POINT 4 COMPUTER

USER MANUAL

POINT 4
DATA CORPORATION

POINT 4

USER
REFERENCE
MANUAL
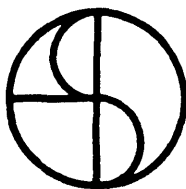
Educational Data Systems

NOTICE

Every attempt has been made to make this reference 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 Educational Data
Systems.

POINT 4 is a trademark of Educational Data Systems, Inc.

POINT 4 USER TABLE OF CONTENTS

SECTION I
INTRODUCTION AND GENERAL DESCRIPTION

SECTION II
INSTRUCTION REPERTOIRE

SECTION III
INPUT/OUTPUT INTERFACES

APPENDICES

TABLES

FIGURES

SECTION I

INTRODUCTION AND GENERAL DESCRIPTION

1.1 SCOPE

This manual provides general reference information, including
system description, equipment specifications, description of the
instruction repertoire, interface data, and installation and
operation information for the POINT 4 computer (see Figure 1-1),
manufactured by Educational Data Systems. This manual is designed
to be a general reference guide for both the programmer and the
operator, and as such contains both detailed information to
familiarize the user with the system, and charts for use as quick
reference material.

The manual is organized into six sections, as summarized below:

Section I	An introduction and system architecture description
Section II	Detailed description of the instruction repertoire
Section III	Peripheral interface guidelines
Section IV	Installation methods and diagnostic guidelines
Section V	Processor operating instructions and control unit functions and operation
Section VI	POINT 4 optional features

1.2 GENERAL DESCRIPTION

The POINT 4 computer is a 16-bit, high-speed, general purpose
minicomputer with a versatile instruction set. The POINT 4
employs a novel design architecture (Programmed Sequential Control
Logic) to achieve the simplicity and flexibility of a
microprogrammed design and the speed of a hard-wired logic design.
In addition the design allows direct addressing of up to 64K words
of MOS random access memory. These features make the POINT 4
computer well suited to OEM applications in business data systems,
communications, and control systems.
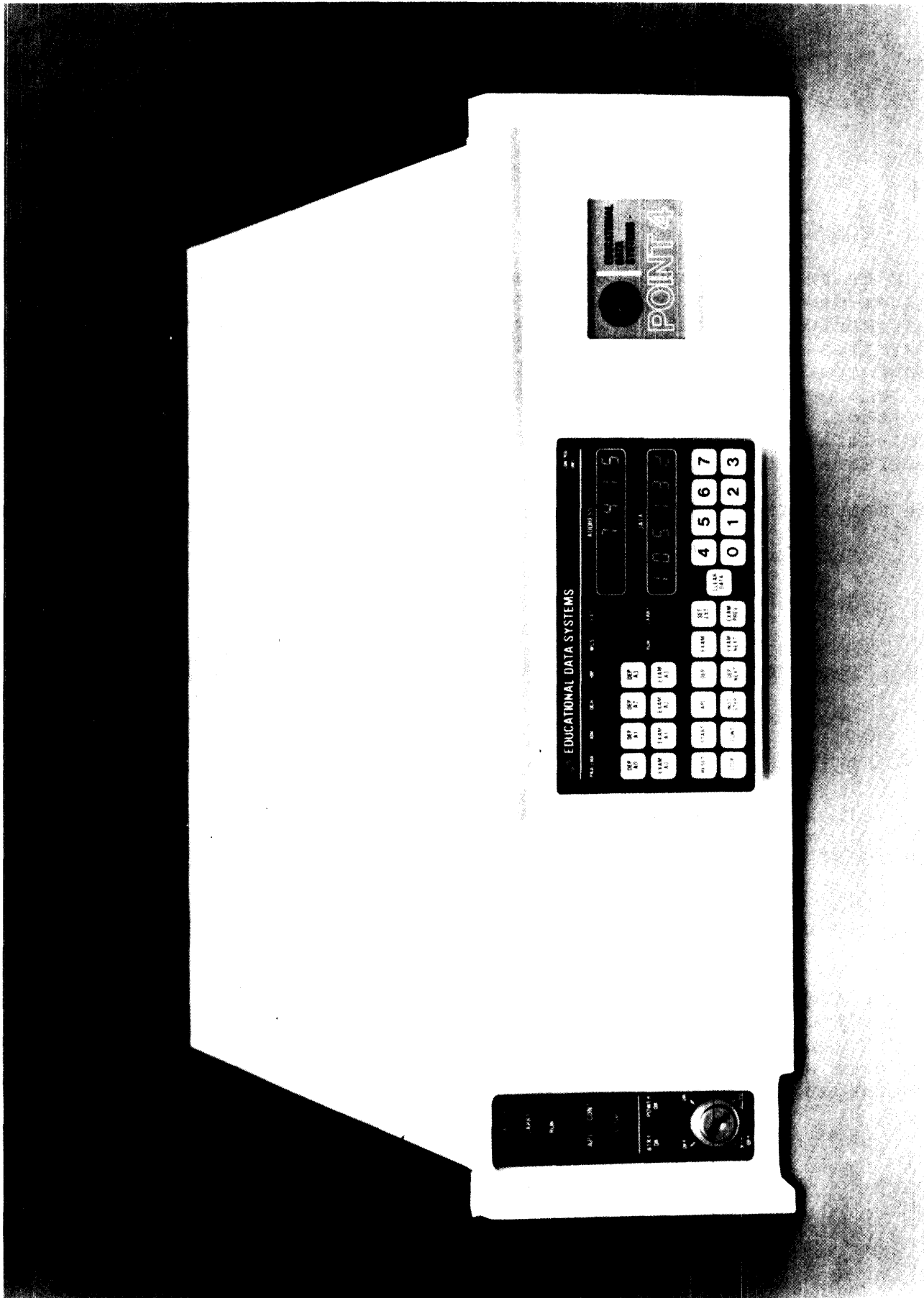
FIGURE 1-1. THE POINT 4 COMPUTER

1.2.1 Features

The Educational Data Systems POINT 4 series computer includes the following features:

- Memory up to 64K words on the same printed circuit board as the central processor
- Industry-compatible instruction set
- High-speed 400-nanosecond cycle time MOS memory
- RAM available in 32K and 64K words
- High-speed instruction processing (400ns ADD or JUMP, 800ns LOAD)
- Standard or high-speed data channel (jumper option)
- High-speed interprocessor bus option
- Programmable Control Store option which permits extension of the instruction set
- Main memory battery backup option
- "Virtual" Control Panel
- Built-in self-test for complete verification of memory and CPU operation
- Jumper-free backplane
- Detachable operator control unit option
- 7-slot front loading chassis and separate power supply
- Full IRIS Operating System available

1.2.2 Models

POINT 4 series computers are available in several configurations. All models use the same CPU/memory printed circuit board. Two models consist of only the CPU/memory board; two others include the chassis, front panel and power supply. Each pair differs in the amount of memory included. Table 1-1 shows the models available and the features included in each model.

1.2.3 Peripherals Supported

A complete range of POINT 4-compatible peripherals is available from various manufacturers. Peripherals available to interface with the POINT 4 are: Teletypes, video display terminals, printing terminals, diskettes, fixed head discs, cartridge discs, disc pack drives, magnetic tape units, cassettes, paper tape readers, paper tape punches, line printers, character printers, plotters, card readers, and card punches. Communications hardware includes high-speed synchronous and asynchronous multiplexer systems, direct IBM 360/370 interfaces, and intercomputer connections. Input/output equipment includes A/D and D/A converters, digital I/O and general purpose interfaces.

TABLE 1-1.   POINT 4 MODELS and FEATURES

Features	Model			
	4/1	4/2	4/3	4/4
400ns CPU	X	X	X	X
32K words of memory	X		X	
64K words of memory		X		X
Self-Test Diagnostics	X	X	X	X
Virtual Front Panel	X	X	X	X
Chassis			X	X
Power Supply			X	X

1.2.4 POINT 4 Computer Options

The following options are available to enhance the POINT 4 computer system:

1. Main memory size is available in 32K words or 64K words, with 128K words available through purchase of dual processors.

2. Main memory parity error halt option assures data integrity by detecting data errors in memory transfers.

3. Main memory battery backup is available to maintain the contents of memory for at least two hours in the event of a power failure.

4. An operator control unit is available which may be attached to the CPU front panel or extended by cable to a convenient working surface. The control unit can be attached or removed without turning off the CPU.

5. A high-speed interprocessor bus allows two POINT 4 computers to load data into each other's main memory at a rate of 2 megawords per second. (Available 1st quarter of 1980).

6. A Programmable Control Store (PROM) with 100-nanosecond cycle time can be used to extend the instruction set to perform complex operations such as decimal arithmetic, string processing and stack operations at high speeds. Such extensions of the standard instruction set will be used to increase the speed of IRIS and application software. (Available 1st quarter 1980.)

7. Memory Mapping - (Information not yet available.)

1.2.5 Software Support

Educational Data Systems has available a full line of POINT 4
compatible software systems including the following:

- The Interactive Real-Time Information System (IRIS)
  Operating System and Business BASIC language translator
- Absolute and extended assemblers which accept (as a subset)
  the instruction set described in this manual
- INFOTRIEVE Data Base Management System
- INFORMAT Report Writing System
- Management, Accounting and Control System (MACS) business
  systems
- STYLUS multi-terminal Word Processing System
- READINET Project Control System, using PERT/CPM techniques
- HIPS (Hospitality Industry Processing System) business
  package for hotels, clubs and restaurants

POINT 4 systems also support a significant array of customer
developed software.

1.2.6 Customer Support

The POINT 4 system is backed by a one-year maintenance warranty.
If execution of the diagnostics in the self-test PROM indicates a
hardware problem, a replacement board will be supplied. In
addition to user and diagnostic manuals, both user and maintenance
training are available.

1.3 EQUIPMENT CHARACTERISTICS

1.3.1 Performance Characteristics

    Word Length: 16-bits
    General Purpose Accumulators: 4
    Memory Cycle Time: 400 nanoseconds
    RAM Access Time: 200 nanoseconds
    Microprogram Cycle Time: 100 nanoseconds
    Crystal-controlled Clock Rate: 30 MHz
    Memory options: 32K or 64K words
    Standard Data Channel:
      Input - 1100 nanoseconds
      Output - 1700 nanoseconds
    High-speed Data Channel:
      Input - 900 nanoseconds
      Output - 1300 nanoseconds
    Interrupt Response: 1200 nanoseconds

1.3.2 Equipment Specifications

ELECTRICAL
    AC Input - 105 to 125 VAC, 5 amps max, 47 to 63 Hz
        Optional - 210 to 250 VAC, 2.5 amps max, 47 to 63 Hz
    AC Power Consumption - 625 watts max

MECHANICAL
    Processor Chassis:
        Height - 5.25 inches
        Width - 19.0 inches
        Depth - 17.5 inches
        Weight - 30 pounds

     Operator Control Unit:
        Height - 3.62 inches
        Width - 6.12 inches
        Depth - 1.1 inch

    Power Supply Chassis:
        Height - 5.25 inches
        Width - 19.0 inches
        Depth - 9.06 inches
        Weight - 30 pounds

ENVIRONMENTAL
    Temperature Range - 0 to 55 degrees Celsius
    Relative Humidity - 10 to 90 percent noncondensing

```
                              ┌─────────────────────────────────────────────┐
   ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐  │                                             │
                              │  CENTRAL            MEMORY                   │
   │  OPTIONAL   ┌ ─ ─ ─ ─ ┐ ││  PROCESSOR         32K OR 64K                │
      PROCESSOR  │ MEMORY   │ │  LOGIC              WORDS                     │
   │             │ 32K OR 64K│ │                                             │
               │ WORDS    │ ││                                             │
   │             └ ─ ─ ─ ─ ┘ ││        HIGH-SPEED    PROGRAMMABLE            │
                              │ <HIP   INTER-        CONTROL                 │
   │                       │ │  BUS>  PROCESSOR     STORE                    │
   └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘  │        BUS CONTROL                           │
                              └─────────────────────────────────────────────┘
                                    I/O CONTROL & DATA LINES
```

CENTRAL
PROCESSOR
LOGIC

MEMORY
32K OR 64K
WORDS

OPTIONAL
PROCESSOR

MEMORY
32K OR 64K
WORDS

HIGH-SPEED
INTER-
PROCESSOR
BUS CONTROL

HIP
BUS

PROGRAMMABLE
CONTROL
STORE

I/O CONTROL & DATA LINES

TERMINAL

MIGHTY
MUX

REAL
TIME
CLOCK

TERMINAL

TERMINAL

TERMINAL

LINE
PRINTER

TAPE
CONTROLLER

DISC
CONTROLLER

MAGNETIC
TAPE
TRANSPORT

DISC
DRIVE

FIGURE 1-2.   TYPICAL CONFIGURATION OF THE POINT 4 COMPUTER

1.4 SYSTEM ARCHITECTURE

The POINT 4 architecture has been streamlined to create a system with a minimum of signal interfacing between boards and with maximum speed of instruction execution. A combination of central processor logic and up to 64K words of RAM on a single circuit board eliminates time delays due to long memory access bus paths. The CPU board contains all basic functions of the computer plus such features as the Interprocessor Bus and Programmable Control Store. This leaves only I/O controller boards to be added to the system. Figure 1-2 illustrates a typical configuration of a POINT 4 computer system.

1.4.1 System Functional Units

The POINT 4 computer is comprised of five functional units: the CPU and memory board, the processor chassis and front panel, the "Mini-panel", an optional Operator Control Unit, and the power supply. Figure 1-3 is a block diagram of the basic units of the computer system showing the functions they perform within the system.

