORY AT BYTE ADDRESS GIVEN IN A1

; INITIAL CONDITIONS: A0 = GIVEN BYTE IN RIGHT HALF, LEFT HALF IMMATERIAL
;     A1 = BYTE ADDRESS
; CALLING SEQUENCE:
;     JSR    PUTBY
;     RETURNS HERE
; RETURN CONDITIONS: A0, A1 UNCHANGED

    1057  54414 PUTBY:STA    3,PUTBR  ;SAVE RETURN ADDRESS
    1060 131220        MOVZR  1,2      ;FORM WORD ADDRESS FROM BYTE ADDR.
    1061  34775        LDA    3,C377   ;GET MASK FOR RIGHT HALF
    1062 163403        AND    3,0,SNC  ;MASK GIVEN BYTE; GOES IN LEFT HALF ?
    1063 101301        MOVS   0,0,SKP  ;  YES, SWAP THE BYTE
    1064 175300        MOVS   3,3      ;  NO, SWAP THE MASK
    1065  25000        LDA    1,0,2    ;FETCH THE WORD WHERE BYTE IS TO GO
    1066 167400        AND    3,1      ;MAKE ROOM FOR THE BYTE
    1067 107000        ADD    0,1      ;INSERT THE BYTE
    1070  45000        STA    1,0,2    ;PUT THE WORD BACK
    1071 145100        MOVL   2,1      ;RESTORE A1
    1072   2401        JMP    @PUTBR   ;RETURN

    1073      0 PUTBR:0               ;SAVE RETURN ADDRESS
```

Figure E-4. Master Terminal I/O Programming Examples (3 of 4)

```
- PAGE 4 -

; INTERRUPT VECTORING

                0         .LOC   0
          0     0         0              ;INTERRUPTED P.C. WILL BE STORED HERE
          1  2000         INTSV          ;POINTER TO INTERRUPT SERVICE

               2000         .LOC   2000
     2000  40427 INTSV:STA   0,INTS0    ;\
     2001  44427         STA   1,INTS1  ; \
     2002  50427         STA   2,INTS2  ;  \ SAVE ACCUMULATORS
     2003  54427         STA   3,INTS3  ;  / AND CARRY
     2004 101100         MOVL  0,0      ; /
     2005  40426         STA   0,INTSC  ;/
     2006  61577         DIBS  0,CPU
     2007 101112         SSP   0,0      ;MUX INTERRUPT ?
     2010   2424         JMP   @IHMUX   ;  YES
     2011  61677         DIBC  0,CPU
     2012 101112         SSP   0,0      ;DISC INTERRUPT ?
     2013   2422         JMP   @IHDSC   ;  YES
     2014  61700         DIBP  0,0
     2015 101112         SSP   0,0      ;TAPE INTERRUPT ?
     2016   2420         JMP   @IHTAP   ;  YES
                         ;
     2017  20414 INTSR:LDA   0,INTSC    ;RETURN FROM INTERRUPT HANDLER ROUTINE
     2020 101200         MOVR  0,0      ; \
     2021  20406         LDA   0,INTS0  ;  \ RESTORE ACCUMULATORS
     2022  24406         LDA   1,INTS1  ;  / AND CARRY
     2023  30406         LDA   2,INTS2  ; /
     2024  34406         LDA   3,INTS3  ;/
     2025  60177         INTEN          ;RE-ENABLE INTERRUPTS
     2026   2000         JMP   @0       ;RETURN TO INTERRUPTED PROGRAM


     2027      0 INTS0:0              ;SAVE A0
     2030      0 INTS1:0              ;SAVE A1
     2031      0 INTS2:0              ;SAVE A2
     2032      0 INTS3:0              ;SAVE A3
     2033      0 INTSC:0              ;SAVE CARRY

     2034      0 IHMUX:0      ;INSERT MUX INTERRUPT HANDLER ADDRESS HERE
     2035      0 IHDSC:0      ;INSERT DISC INTERRUPT HANDLER ADDRESS HERE
     2036      0 IHTAP:0      ;INSERT TAPE INTERRUPT HANDLER ADDRESS HERE


---

C2      1011   C377    1056   C377L   1027   C.BIT   1045   C.OCT   1046
GETBY   1047   IHDSC   2035   IHMUX   2034   IHTAP   2036   INTS0   2027
INTS1   2030   INTS2   2031   INTS3   2032   INTSC   2033   INTSR   2017
INTSV   2000   PUTBR   1073   PUTBY   1057   PZD       11   PZS       10
TPOC2   1031   TPOCT   1030   TYPE    1012   TYPE2   1014
```

Figure E-4. Master Terminal I/O Programming Examples (4 of 4)

COMMENT SHEET

MANUAL TITLE POINT 4 MARK 3 Computer System Manual

PUBLICATION NO. HM-081-0019 REVISION B

FROM: NAME/COMPANY: _____

BUSINESS ADDRESS: _____

CITY/STATE/ZIP: _____

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 5755 SANTA ANA, CA.

POSTAGE WILL BE PAID BY ADDRESSEE:

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
2569 McCabe Way
Irvine, CA 92714

CUT ON THIS LINE

POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 863-1111