1.4.1.1 Central Processor and Memory Board

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

- Four general purpose Accumulators plus 4 special purpose registers
- The Arithmetic/Logic Unit (ALU)
- The Instruction Register
- The Main Data Bus
- The Program Counter
- The Effective Address Register
- Timing Control
- Input/Output Control

In addition to these basic CPU elements, the CPU board contains a variety of features which are often omitted or housed on separate boards. The compact CPU board contains the following features:

- 32K or 64K words of Random Access Memory (RAM)
- Parity Generation and Error Detection Logic
- Operator Control Unit Interface Logic
- Battery Backup Interface Logic
- Programmable Control Store (PCS)
- APL PROM containing a program to implement the "Virtual" Control Panel features
- A Self-Test PROM containing a hardware checkout routine
- High-speed Interprocessor Bus control logic

Figure 1-4 is a block diagram of the CPU/memory board, showing logic to handle each of the above functions.

```
                    ┌──────────────────┐  ┌─────────────────────────────────────────┐   ┌──────────────────────────┐
                    │                  │  │  CENTRAL PROCESSOR BOARD INCLUDES       │   │ ACTIVE STATE    │ O      │
                    │                  │  │                                         │   │   MONITORS      │ P      │
                    │                  │  │  32 OR 64K WORDS OF MEMORY PLUS PARITY   │   ├─────────────────┤ E      │
                    │  I/O DEVICE      │  │  ACCUMULATORS (4)                       │   │ ADDRESS & DATA  │ R      │
                    │  CONTROLLERS     │  │  ARITHMETIC/LOGIC UNIT (ALU)            │   │   DISPLAYS      │ A      │
                    │                  │  │  INSTRUCTION REGISTER                    │   ├─────────────────┤ T      │
                    │                  │  │  PROGRAM COUNTER                         │   │                 │ O      │
                    │                  │  │  EFFECTIVE ADDRESS REGISTER              │   │ MEMORY ACCESS   │ R      │
                    │                  │  │  TIMING AND CONTROL LOGIC                │   │   CONTROLS      │        │
                    │                  │  │  I/O CONTROL                             │   ├─────────────────┤ C      │
```

I/O DEVICE CONTROLLERS

CONTROL SIGNALS

16-BIT DATA BUS DMA & PIO

SECOND PROCESSOR (OPTIONAL)

HIP BUS

CENTRAL PROCESSOR BOARD INCLUDES

32 OR 64K WORDS OF MEMORY PLUS PARITY
ACCUMULATORS (4)
ARITHMETIC/LOGIC UNIT (ALU)
INSTRUCTION REGISTER
PROGRAM COUNTER
EFFECTIVE ADDRESS REGISTER
TIMING AND CONTROL LOGIC
I/O CONTROL
MICROPROGRAM SEQUENCER
MAIN DATA BUS
HIGH SPEED INTERPROCESSOR CONTROL
    LOGIC & BUS
PROGRAMMABLE CONTROL STORE
OPERATOR CONTROL UNIT LOGIC
APL PROMS (MANIP & SELF TEST)
BATTERY BACKUP LOGIC

OPERATOR CONTROL UNIT

ACTIVE STATE MONITORS
ADDRESS & DATA DISPLAYS
MEMORY ACCESS CONTROLS
ACCUMULATOR ACCESS CONTROLS
PROGRAM EXECUTION CONTROLS
APL CONTROLS
DATA ENTRY

RUN MONITOR	PARITY ERROR & CARRY MONITOR	STOP & CONTINUE SWITCHES	AUTO PROGRAM LOAD CONTROL	BATTERY MONITOR & SWITCH

FRONT PANEL

AC POWER   POWER ON CONTROL

POWER SUPPLY MODULE
+5   +12
±15   -5

BATTERY BACKUP MODULE

FIGURE 1-3.   POINT 4 COMPUTER SYSTEM BLOCK DIAGRAM

CPU/Memory Board Block Diagram
October 1979
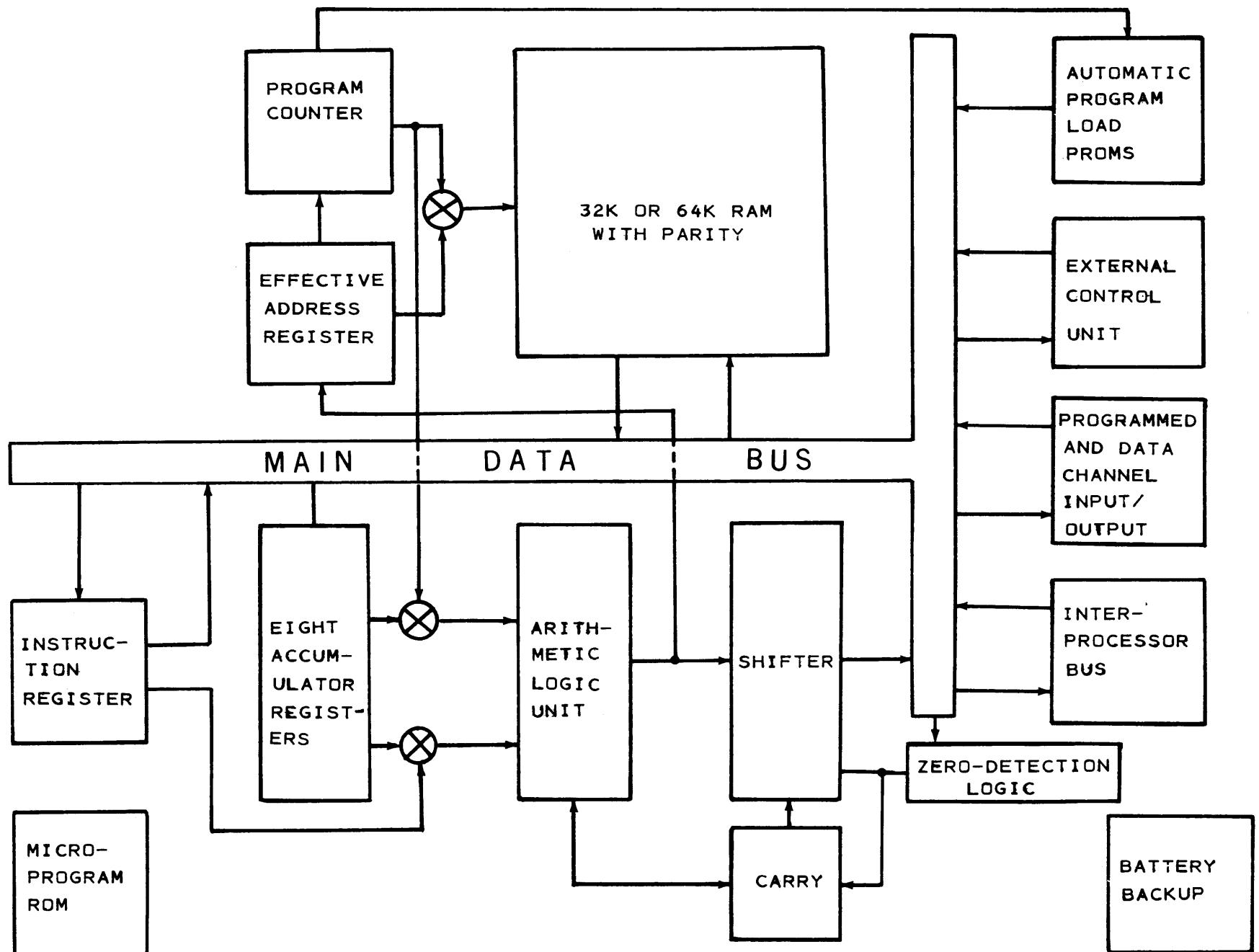
FIGURE 1-4.   POINT 4 CPU/MEMORY BLOCK DIAGRAM

1.4.1.2 Processor Chassis and Front Panel

The POINT 4 processor chassis is designed to be mounted in a
standard 19-inch equipment rack. The chassis contains seven slots
spaced on .6-inch centers. The processor chassis is 5.25 inches
high, 19 inches wide and 17.5 inches deep.

Cooling is provided by two whisper-quiet fans. These are mounted
on the left side of the chassis behind the "Mini-panel".

The front panel snaps on and off. There are no screws or hinges
holding it in place. No cabling exists between the front panel
and the chassis since the "Mini-panel" is mounted directly onto
the chassis and its controls are accessible through a slot on the
left side of the front panel.

1.4.1.3 Processor "Mini-Panel"

The "Mini-Panel" on the POINT 4 chassis houses an efficient set of
controls and indicators for basic processor operation. Mini-panel
indicators include light emitting diode indicators for monitoring
of parity errors, the carry flag, CPU operation, power OK and
battery OK. Processor control switches include a four-position
switch controlling ON, AUTO operation, STANDBY and OFF. Program
control switches allow stopping and restarting of program
execution and enabling of the Virtual Control Panel. For further
details see Section 5.3, on Mini-panel operations.

The Virtual Control Panel allows monitoring and control of the
processor from a master terminal through use of the manipulator
program "MANIP". For further details see Section 5.5 on use of
the Virtual Control Panel.

1.4.1.4 Operator Control Unit (Optional)

In addition to the basic controls and indicators on the POINT 4
processor chassis, an Operator Control Unit is available which
enhances operator access to the processor. This detachable
control unit can be attached to the front panel of the POINT 4
chassis or can be extended via ribbon cable to any convenient
working surface. The compact control unit contains all switches
and indicators necessary to monitor and control the processor.
Figure 1-5 is a photograph of the Operator Control Unit.

Displays include two octal displays for address and data, and
eight light emitting diodes indicating activity in the following
areas: data channel, programmable control store, high-speed
interprocessor bus, 64K word addressing, program execution,
interrupts enabled, carry condition, and parity error detected.
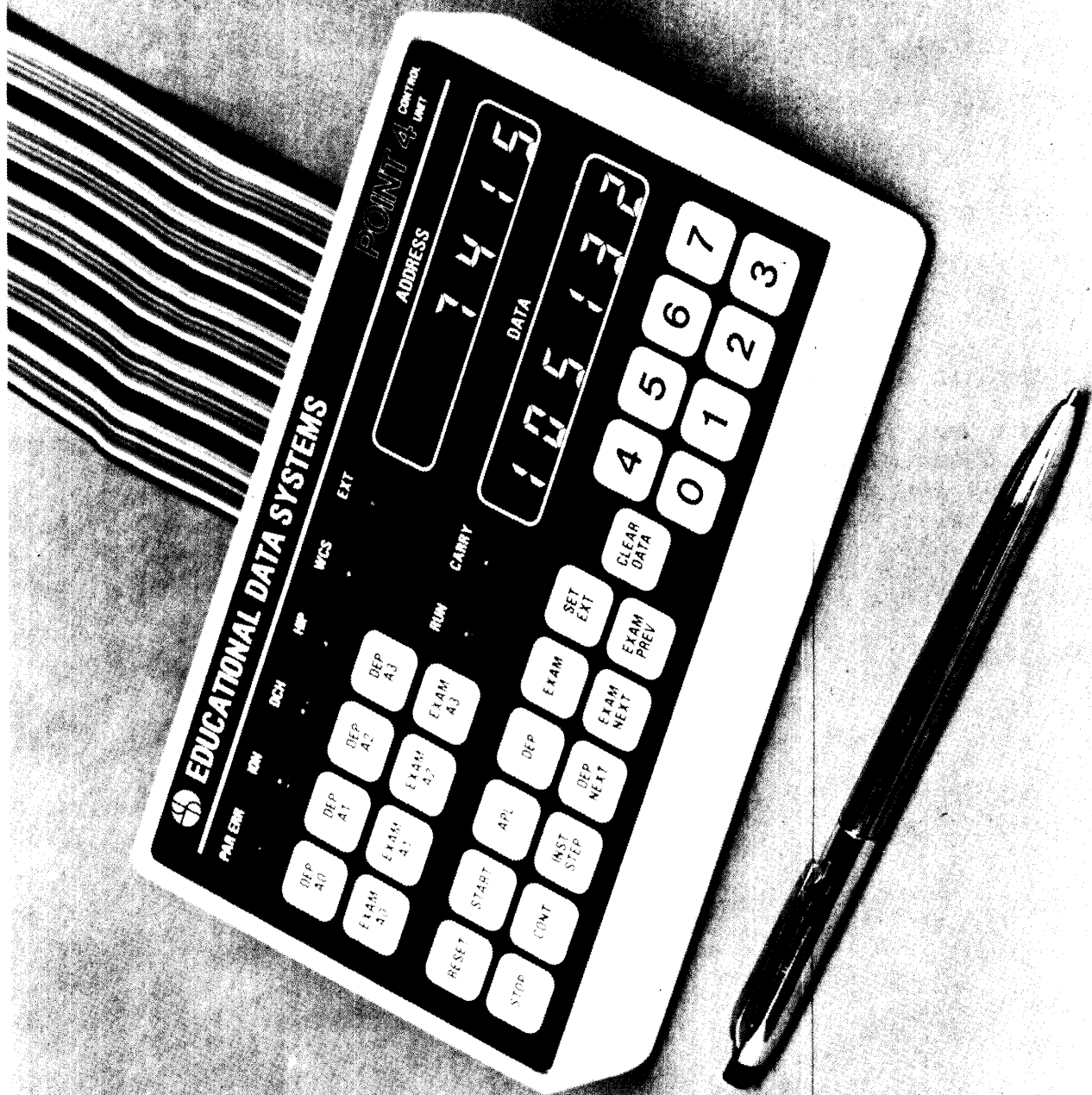
FIGURE 1-5. POINT 4 OPERATOR CONTROL UNIT

Operator Control Unit
October 1979

Sealed membrane switches provide an octal data entry panel and a clear data button, as well as the following controls:

- Memory is accessible through control switches to examine and deposit in memory, as well as to enable 64K word addressing
- Accumulators are accessible through examine and deposit controls
- Program execution controls include reset, start, stop, continue and instruction step
- An Automatic Program Load (APL) switch

See Section 5.4 for further description of the Operator Control Unit.

1.4.1.5 Power Supply Module

The power supply module is housed in a separate chassis (19 inches wide, 5.25 inches high, and 8.06 inches deep) for flexibility of installation. The power supply consists of two units: a regular power supply for normal use, and a backup battery unit (optional) which is used in case of power failure.

The regular power supply module delivers the power required for the CPU board and front panel logic, the operator control unit, plus the power required for I/O device controllers housed in the POINT 4 chassis.

The power supply requires an input voltage of 110 VAC or 220 VAC, 47 to 400 Hz. Power supply output voltages are:

Voltage	Supplied to
+ 5V	CPU board, Operator Control Unit, I/O Controllers
- 5V	CPU, I/O Controllers
+15V	For use by the I/O Controllers
-15V	For use by the I/O Controllers

These voltages are available to the user for I/O controller applications. Output current available for each voltage and supply type are:

Voltage	Output	Power Supply Type
+ 5 VDC	35.0 amps	Switching supply
- 5 VDC	1.2 amps	Linear regulated supply
+15 VDC	3.0 amps	Linear regulated supply
-15 VDC	3.0 amps	Linear regulated supply

The battery backup option protects memory contents in the case of a power failure. Memory contents will be maintained for at least two hours after such failure. The battery unit is maintained in a charged state by the power supply as long as the unit is plugged in and AC power is available (even if the key switch is in the OFF position). Battery backup voltages are:

Voltage	Supplied to
+ 5V BU	CPU Board, Mini Panel, Memory Refresh logic
- 5V BU	Memory on CPU Board
+12V BU	Memory on CPU Board

These outputs are supplied to the CPU board only. They are low current outputs.

The power supply unit converts the AC line voltage to a set of regulated DC output voltages for use by the CPU and controllers. The input voltage is passed through a filter, the output of which drives the fan module, the battery backup unit (if installed), and a solid state relay. Refer to Figure 1-6 for a block diagram of the power supply unit.

Battery backup interface logic includes a sensor to determine whether the power control cable is connected to the CPU chassis. The power supply unit will not supply power to the processor if this cable is disconnected. The processor may also be nonfunctional if the cable is connected but the CPU mini-panel key switch is set to the OFF position, disabling all power to the system.
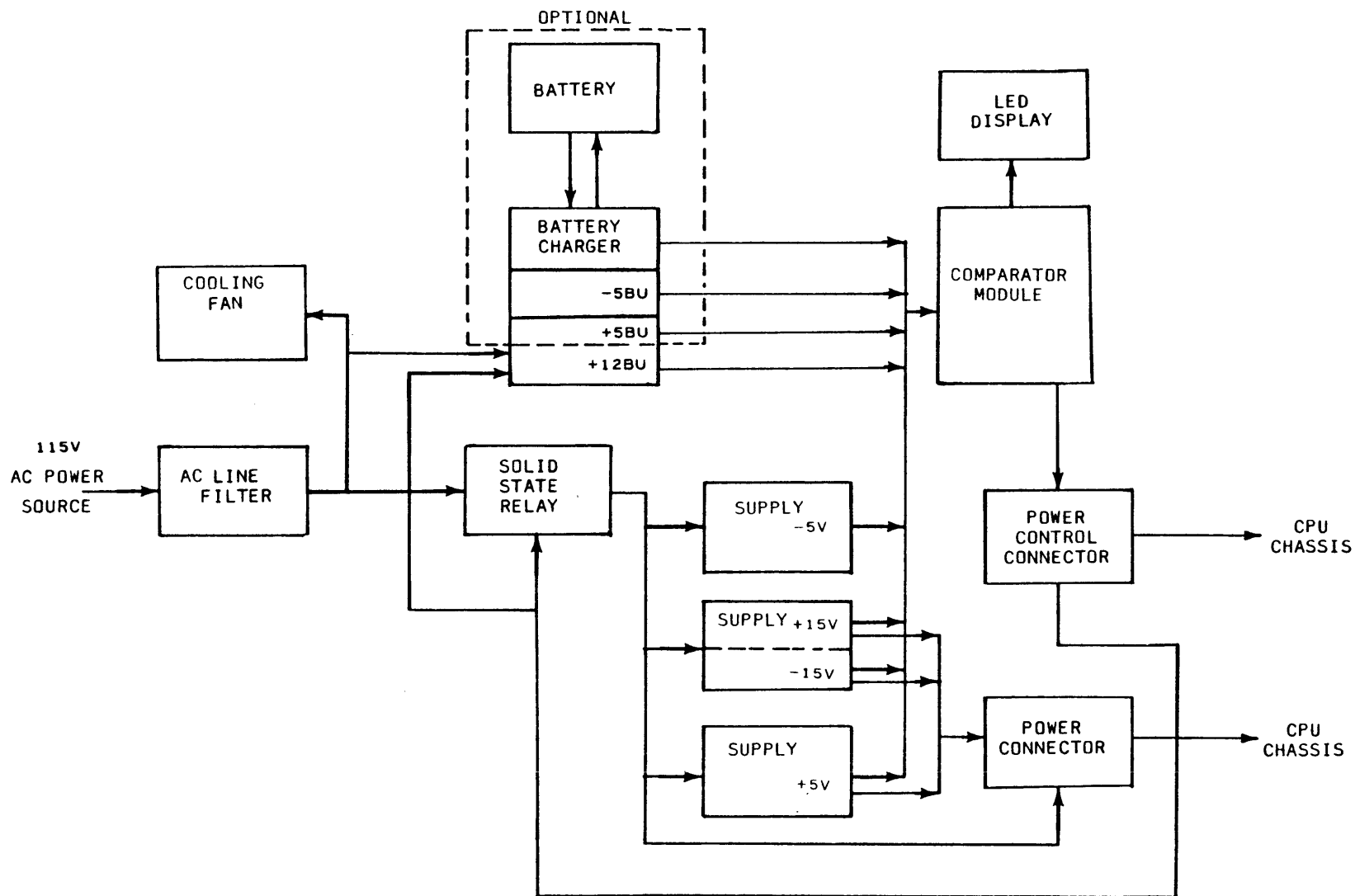
FIGURE 1-6.   POINT 4 POWER SUPPLY BLOCK DIAGRAM

If the mini-panel key switch is set to any position other than OFF, power will be supplied to the system. From this point on operation of the unit depends upon the presence or lack of the battery backup unit:

- If the battery backup unit is not present, a control signal will be sent to the relay, which enables the +5V, -5V, +15V and -15V voltage supplies for CPU and user applications. The +12V BU (backup) supply is operational even without the backup unit. The CPU always receives the +12 voltage from this supply. The ±5V voltages for the CPU are drawn from the ±5V user supplies. The ON and AUTO switch positions will cause the +5V, -5V, +15V and -15V supplies to be enabled for user applications. See Section 5.3 for Processor Mini-panel switch positions.

- When the battery backup is installed, the relay receives its control from the CPU Mini-panel via the power control cable. Under normal operation (no power-fail) the relay enables the +5V, -5V, +15V and -15V supplies for CPU and user applications whenever the mini-panel switch is in either the ON or the ..UTO position. The +12V BU (back-up), +5V BU, and -5V BU supplies are operational whenever the mini-panel switch is not in the OFF position, and they supply power to the memory on the CPU board. (See Section 5.3 for Processor Mini-Panel switch positions.) In case of AC power failure, a control signal informs the CPU of the Power Fail condition before the power is out of tolerance. The +5V BU, -5V BU and +12V BU take over supply of power to the CPU Memory only. No voltages are available for user applications.

Logic on the comparator board tests whether output voltages are in tolerance. If the voltage is in tolerance the light emitting diode on the power supply mini-panel for the specified voltage is illuminated. If all voltage outputs are in tolerance, a signal is sent to the CPU mini-panel which turns on the POWER OK light emitting diode on that panel. The BTRY OK light emitting diode on the CPU mini-panel illuminates when the batteries (if installed) are fully charged, or are being used as a power source (i.e. AC power failure).

1.4.2 Data Channel

In order to handle data transfers between input/output devices and memory via program control, an interrupt plus the execution of several instructions is required for each word transferred. To allow greater transfer rates between memory and external devices, the processor is equipped with a data channel through which a device, at its own request, can access memory directly using a minimum of processor time. A high-speed device, such as a disc, tape or storage unit can thus access memory without assistance from the program. Program execution simply pauses momentarily while a data channel transfer takes place on the I/O Bus.

The data channel releases processor time by allowing execution of a program concurrently with data transfers for a device. Many devices may share the data channel.

1.4.2.1 Data Channel Options

The POINT 4 has two jumper selectable data channel speed options. They are:

>   Standard Data Channel
>   Input - 1100 nanoseconds
>   Output - 1700 nanoseconds
>
>   High-Speed Data Channel
>   Input   - 900 nanoseconds
>   Output - 1300 nanoseconds

Choice of standard or high-speed data channel depends on the ability of all peripherals to respond within the maximum time allowed. See Section 6.4 for data channel jumpering instructions.

SECTION II

INSTRUCTION REPERTOIRE

2.1 INTRODUCTION

This section explains the function and use of POINT 4
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.

2.2 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.

```
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
┌──┬───────────┬───────────┬───────────┬───────────┐
│  │           │           │           │           │
└──┴───────────┴───────────┴───────────┴───────────┘
```
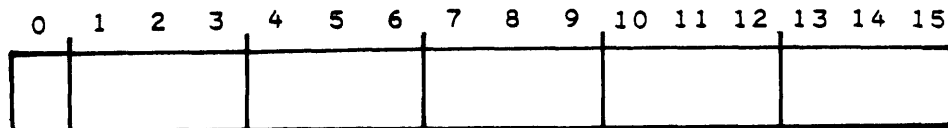
POINT 4 16-BIT BINARY WORD FORMAT

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	0000000000010000	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. Therefore, a review of two's complement arithmetic is necessary.

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

2.3 INSTRUCTION TYPES

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

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

a) Memory Reference Instructions (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

b) Arithmetic/Logic Instructions (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).

c) Input/Output Instructions (instructions for input/output operations with a specific peripheral device).

Figure 2-1 is an overview of the formats for each type of instruction. Each of these three classes is discussed in detail in the succeeding subsections.

		0	1	2	3	4	5	6	7	8	9	10 11 12	13 14 15
MEM. REF.	JMP	0	0	0	0	0	INDIRECT	INDEX				DISPLACEMENT	
	JSR	0	0	0	0	1							
	ISZ	0	0	0	1	0							
	DSZ	0	0	0	1	1							
	LDA	0	0	1	AC								
	STA	0	1	0	AC								
I/O		0	1	1	AC		OPCODE			CTRL		DEVICE CODE	
A/L		1	ACS		AC		OPCODE			SH		CY	NL	SK

```
  AC = Accumulator
CTRL = Control pulse
 ACS = Source accumulator
 ACD = Destination accumulator
  SH = Shift control
  CY = Carry preselection
  NL = No-load
  SK = Skip condition
```

FIGURE 2-1. POINT 4 INSTRUCTION FORMAT SUMMARY

2.4 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:

    a)   Move Data Instructions: LDA, STA
    b)   Jump Instructions: JMP, JSR
    c)   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.

2.4.1 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.

All memory reference instructions use the same binary format:

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

┌───────────────┬─────────┬───┬────────┬─────────────────────────┐
│               │ OP CODE │   │ INDEX  │                         │
│    OP CODE    │   OR    │ I │        │      DISPLACEMENT        │
│               │   AC    │   │ MODE   │                         │
└───────────────┴─────────┴───┴────────┴─────────────────────────┘

                           I = INDIRECT
```
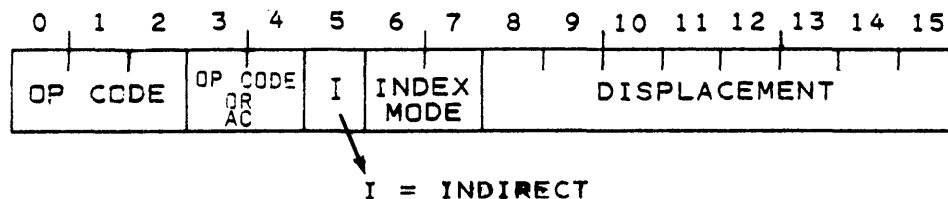
MEMORY REFERENCE INSTRUCTION FORMAT

Bit 5 of the instruction word is the indirect, or I field; bits 6
and 7 are the index, or X field, and bits 8-15 are the
displacement, or D field.

  • The I field determines whether the X and D fields specify the
    effective address (E) directly or whether indirect addressing
    is to be used.

  • The X field defines one of four addressing modes. Each
    addressing mode may be thought of as a "page" of 256 words
    which the instruction can address directly.

  • The D field specifies the specific word addressed on the
    selected page.

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.

2.4.1.1 Indexing Mode

The X field selects one of the following 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.

2.4.1.2 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.

This address word can be interpreted as follows:

- If the most significant bit of the address word equals 0, this address word is the effective address, E.

- If the most significant bit of the address word equals 1, the action taken depends on whether the processor is set in 32K or 64K addressing mode, as described below.

When the processor is in 32K (normal) mode, a second level of indirect addressing is allowed. In this case, if the most significant bit (bit 0) of the address word fetched from memory contains a 1, another level of indirect addressing is required, and the address word in the Effective Address Register specifies the address word to be fetched from memory. The process continues until an address word with bit 0 = 0 is found. Through programming error it is possible to become caught in an infinite loop of indirect addressing. Caution should be taken when using indirect addressing to avoid this problem.

When the processor has 64K memory and 64K addressing is enabled, a second level of indirect addressing is not permitted. In this case all 16 bits of the word fetched from memory are used as the effective address. A 1 in bit 0 of the address word simply indicates an address in the upper 32K words (100000 - 177777) of memory.

Automatic Incrementing and Decrementing of Locations

If at any time during the indirect addressing sequence, the Effective Address Register contains an address in the range from 000020 octal to 000037 octal, the following auto-indexing action is performed:

1. The contents of the memory location specified by the Effective Address Register are fetched, and the contents are either incremented or decremented by one, as follows:

   - If the address is in the range 000020 octal through 000027 octal, the contents are incremented.
   - If the address is in the range 000030 octal through 000037 octal, the contents are decremented.

2. The incremented or decremented value is written back into the same memory location from which the value was fetched in step 1.

3. The incremented or decremented value produced in step 1 is stored in the Effective Address Register and used for the next level of indirect addressing (if bit 0 = 1 and the processor is set to 32K addressing mode), or for the effective address (if bit 0 = 0 or the processor is in 64K addressing mode).

NOTE:

The value of bit 0 _after_ incrementing or decrementing takes place controls continuation of indirect addressing.

2.4.2 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 2.4.1 for basic memory reference instruction formats and field definitions. Refer to Appendix A, Von Neumann Map of Point 4 Command Structure, for octal formats of each instruction and to Appendix B, POINT 4 Instruction Reference Chart, for octal to symbolic conversion of memory reference instructions.

2.4.2.1 Move Data Instructions

When the effective address, E, is in the Effective Address Register, one of the following two operations is performed, depending on the code in bits 1 and 2 of the OPCODE field:

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.

2.4.2.2 Jump and Modify Memory Instructions

When the effective address, E, is in the Effective Address
Register, one of the following four operations is performed if
bits 1 and 2 are both 0. The operation is determined by the code
in bits 3 and 4 of the OPCODE field:

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 effective address E are fetched, 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.

2.4.2.3 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 2-1 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 the indirect addressing indicator, the displacement and the indexing indicator). The assembly language instruction will be formatted as follows:

```
        +-------+-------+-----+
        | D S Z | 4 5,  |  2  |
        +-------+-------+-----+
           /       |       \
       OPCODE   I and D   X Field
                Fields
```
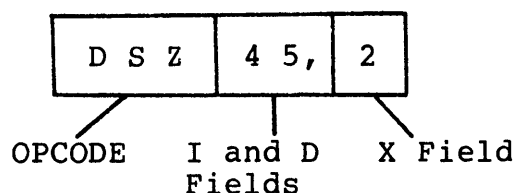
The move data instructions, LDA and STA, also require that an accumulator (A0 - A3) be specified. For example:

```
        +-------+-----+---------+-----+
        | L D A | 2,  | - 4 5,  |  3  |
        +-------+-----+---------+-----+
           /       |       |       \
       OPCODE     AC    I and D   X Field
                        Fields
```
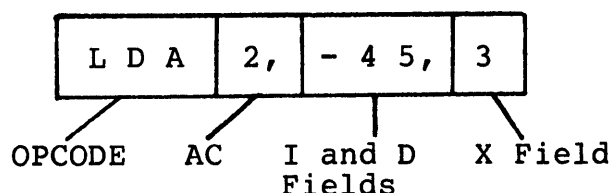
Fields that are not specified will be assembled containing 0's. An "@" symbol denotes indirect addressing and places a 1 in bit 5 of the instruction. For example:

        LDA   1, @ 20

specifies indirect, page-zero (X Field = 00) addressing.

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.

Memory Reference Instructions
October 1979

TABLE 2-1. ASSEMBLER LANGUAGE CONVENTIONS
FOR MEMORY REFERENCE INSTRUCTIONS

| Instruction Function	OP CODE Mnemonic	Separator Space or Tab	Accumulator Number	Memory Address			
				I(Indirect)	D(Displ.)	,	X(Index)
Load Accumulator	LDA		a*				
Store Accumulator	STA						blank
Jump	JMP			blank	Displace-		1**
Jump Subroutine	JSR			@	ment	,	2
Increment and Skip if Zero	ISZ		None				3
Decrement and Skip if Zero	DSZ						

* a = 0, 1, 2, 3 representing A0, A1, A2, A3
** Instead of "displacement,1" the following sequence may be used: ".± displacement"

2.5 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 ALU function specified by bits 5-7, as shown below:

0	1 2	3 4	5 6 7	8 9	10 11	12	13 14 15
1	ACS	ACD	OP CODE	SH	CY	NL	SK
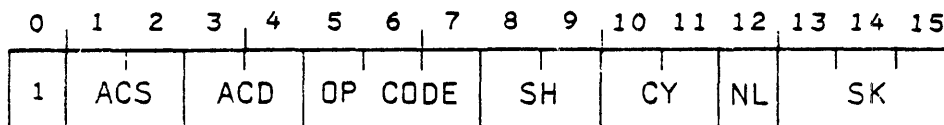
ARITHMETIC/LOGICAL INSTRUCTION FORMAT

2.5.1 Arithmetic and Logical Processing

Before describing the eight arithmetic and logical instructions and their auxiliary control fields, it is necessary to describe the organization of the arithmetic/logical processing unit. From the programmer's point of view, the arithmetic/logical processing subsystem is organized as shown in Figure 2-2 and as described in the following subsections.

2.5.1.1 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 2.5.1.2 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.
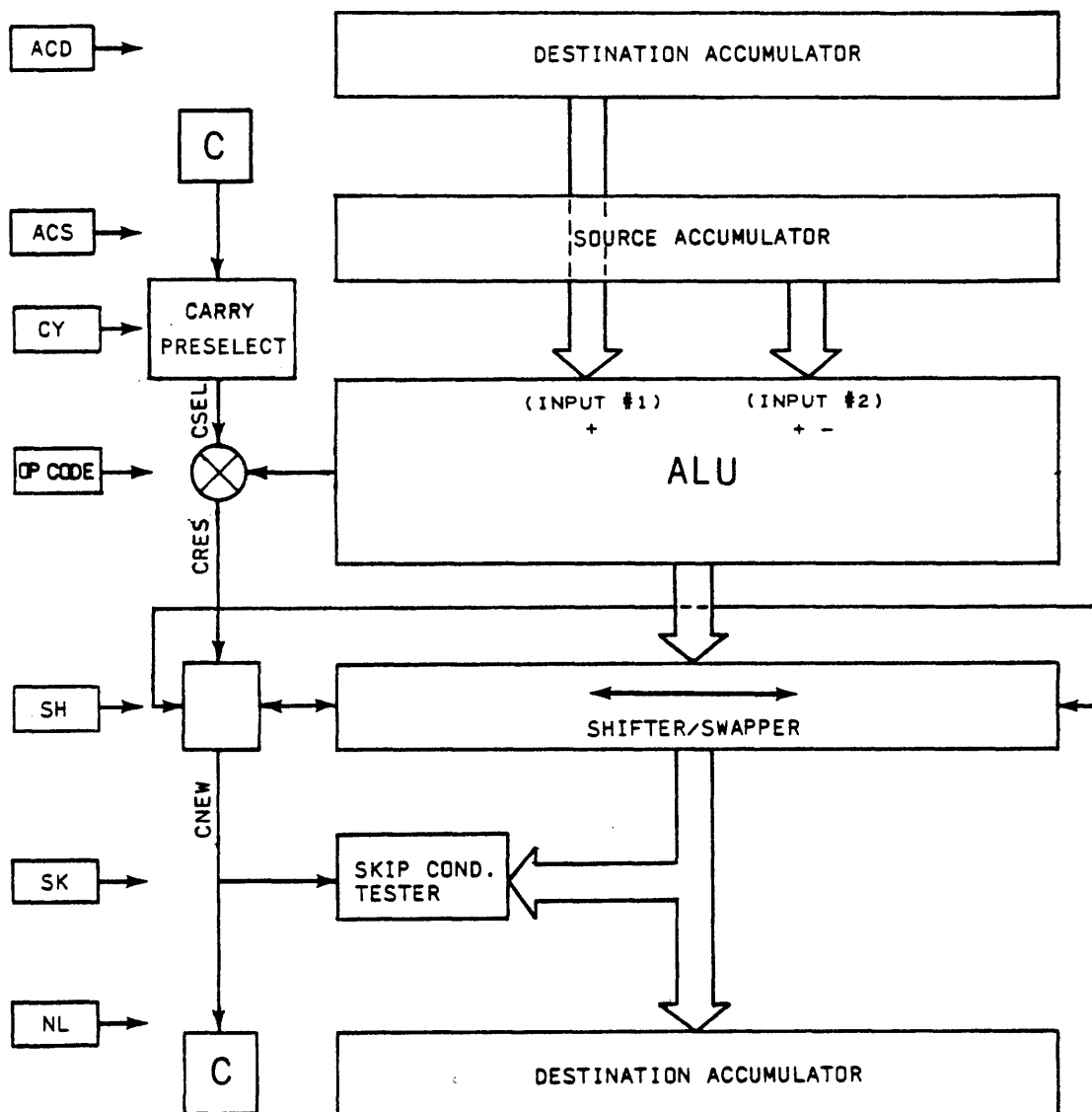
FIGURE 2-2. ARITHMETIC/LOGICAL OPERATIONS

2.5.1.2 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 (in ALU)	Unsigned Interpretation Octal	Dec.	Signed Interpretation Octal	Dec.
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.

There is a relationship between underflow/overflow and the carry-out from the ALU MSB, but the relationship is not a simple one, as 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), being 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 2-3.
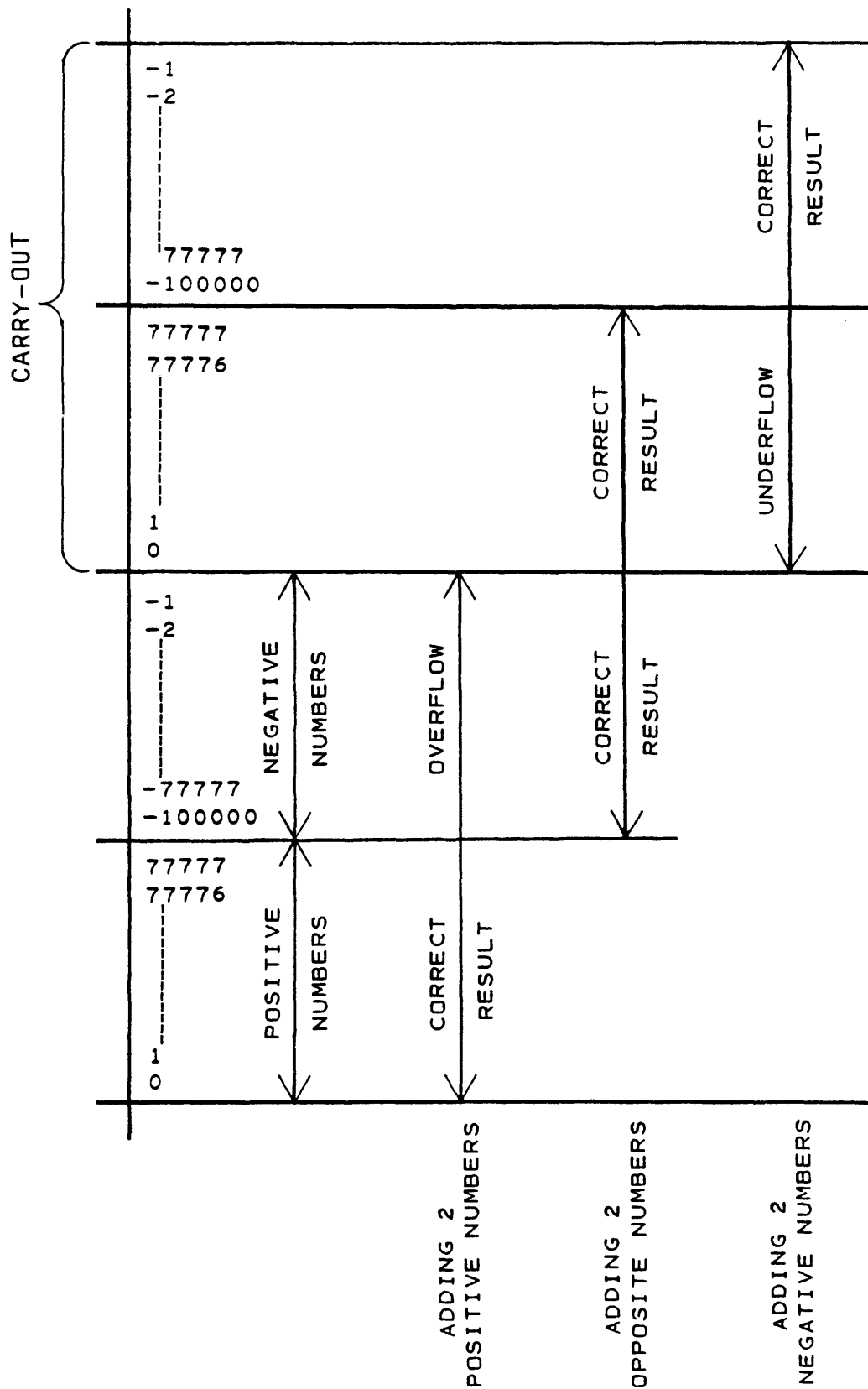
FIGURE 2-3.   OVERFLOW AND CARRY OPERATIONS ANALYSIS FOR SIGNED INTEGERS

2.5.2 Arithmetic/Logic Functions

The OPCODE field (bits 5 through 7) defines one of the following eight arithmetic/logic operations to be performed by the 16-bit ALU:

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.

2.5.3 Secondary Functions

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

2.5.3.1 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 follows:

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 8 MSBs of the result with the eight LSBs. The carry resulting from the ALU is unaffected.

2.5.3.2 Carry Control Field (CY)

The CY field (bits 10 and 11) specifies the base to be supplied to the ALU for carry calculation, as follows:

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.

2.5.3.3 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.

2.5.3.4 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 as follows:

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 non-zero
100	SZR	Skip if result is zero
101	SNR	Skip if result is non-zero
110	SEZ	Skip if either carry bit or result is zero
111	SBN	Skip if both carry bit and result are non-zero

2.5.4 Assembler Language Conventions and Examples

The assembly 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 2-2 shows the programming
conventions for Arithmetic and Logical Instructions.

The format is as follows:

```
 ┌───────┬─┬─┬─┬────┬────┬───────┐
 │ A D D │C│L│#│ 0, │ 1, │ S Z C │
 └───────┴─┴─┴─┴────┴────┴───────┘
```

```
Instruction                    ACS    ACD    SK (Skip)
  Mnemonic                                      Field
            CY            NL
         (Carry)    SH  (No-Load)
          Field   (Shift)  Field
                   Field
```
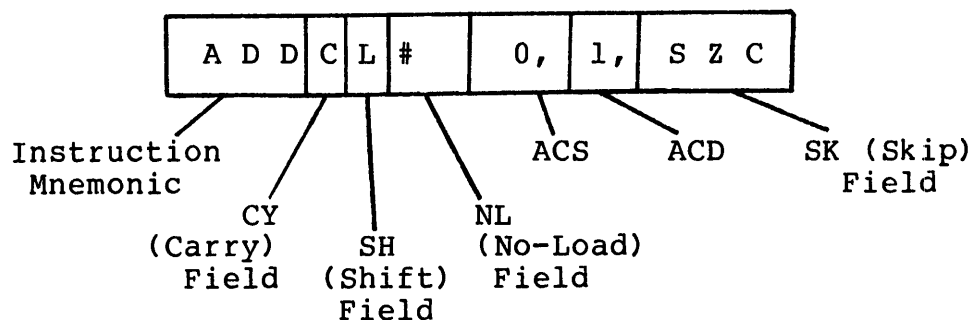
The CY, SH, NL, and SK fields are specified by adding the
appropriate mnemonic symbols. None of these four fields has to be
specified, but their symbols must appear in the proper order and
place if they are included. Those fields not specified will be
assembled containing 0's. For example:

        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 into A1 and the Carry flag.

2-23

TABLE 2-2.   ASSEMBLER LANGUAGE CONVENTIONS FOR
ARITHMETIC AND LOGICAL INSTRUCTIONS

Instruction Function	OP CODE Mnemonic	Optional Secondary Functions			Separator Space or Tab	Accumulators				Optional 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.

2.6 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 following format:

```
 0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15
┌───────────┬───────┬───────────┬───────┬───────────────────────┐
│ 0   1   1 │  AC   │  OP  CODE │ CTRL  │      DEVICE  CODE      │
└───────────┴───────┴───────────┴───────┴───────────────────────┘
```
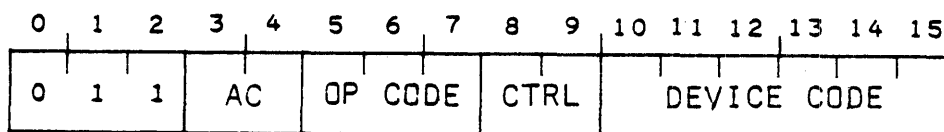
INPUT/OUTPUT INSTRUCTION FORMAT

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.

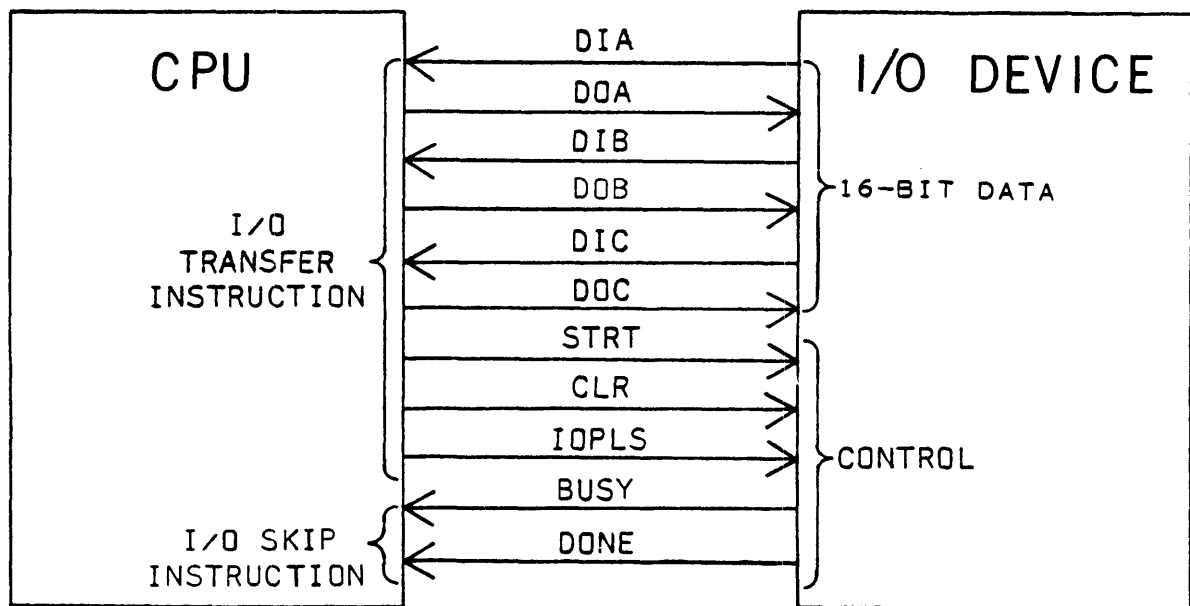### 2.6.1 Regular I/O Instructions

Regular I/O instructions apply to all device codes except code 77. These instructions fall into two basic categories, depending on whether they transfer data or test the state of the device.

    1. I/O Transfer: NIO, DIA, DOA, DIB, DOB, DIC, DOC

    2. I/O Skip: SKPBN, SKPBZ, SKPDN, SKPDZ

The POINT 4 input/output system provides for the following functions to be specified:

- Full 16-bit data transfer:
  - Three input channels (DIA, DIB, DIC)
  - Three output channels (DOA, DOB, DOC)

- Control functions:
  - Three outgoing control pulses (START, CLEAR, or IOPULSE - designated by S, C or P, respectively).
  - Two device flags (BUSY and DONE) that can be sensed by I/O Skip Instructions

These input/output functions are illustrated in the diagram below:

```
 _____                              _____
|                   |          DIA               |                   |
|   C P U           |<────────────────           |  I/O  DEVICE      |
|                   |          DOA               |                   |
|                   |────────────────>           |                   |
|                   |          DIB               |                   |
|                   |<────────────────           |                   |
|                   |          DOB               |                   |
|    I/O            |────────────────>           |  16-BIT DATA      |
|   TRANSFER        |          DIC               |                   |
|  INSTRUCTION      |<────────────────           |                   |
|                   |          DOC               |                   |
|                   |────────────────>           |                   |
|                   |          STRT              |                   |
|                   |────────────────>           |                   |
|                   |          CLR               |                   |
|                   |────────────────>           |                   |
|                   |          IOPLS             |  CONTROL          |
|                   |────────────────>           |                   |
|    I/O  SKIP      |          BUSY              |                   |
|                   |<────────────────           |                   |
|  INSTRUCTION      |          DONE              |                   |
|                   |<────────────────           |                   |
|_____|                            |_____|
```

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.

The Control (CTRL) field can have two different functions, depending on the category into which the instruction falls:

- In conjunction with a Data Transfer Instruction one of the three different Control pulses may be sent to the device - START, CLEAR, or IOPULSE

- In conjunction with I/O Skip Instructions, the control field determines which of the two flags in the I/O device will be tested - BUSY or DONE

2.6.1.1 I/O Transfer and Device Control Instructions

I/O Transfer instructions move data between the processor and the
device interface. There are six possible device buffers, labeled
A, B and C. Each label may refer to two separate buffers - an
input (read) and an output (write) buffer.

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 (if any). The
type of transfer is determined by the code in the OPCODE field as
follows:

Bits 5 - 7	OPCODE	Definition
000	NIO	No data transfer involved. Device control only.
001	DIA	Move the contents of the A buffer in device D to the accumulator specified in the AC field, and send the control pulse specified by the Control field to the selected device.
010	DOA	Move the contents of the accumulator specified in the AC field to the A buffer in device D, and send the control pulse specified by the Control field to the selected device. The original contents of AC are unaffected.
011	DIB	Move the contents of the B buffer in device D to the accumulator specified in the AC field, and send the control pulse specified by the Control field to the selected device.
100	DOB	Move the contents of the accumulator specified in AC to the B buffer in device D, and send the control pulse specified by the Control field to device D. The original contents of AC are unaffected.
101	DIC	Move the contents of the C buffer in device D to the accumulator AC, and send the control pulse specified by the Control field to device D.
110	DOC	Move the contents of the accumulator specified in AC to the C buffer in device D, and send the control pulse specifiied by the Control field to device D. The original contents of AC are unaffected.

The Control field (bits 8 and 9) for I/O Transfer Instructions determines which control pulse should be transmitted, if any. The processor first performs the data transfer and then outputs the pulse. The Control field is defined for regular I/O transfer instructions as follows:

OPCODE Bits 5-7	Control Bits 8-9	Control Mnemonic	Definition
000-110	00	None	None
000-110	01	S	Produce the STRT pulse. Typically this starts the device by clearing its DONE flag, setting its BUSY flag, and clearing its interrupt request flag.
000-110	10	C	Produce the CLR pulse. Typically this clears both the BUSY and DONE flags, and the interrupt request flag, idling the device.
000-110	11	P	Pulse the special I/O bus control line (IOPLS). The effect, if any, depends upon the device.

2.6.1.2 I/O Skip Instructions

When the OPCODE field (bits 5-7) contains 111, the Control field (bits 8 & 9) selects the flag to be tested in the conditional I/O skip. The control codes for skip operations are:

OPCODE Bits 5-7	Control Bits 8-9	Control Mnemonic	Definition
111	00	SKPBN	Skip the next instruction if the BUSY flag in the device is nonzero.
111	01	SKPBZ	Skip the next instruction if the BUSY flag in the device is zero.
111	10	SKPDN	Skip the next instruction if the DONE flag in the device is nonzero.
111	11	SKPDZ	Skip the next instruction if the DONE flag in the device is zero.

2.6.1.3 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 2-3 shows the programming conventions for Regular Input/Output Instructions.

For Example:

```
 ┌───────┬───┬─────┬─────┐
 │ D I A │ C │ 2,  │ 1 0 │
 └───────┴───┴─────┴─────┘
```

Instruction Mnemonic
Control Function
Accumulator
Device Code
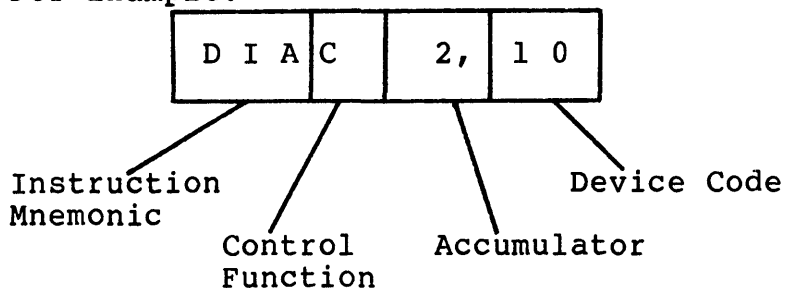
This instruction performs the function: Move the data from register A of device 10 into A2. Clear (reset) the device.

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

        D I A C   2, T T I

is equivalent to the previous example because "TTI" represents device 10 (input buffer for a teletype or CRT terminal).

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

        N I O S   T T I

        S K P B Z   T T I

TABLE 2-3.   ASSEMBLY LANGUAGE CONVENTIONS
FOR INPUT/OUTPUT INSTRUCTIONS

Instruction Function	OP CODE Mnemonic	Optional Device Function	Separator Space or Tab	Accumulator	,	Device Code
No Input/Output	NIO			none	none	
Data In Buffer A	DIA					
Data Out Buffer A	DOA	S (Start)*		0		
Data In Buffer B	DIB	C (Clear)*		1		
Data Out Buffer B	DOB	P (General Pulse)*		2	,	
Data In Buffer C	DIC			3		
Data Out Buffer C	DOC					00-76 octal
Skip if Busy Flag is Non-Zero	SKPBN					
Skip if Busy Flag is Zero	SKPBZ					
Skip if Done Flag is Non-Zero	SKPDN					
Skip if Done Flag is Zero	SKPDZ					
* Optional						

2.6.2 Special Code 77 (CPU) Instructions

Certain system functions, setting and testing of processor flags 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 device and the device code mnemonic is CPU.

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

OPCODE Field:

- Channel A (DIA) is used to read the mini-switches at the front edge of the CPU board, or the DATA display on the optional Operator Control Unit. DOA is not defined.

- Channel B addresses all I/O devices simultaneously for certain interrupt control functions.

- Channel C does no data transfer. DIC and DOC are used for resetting all I/O devices and for halting the computer.

Control pulses:

- S and C are used to enable or disable interrupts

- P is used to set 32K or 64K addressing mode

See the chart of special I/O instructions for details of instruction functions.

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

- BUSY = ION set (Interrupts are enabled)

- DONE = Power-failure has been detected (will cause interrupt if ION set)

The assembler also recognizes several special mnemonics for CPU instructions.

The regular instruction mnemonic and the special mnemonic are listed below along with a description of the special function of the CPU 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.
NIOP CPU	None	Set 32K or 64K addressing mode, depending on the least significant bit of A0:

<table>
<tr><td><u>Value of LSB</u></td><td><u>Addressing Mode</u></td></tr>
<tr><td>0</td><td>32K Mode</td></tr>
<tr><td>1</td><td>64K Mode</td></tr>
</table>

Instruction	Special Mnemonic	Definition
DIA a,CPU	READS a	Read the setting of the mini-switches at the front edge of the CPU board (or the value in the DATA readout of the optional Operator Control Unit, if installed) into accumulator a.
DIB a,CPU	INTA a	Read the device code of the highest priority device requesting an interrupt into accumulator a.
DOB a,CPU	MSKO a	Set up the Interrupt Disable flags in all devices simultaneously, according to the mask code in accumulator a. Each device is associated with one of the bits in the accumulator, and its flag is set (mask bit = 1) disabling interrupt from that device, or cleared (mask bit = 0) enabling interrupts from it. A MSKO with (a) = 177777 disables interrupts from all devices.
DICC 0,CPU	IORST	Generate I/O Reset to clear the BUSY, DONE, and Interrupt Disable flags in all devices. This instruction also clears the processor's ION flag and sets it into 32K addressing mode. (Does not change contents of selected accumulator.)
DOC 0,CPU	HALT	Halt the processor. Requires manual action to restart processor.

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

> READS 3

when executed, deposits the value entered via the Operator Control Unit or mini-switches into A3. If the programmer wishes to also set ION, the assembler instruction mnemonic would have to be used:

> DIAS 3,CPU

This instruction sets ION after reading the DATA display on the Operator Control Unit, or the mini-switches.

The instruction IORST, however, assumes the C function. All I/O device flags are reset and the ION flag is cleared. In order to reset the I/O devices without clearing the ION flag, the regular assembler instruction must be used:

> DIC 0,CPU

As with regular I/O 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 follows:

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 non-zero.
11	SKPDZ CPU	Skip next instruction if the Power Failure flag is zero.

2.6.2.1 Assembler Language Conventions and Examples

CPU instructions are usually written using the special mnemonics shown in Section 2.6.2, however they may also be written in the same manner as regular I/O 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.

2.7 INSTRUCTION EXECUTION TIMES

One of the outstanding features of the POINT 4 computer is the substantial reduction in instruction time over execution time in comparable mini-computers. Table 2-4 gives instruction execution times for the POINT 4 computer.

TABLE 2-4. INSTRUCTION EXECUTION TIMES

Instruction Category	Instruction (Generic Types)	Execution Times* (nanoseconds)
MEMORY REFERENCE	Load or Store Accumulator (LDA,STA)	800
	Increment or Decrement if Zero (ISZ,DSZ)	1100
	Jump (JMP)	400
	Jump to Subroutine (JSR)	400
	Additional Times for Various Addressing Modes:	
	- Each Level of Indirect Addressing	400
	- Auto Indexing	200
ARITHMETIC/ LOGIC	Arithmetic/Logic Instructions (COM,NEG,MOV,INC,ADC,SUB,ADD,AND)	400
	For skip (SKP) add	0
INPUT/ OUTPUT	Input	900
	For START, CLEAR, or PULSE add	600
	Output	1200
	For START, CLEAR, or PULSE add	400
	No I/O Transfer (NIO)	1200
	For START, CLEAR, or PULSE add	400
	I/O Skips (SKPBN,SKPBZ,SKPDN,SKPDZ)	900
	Interrupt Acknowledge (INTA)	1000
DATA CHANNEL TRANSFERS	Standard Data Channel Transfers:	
	Input	1100
	Output	1700
	High-Speed Data Channel Transfers:	
	Input	900
	Output	1300

*See Next Page

*These times are exclusive of three types of overhead:

1. A 500-nanosecond refresh cycle takes place once every
   16 microseconds - this adds about 3% overhead.

2. Arithmetic/Logic instructions on RAM page boundaries
   (2 least significant digits of address = 76 or 77)
   take an extra 100 nanoseconds - this results in
   approximately 1/4% overhead.

3. If the Optional Operator Control Unit is connected to
   the CPU, a 1.8-microseconds address display cycle takes
   place once every 600 microseconds, resulting in 0.3%
   overhead.

SECTION III

INPUT/OUTPUT INTERFACES

3.1 INTRODUCTION

This section provides information to the user about the basic
operating principles and programming methods for the input/output
devices that are compatible with the POINT 4 computer. There are
two types of I/O devices:

- Those that transfer data via I/O programmed instructions only
- Those that use the data channel for input/output transfers

The following subsections outline the interrupt handling and
priority scheme, conventions for handling the master Teletype or
CRT, programmed transfer handling and data channel transfer
handling. Also provided are I/O bus signal descriptions and I/O
transfer timing diagrams.

3.2 PROGRAM INTERRUPT AND PRIORITY SCHEME

Many input/output devices require service within a short time
after they request it, but they need service infrequently relative
to the processor speed and only a small amount of time is required
to service them. Failure to service within the specified time
(which varies among devices) causes operation of the device below
its maximum speed and can result in loss of information.

The use of interrupts in the current program sequence facilitates
concurrent operation of the main program and a number of
peripheral devices. The program interrupt scheme allows an I/O
device to gain control of the processor. When an interrupt
occurs, the processor suspends normal program execution and starts
a device service routine. When the routine is completed, the
processor returns to the interrupted program.

3.2.1 Interrupt Sequence

When a device needs service, it sets its Interrupt Request flag.
The processor begins servicing interrupts if all four of the
following conditions exist:

- The processor has just completed an instruction fetch or a
  data channel transfer

- At least one device has a pending Interrupt Request

- Interrupts are enabled (i.e. ION is set)

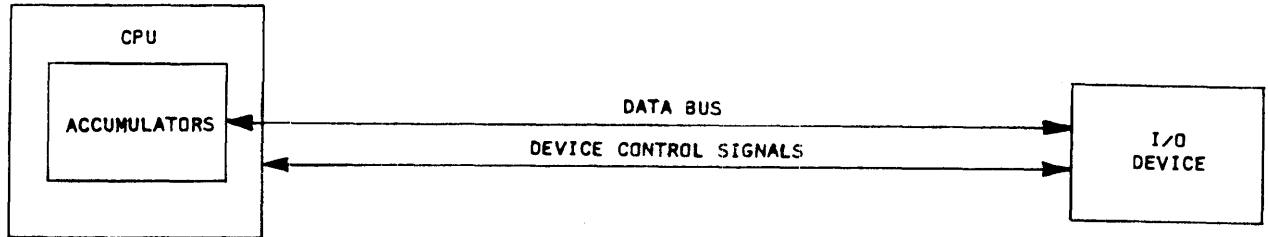- No device is waiting for a data channel transfer

The processor responds to the interrupt request by storing the
value of the program counter into memory location 0 and jumping to
the instruction addressed by memory location 1. Location 1 must
contain the address of the interrupt handling routine. Interrupts
are disabled at the start of the interrupt service cycle and must
be re-enabled by the software at the end of the interrupt
service. |

3.2.2 Device Priority

The processor features a special Interrupt Acknowledge instruction
that eliminates the need for lengthy device polling. This
instruction inputs the address of the interrupting device into an
accumulator register permitting the interrupt service routine to
identify the device requesting service. The computer uses a
three-way priority system to determine which, if any, device may
interrupt the processor at a given moment. First: the processor
contains a programmable Interrupt ON (ION) flag. The processor
recognizes interrupt requests only when this flag is set. Second:
the processor can selectively disable the interrupt capability of
each device with the device Interrupt Mask flags (See Section |
2.6.2, MSKO instruction). Third: If two or more devices request |
interrupts simultaneously the priority resolution is made by the
"Jumper-Saver" logic on the backplane. A device whose controller
board is physically nearer the processor is given priority over a
device that is further away. (See Appendix E for an example of |
interrupt programming.) |

3.3 PROGRAMMED TRANSFERS

For programmed input/output the program directly controls the data
transfer between the CPU and the I/O device. As discussed in
Section 2.6, on Input/Output Instructions, each data word is
transferred between an accumulator specified in the instruction
and an I/O device buffer (A, B, or C) specified in the
instruction.

```
 ┌─────────────────────────┐
 │    CPU                   │
 │  ┌──────────────┐        │                                      ┌──────────┐
 │  │              │        │          DATA BUS                    │          │
 │  │ ACCUMULATORS │◄───────┼──────────────────────────────────►  │   I/O    │
 │  │              │        │      DEVICE CONTROL SIGNALS          │  DEVICE  │
 │  │              │◄───────┼──────────────────────────────────►  │          │
 │  └──────────────┘        │                                      └──────────┘
 │                          │
 └─────────────────────────┘
```

3.3.1 Master CRT/Teletype Terminal Interface

The standard Teletype or CRT I/O controller has separate interface
logic for input and output. Input and output device codes are
also separate. The input logic interfaces to the keyboard and, in
some Teletypes, a paper tape reader. The output logic interfaces
to the printer or video display and, in some Teletypes, the paper
tape punch. When the CRT or Teletype is connected to the
computer, a character entered on the keyboard for input to the
computer must be "echoed" back to the output interface logic on
the terminal in order to appear on the screen or paper.

All alphanumeric and control characters are represented by
standard ASCII code (see Appendix C) consisting of eight bits, the
most significant of which is usually an even parity bit.

The following are programming conventions for handling CRT and
Teletype terminals:

Instruction Formats: Terminal output and input use separate codes
        in bits 10-15 of the instruction. The S or C pulse may be
        sent to clear BUSY and/or DONE flags and control the
        starting of the transfer between the interface and the
        device. The data transfer output instructions transmit bits
        8-15 from the specified accumulator to the output interface
        register. The input instruction loads the input interface
        register into bits 8-15 and resets bits 0-7 of the specified
        accumulator.

Terminal Output: Transmission to the terminal takes place when an
        S pulse is sent, which sets the output BUSY flag. When the
        character has been printed, the interface clears the BUSY
        flag, sets the DONE flag and requests an interrupt, if
        interrupts are enabled. Terminal output uses the device code
        11, the mnemonic TTO, and uses bit 15 to control the
        interrupt mask flag. To transfer a character from an
        accumulator to the terminal output buffer, the following
        instruction is used:

```
                    DOA  x,TTO
```

(where x may be any of the four accumulators).

If the NIOS instruction is used to send the S pulse, the
instructions must appear in the following sequence:

```
                    DOA  x,TTO
                    NIOS TTO
```

Normally this operation is done in one instruction:

```
                    DOAS x,TTO
```

Terminal Input or via CRT Keyboard: Terminal input uses the octal
    device code 10, the mnemonic TTI, and uses bit 14 to control
    the interrupt mask flag. When a key is pressed on the
    keyboard, the character is placed in the input buffer, and
    DONE is set. If interrupts are enabled for that device, an
    interrupt is requested. The program then reads the
    character with a

```
                    DIA  x,TTI
```

instruction and uses an S pulse to clear DONE. The S pulse
may either be part of the DIA instruction,

```
                    DIAS x,TTI
```

or be entered with a NIO instruction,

```
                    DIA  x,TTI
                    NIOS x,TTI
```

Terminal Input via Paper Tape Reader: When paper tape is used for
    input, the paper tape reader must be started by the program
    using a

```
                    NIOS TTI
```

instruction. This instruction sets BUSY and clears DONE.
When the character has been read from paper tape into the
controller, the device controller clears BUSY and sets DONE,
producing an interrupt if interrupts are enabled. The
program then reads the character with a

```
                    DIA  x,TTI
```

instruction. Sequential characters can be read by using a

```
                    DIAS x,TTI
```

instruction. This results in the reading of one character
and the restarting of the paper tape reader for reading of
the next character.

3.4 DATA CHANNEL TRANSFERS

Mass storage devices such as tape drives, discs and mass storage units can transfer blocks of data at high speeds directly into memory, without requiring programmed I/O instructions for each word transferred, by using the DMA (direct memory access) data channel. Data channel device interface logic contains both conventional device registers and flags, and special data channel logic.

The program initiates a data channel transfer by supplying certain parameters to the device registers and starting the device. The device automatically transfers one or more data words to or from memory. When finished with the DMA transfer, the device generates an interrupt if so enabled. At the start of each instruction cycle, the processor checks to see if a device is requesting data channel service. If a device is requesting data channel service, the data channel transfer is performed before going on with the instruction. Several data channel devices can be active at the same time, with devices closest to the processor having channel priority over devices further away.

The POINT 4 has two jumper selectable data channel speed options. They are:

        Standard Data Channel
            (Jumper CPU board Pin A93 to ground)
            Input   - 1100 nanoseconds
            Output - 1700 nanoseconds

        High-Speed Data Channel
            (Do not jumper CPU board Pin A93 to ground)
            Input   - 900 nanoseconds
            Output - 1300 nanoseconds

Choice of standard or high speed data channel depends on the speed of the devices connected to the channel. See Section 6.4 for data channel selection criteria.

The time a device must wait for data channel access depends on when its request is made within an instruction and how many devices of higher priority are also requesting access. Once the processor reaches a point at which it can pause to handle transfers, a given device must wait until all devices closer than it on the bus have been serviced. Under normal conditions, a device can preempt all processor time if it requests access at the maximum rate. An exception is made if Power-Fail has been sensed, in which case the data channel is allowed only every other cycle - the alternate cycle being used for Power-Fail Interrupt processing. At less than the maximum rate the closest device never waits longer than the time required for the processor to finish the instruction that is being performed when the request is synchronized. However, indirect addressing can extend this beyond the normal instruction execution time.

3.5 INPUT/OUTPUT BUS INTERFACE SIGNALS

Input/Output Bus signals connect the processor logic to peripheral device logic. The logic for programmed I/O transfers and data channel transfers forms the interface between the processor Main Data Bus and the peripheral device controller logic. Logic to implement both I/O transfer and I/O skip 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 subsection describes the POINT 4 I/O bus and control signals.

3.5.1 Input/Output Interface Signals

Signals on the Input/Output Bus can be grouped into the following signal classifications:

a) 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.

b) Device Codes (6 lines): Codes used to designate the peripheral device involved in an input/output instruction.

c) Programmed Transfer Signals (6 lines): These signals, generated by the CPU in response to input/output instructions for data transfers, are used to control data transfers for programmed input/output devices.

d) Device Control Signals (4 lines): These signals are generated by the CPU in response to input/output instructions, and are used to initialize and control I/O devices. The signals affect only the device whose device code is in the instruction, except in the case of the IORST instruction which resets all devices.

e) Skip Testing Flags (2 lines): Flags supplied to the CPU, when skip-testing is required.

f) Interrupt Control Signals (5 lines): Signals used to initialize and control the interrupt sequence.

g) Data Channel Transfer Signals (6 lines): Signals used to control data channel transfers between memory and a peripheral device.

Figure 3-1 is a diagram of I/O signals across the I/O Bus. Table 3-1 divides these signals by signal classifications, designates the signal name and defines each signal function.

CPU

DATA 0 — DATA 15 (DATA BUS)   16 } DATA BUS

DS0 — DS5 (DEVICE CODE)   6 } DEVICE CODE

PROGRAMMED DATA TRANSFER SIGNALS
(DATIA, DATIB, DATIC, DATOA, DATOB,
DATOC)   6 } PROGRAMMED TRANSFER SIGNALS

DEVICE CONTROL SIGNALS
(IORST, STRT, CLR, IOPLS)   4 } DEVICE CONTROL SIGNALS

BUSY AND DONE FLAGS
(SELB, SELD)   2 } FLAGS FOR I/O SKIPS

I/O
DEVICE

REQUEST ENABLE (RQENB)   1

INTERRUPT REQUEST (INTR)   1

INTERRUPT PRIORITY (INTPIN)   1 } INTERRUPT CONTROL SIGNALS

INTERRUPT ACKNOWLEDGE (INTA)   1

MASK OUT (MSKO)   1

DATA CHANNEL REQUEST SIGNALS TO CPU
(DCHR, DCHMO)   2

DATA CHANNEL SERVICE SIGNALS TO DEVICE
(DCHPIN, DCHA, DCHI, DCHO)   4 } DATA CHANNEL CONTROL SIGNALS

FIGURE 3-1. INPUT/OUTPUT SIGNALS

TABLE 3-1. INPUT/OUTPUT SIGNALS BY CLASSIFICATION

Signal Group	Signal Name*	Direction	Description
Data Bus	DATA0- to DATA15-	Bidirectional	All data and addresses are supplied to and from the device via these lines. DATA0- is the MSB.
Device Code	DS0- to DS5-	From CPU	The CPU places the device code (bits 10-15 of the instruction word) on these lines during the execution of an input/output instruction. DS0- is the MSB.
Programmed Data Transfer Signals	DATIA+	From CPU	Data In A. Generated by a DIA instruction. Causes the A buffer of the device whose device code is on the lines to be placed on the Data Bus for entry into the accumulator specified by the instruction.
	DATOA+	From CPU	Data Out A. Generated by a DOA instruction. Causes the accumulator specified in the DOA instruction to be placed on the Data Bus for entry into the A buffer of the device whose device code is on the lines.
	DATIB+	From CPU	Data In B. Generated by a DIB instruction. Functions like Data In A, except uses buffer B.
	DATOB+	From CPU	Data Out B. Generated by a DOB instruction. Functions like Data Out A, except uses buffer B.
	DATIC+	From CPU	Data In C. Generated by a DIC instruction. Functions like Data In A, except uses buffer C.
	DATOC+	From CPU	Data Out C. Generated by a DOC instruction. Functions like Data Out A, except uses buffer C.

*All signal names ending with "+" are active high, those ending with "-" are active low.

TABLE 3-1. INPUT/OUTPUT SIGNALS BY CLASSIFICATION (Continued)

Signal Group	Signal Name*	Direction	Description
Device Control Signals	IORST+	From CPU	Input/Output Reset. Generated when APL is pressed on the Mini-panel, when RESET is pressed on the Operator Control Unit, when an IORST instruction is being executed, and during power turn-on.
	STRT+	From CPU	Start. Generated when the CTRL field of an input/output transfer instruction contains code 01. It usually clears the DONE flag and Interrupt Request, and sets the BUSY flag in the device whose device code is on the lines.
	CLR+	From CPU	Clear. Generated when the CTRL field of an input/output transfer instruction contains code 10. It usually clears the BUSY and DONE flags and the Interrupt request in the device whose device code is on the lines.
	IOPLS+	From CPU	I/O Pulse. Generated when the CTRL field of an input/output transfer instruction contains code 11. The effect, if any, depends on the device.
Skip Testing Flags	SELB-	From Device	Selected Device Busy. Supplied to CPU by the device whose device code is on the lines when the BUSY flag is set. Indicates that the device is busy.
	SELD-	From Device	Selected Device Done. Supplied to the CPU by the device whose device code is on the lines when the DONE flag is set. Indicates that the device is done.

TABLE 3-1. INPUT/OUTPUT SIGNALS BY CLASSIFICATION (Continued)

Signal Group	Signal Name*	Direction	Description
Interrupt Control Signals	RQENB-	From CPU	Request Enable. Generated during each memory read/write cycle to synchronize INTR- and DCHR-. In any device, changes in INTR- or DCHR- may only occur following the leading edge (high-to-low transition) of RQENB-.
	INTR-	From Device	Interrupt Request. This signal goes low (following the leading edge of RQENB-) if the device wants to request an interrupt.
	INTPIN-	From CPU	Interrupt Priority Input. Produced by "Jumper-Saver" logic on the backplane for the highest priority device requesting an interrupt.
	INTA+	From CPU	Interrupt Acknowledge. Generated by an INTA instruction. Causes the device whose INTPIN- line is low to place its device code in bits 10-15 of the Data Bus for entry into accumulator specified in the instruction.
	MSKO-	From CPU	Mask Out. Generated by a MSKO instruction. Commands all I/O devices to set their Interrupt Disable flags according to the state of the associated Mask Bit in the word on the Data Bus.

TABLE 3-1.   INPUT/OUTPUT SIGNALS BY CLASSIFICATION (Continued)

Signal Group	Signal Name*	Direction	Description
Data Channel Transfer Signals	DCHR-	From Device	Data Channel Request. This signal goes low (following the leading edge of RQENB-) if the device wants to request a data channel transfer.
	DCHPIN-	From CPU	Data Channel Priority. Produced by "Jumper-Saver" logic on the backplane for the highest priority device requesting an interrupt.
	DCHA-	From CPU	Data Channel Acknowledge. Generated by CPU in response to a Data Channel Request. Initiates a data channel cycle in the device whose DCHPIN- is low. The device places the memory address for data channel access on the Data Bus.
	DCHMO-	From Device	Data Channel Mode. Generated by a device connected to the data channel while DCHA- is low. Indicates the type of data channel cycle being requested as follows: DCHMO- Type of Cycle 0 (high) Data Cut (from CPU) 1 (low) Data In (to CPU)
	DCHI+	From CPU	Data Channel In. When the mode is Data In, DCHI+ is generated during the time the device is placing a data word on the Data Bus.
	DCHO+	From CPU	Data Channel Out. When the mode is Data Out, DCHO+ is generated during the time that the word accessed from memory is on the Data Bus.

3.5.2 Backplane Pin Signal Connectors

All signal connections between the processor and each controller take place via two 100-pin backplane connectors. Figure 3-2 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 peripheral controllers.

Bottom	A		Top	Bottom	B		Top
GND	2	1	GND	GND	2	1	GND
+5V	4	3	+5V	+5	4	3	+5V
−5V	6	5	+5BU		6	5	
	8	7	PWRGON		8	7	
+15V	10	9	−5BU		10	9	
	12	11	PWRF		12	11	
	14	13			14	13	
	16	15			16	15	
	18	17			18	17	DCHMO−
	20	19			20	19	
	22	21			22	21	
	24	23			24	23	
	26	25			26	25	
	28	27			28	27	
	30	29			30	29	INTR−
	32	31			32	31	
GND	34	33	GND		34	33	DCHO+
	36	35			36	35	DCHR−
MSKO−	38	37			38	37	DCHI+
INTA+	40	39			40	39	
DATIB+	42	41			42	41	RQENB−
DATIA+	44	43			44	43	
DS3−	46	45		+15V	46	45	
DATOC+	48	47			48	47	
CLR+	50	49		GND	50	49	
STRT+	52	51			52	51	
DATIC+	54	53			54	53	
DATOB+	56	55		DATA14−	56	55	DATA7−
DATOA+	58	57		DATA11−	58	57	DATA5−
DCHA−	60	59		DATA8−	60	59	DATA12−
DS4−	62	61		DATA0−	62	61	DATA4−
DS5−	64	63		DATA13−	64	63	DATA9−
DS2−	66	65		DATA15−	66	65	DATA1−
DS1−	68	67			68	67	
IORST+	70	69			70	69	
DSO−	72	71			72	71	
IOPLS+	74	73			74	73	DATA3−
	76	75			76	75	DATA10−
	78	77			78	77	
SELD−	80	79			80	79	
SELB−	82	81		DATA2−	82	81	−5V (not CPU)
PEL−	84	83		+15V	84	83	
RUNL−	86	85			86	85	
CL−	88	87			88	87	
CONT−	90	89			90	89	
STOP−	92	91		GND	92	91	−15V (not CPU)
DCHPIN−	94	93	DCHPOUT−	+12BU	94	93	−15V (not CPU)
INTPIN−	96	95	INTPOUT−		96	95	DATA6−
+5V	98	97	+5V	+5V	98	97	+5V
GND	100	99	GND	GND	100	99	GND

Side labels (A, Bottom): (not CPU); CPU MINI-PANEL; (not CPU)
Side labels (A, Top): (CPU only); (HISPDC− on CPU)
Side labels (B, Bottom): (not CPU); (not CPU); (CPU only)

FIGURE 3-2. BACKPLANE I/O SIGNALS

3.6   INPUT/OUTPUT TRANSFER TIMING

Three classes of operations take place on the I/O Bus: operations associated with programmed I/O instructions, operations associated with interrupt handling, and operations associated with data channel transfers.   Timing diagrams in this section represent each signal or group of signals by a horizontal line with a raised section representing the active state.   Control signals generated at a specific time to control a particular function show the raised line for the time that the signal is active.   For signals carrying binary information, the raised line indicates the amount of time during which that information remains on the bus.   Raised lines may represent either high or low voltage levels, depending on whether the signal is active when low or high.   (See Table 3-1 for active voltages on all signals.)   All times are in nanoseconds on timing diagrams.

3.6.1   Programmed I/O Instruction Timing

Figure 3-3 shows the timing for Data In (DIx) and Data Out (DOx) instructions both without and with a Control Pulse (S, C or P) specified.   In all cases, the processor first places the appropriate device code on the device select lines DS0-DS5.   The selected device then responds to the signals which follow.

Data In: During Data In transfers, the processor generates a
       DATIA, DATIB, or DATIC signal.   The device selected then
       places the contents of its appropriate buffer onto the data
       transfer lines.   At the end of the DATIx active signal the
       processor strobes the value on the data lines into the
       appropriate processor accumulator.   Following the transfer
       the processor generates the pulse for a START (S), CLEAR
       (C), or PULSE (P), if called for by the instruction.

Data Out: During Data Out transfers, the processor loads the
       contents of the accumulator selected by the instruction onto
       the data transfer lines.   The processor then generates a
       DATOA, DATOB, or DATOC signal, which causes the device to
       strobe in the data to the buffer specified.   When the data
       has been loaded into the buffer, the processor generates
       the pulse for START (S), CLEAR (C), or PULSE (P), if called
       for by the instruction.
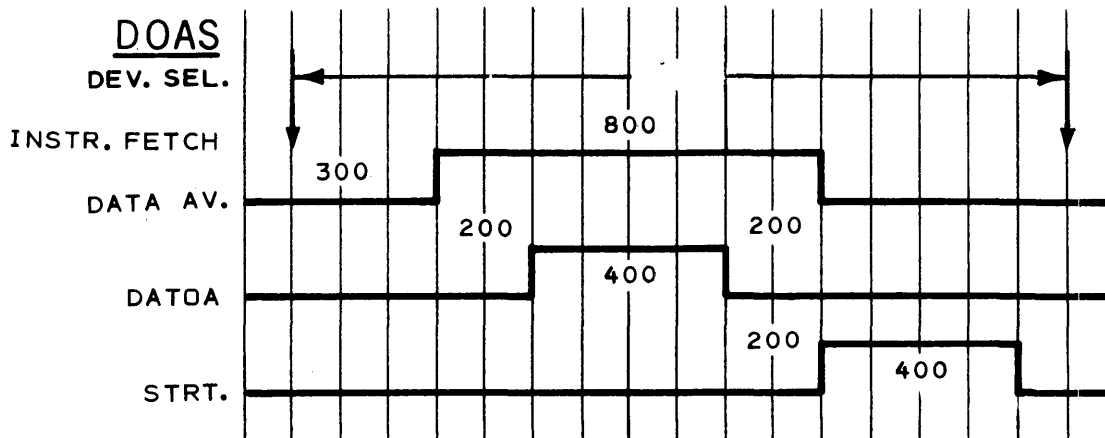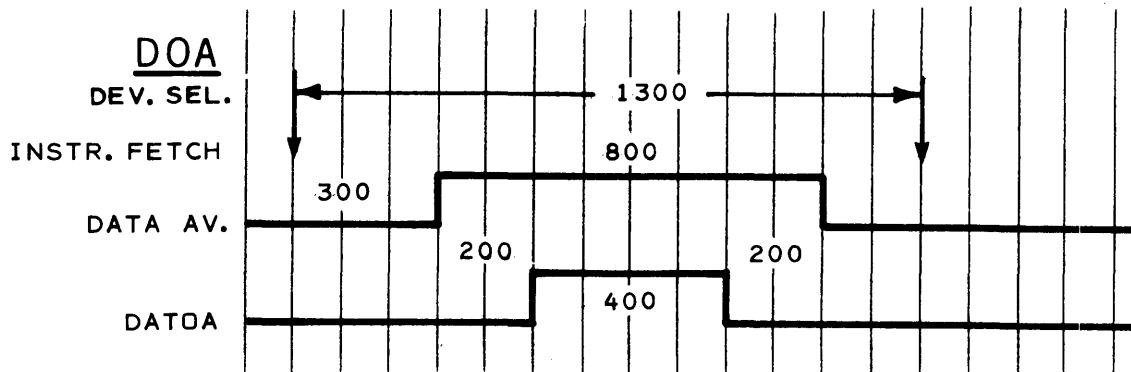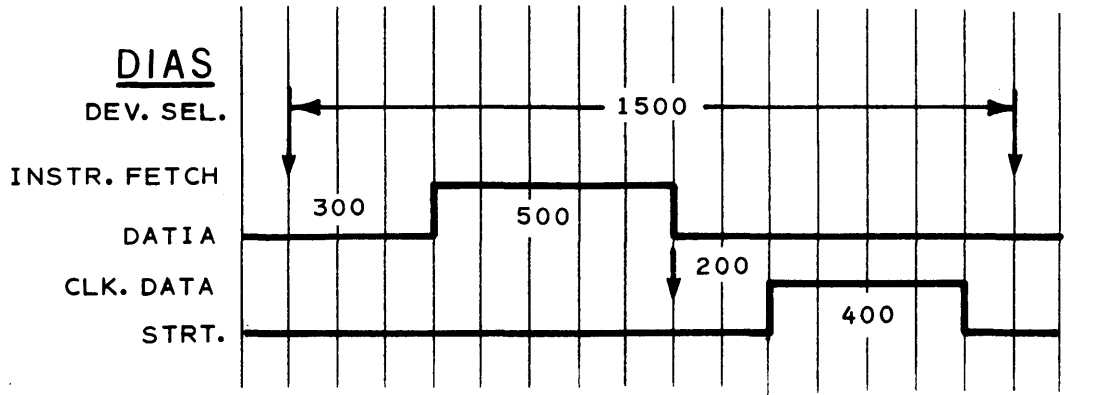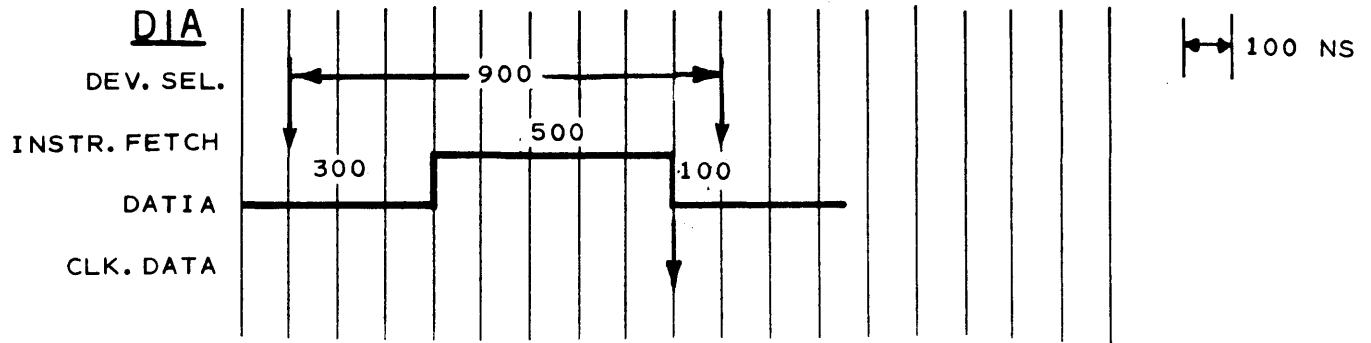
FIGURE 3-3. PROGRAMMED I/O INSTRUCTION TIMING

3.6.2 Program Interrupt Timing

At the end of every memory cycle the processor generates the
signal RQENB and places it on the I/O Bus. All devices receive
the RQENB signal and each responds according to its need for
service. Any device requiring interrupt servicing pulls the
signal INTR- low.

Figure 3-4 is a timing diagram of interrupt handling.

RQENB

|←100→|←300→|

INTR

SENSED

IR LOAD

IR EXEC

JMP/JSR
OR ALU

SAVE PC
IN LOC 0

FETCH ADR
FR LOC 1

FETCH INSTR.
a LOC 1

LAST
INSTRUCTION
PRIOR
TO
INTERRUPT

INTERRUPT RESPONSE

FIRST
INSTRUCTION
OF
INTERRUPT
SERVICE

|←———— 1200 NS. ————→|
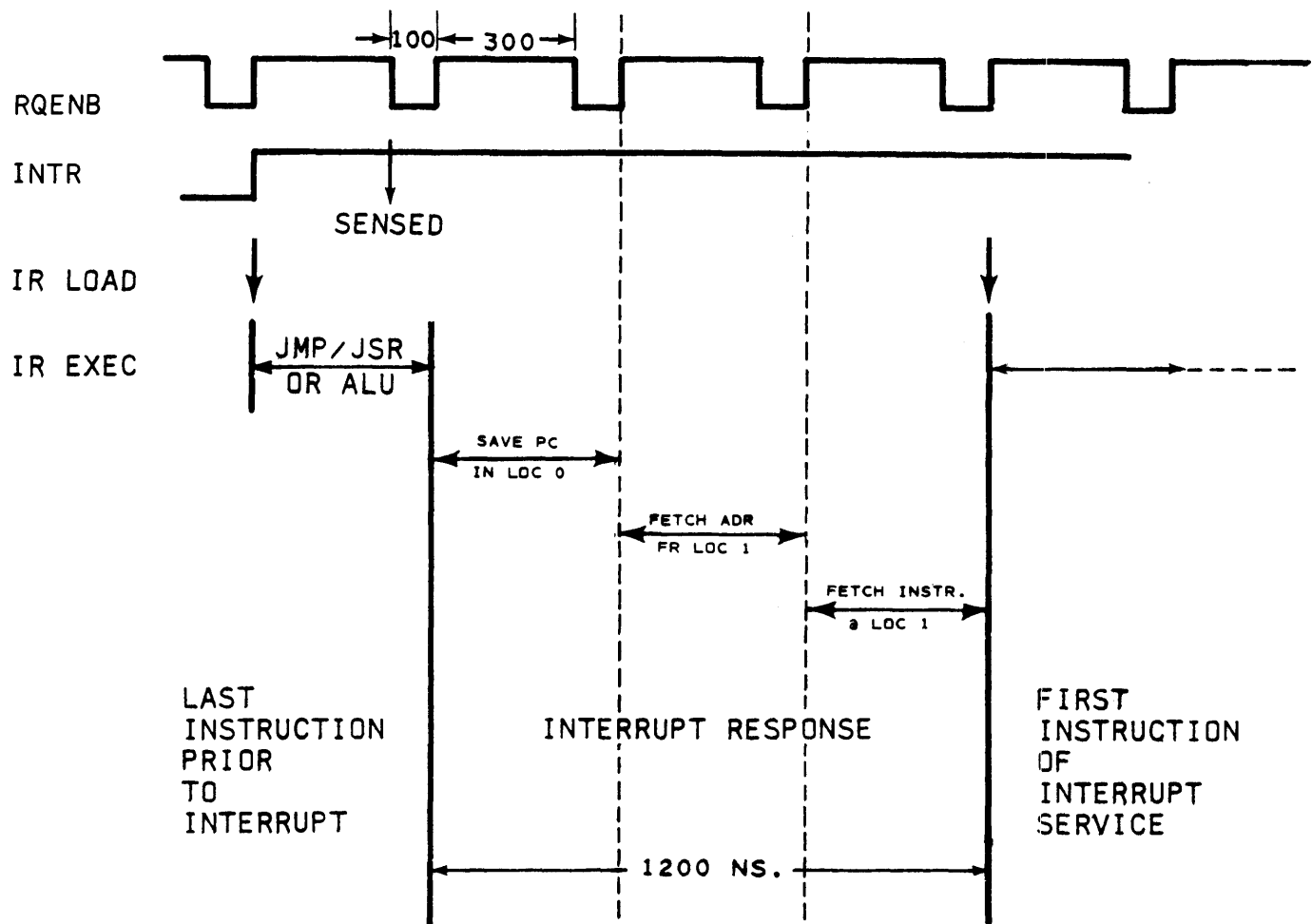
RQENB = Request Enable: Provides a clock signal to synchronize
Interrupt requests (as well as Data channel requests) from
peripheral controllers. Its nominal period is 400ns
(100ns off, 300ns on). Controllers may change INTR
(Interrupt Request) or DCHR (Data Channel Request) only at
the leading edge of RQENB.

INTR = Interrupt Request: Sensed by CPU at the trailing edge of
RQENB which occurs 100ns before the end of each
instruction. If INTR is present, the next three memory
cycles (1200ns) are taken to save PC in location 0 and to
jump to the Interrupt Service routine whose address is in
location 1.

FIGURE 3-4.   INTERRUPT TIMING

3.6.3 Data Channel Transfer Timing

Data channel transfers are in either the input or output direction: Data Channel Input being a write into memory and Data Channel Output being a read from memory. In either case the device first requests use of the I/O Bus. When the processor acknowledges the request, it stops program execution long enough to conduct the transfer between the device and memory.

Operations in data channel requests are similar to those of an interrupt request. At the end of every memory cycle the processor generates the signal RQENB and places it on the I/O Bus. All devices receive the RQENB signal and each responds according to its need for service. Any device requiring data channel service pulls the DCHR- line low. The "Jumper Saver" logic on the CPU backplane then determines which is the highest priority device requesting data channel service, and sends DCHP- (Data Channel Priority) to that device. Devices whose DCHP- line is inactive (high) ignore subsequent data channel control signals.

When the processor is ready to process the data channel request, it activates the signal DCHA (Data Channel Address). The device whose DCHP- line is active (low) places the address for the DMA transfer on the I/O bus during DCHA. At the same time the device also activates or negates DCHM0 to specify whether an input or output transfer is to take place.

When DCHA terminates, the processor strobes the address into its memory address register. From this point on the operation depends on the direction of the data channel transfer.

Data Channel Input: When a data input transfer is required, the processor transmits DCHI immediately following the trailing edge of DCHA. The device then places the data word onto lines DATA0 through 15. Near the trailing edge of DCHI, the processor stores the data word into memory, and the device removes the data word from lines DATA0 through 15.

Data Channel Output: In an output transfer, the processor starts a Read memory cycle at the trailing edge of DCHA. When the data has been fetched from memory, the processor places the word on lines DATA0 through 15 and activates a DCHO signal. The device then fetches the data from the data lines.

When the transfer required is a single-word transfer, the device clears DCHR the next time it receives RQENB. If the transfer required is several words in consecutive data channel cycles, the DCHR flag should remain active until the leading edge of RQENB following the DCHA of the last transfer desired.

Figure 3-5 is a timing diagram of Standard Data Channel operations and Figure 3-6 is a timing diagram of High Speed Data Channel operations.
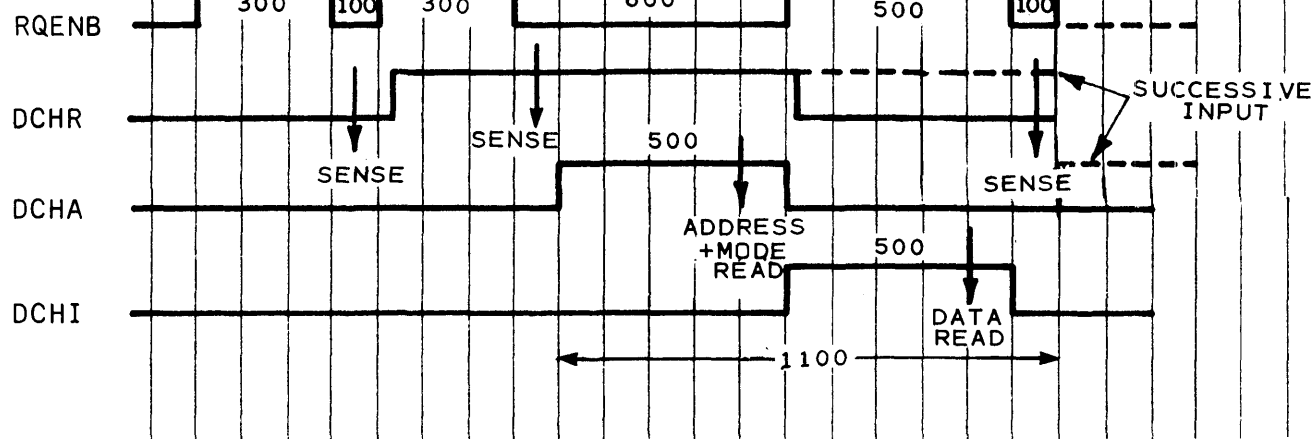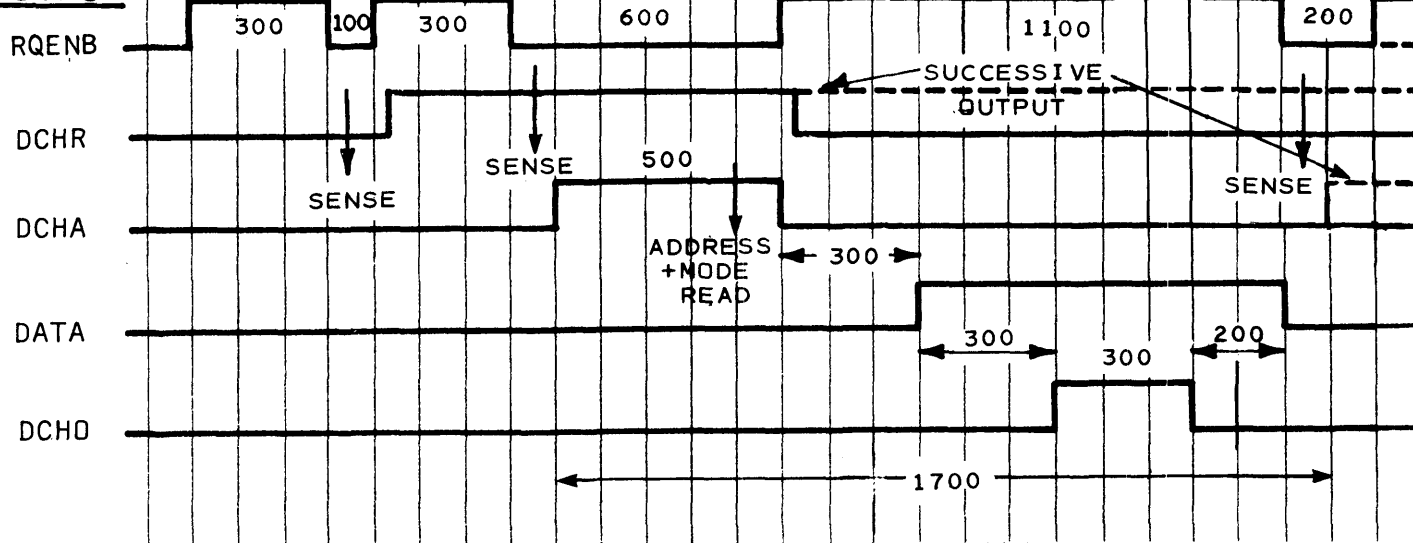
FIGURE 3-5.   STANDARD DATA CHANNEL TIMING

INPUT

RQENB 300 100 300 500 400 100 300

DCHR

SENSE

SENSE

DCHA SUCCESSIVE INPUT

SENSE

400
ADDRESS
+MODE
READ

DCHI

400

DATA READ

900

OUTPUT

RQENB 300 100 300 500 800 100 300

DCHR

SENSE

SUCCESSIVE OUTPUT

400 SENSE

SENSE

DCHA

ADDRESS
+MODE
READ 300 500

DATA

200 200

DCHO

1300
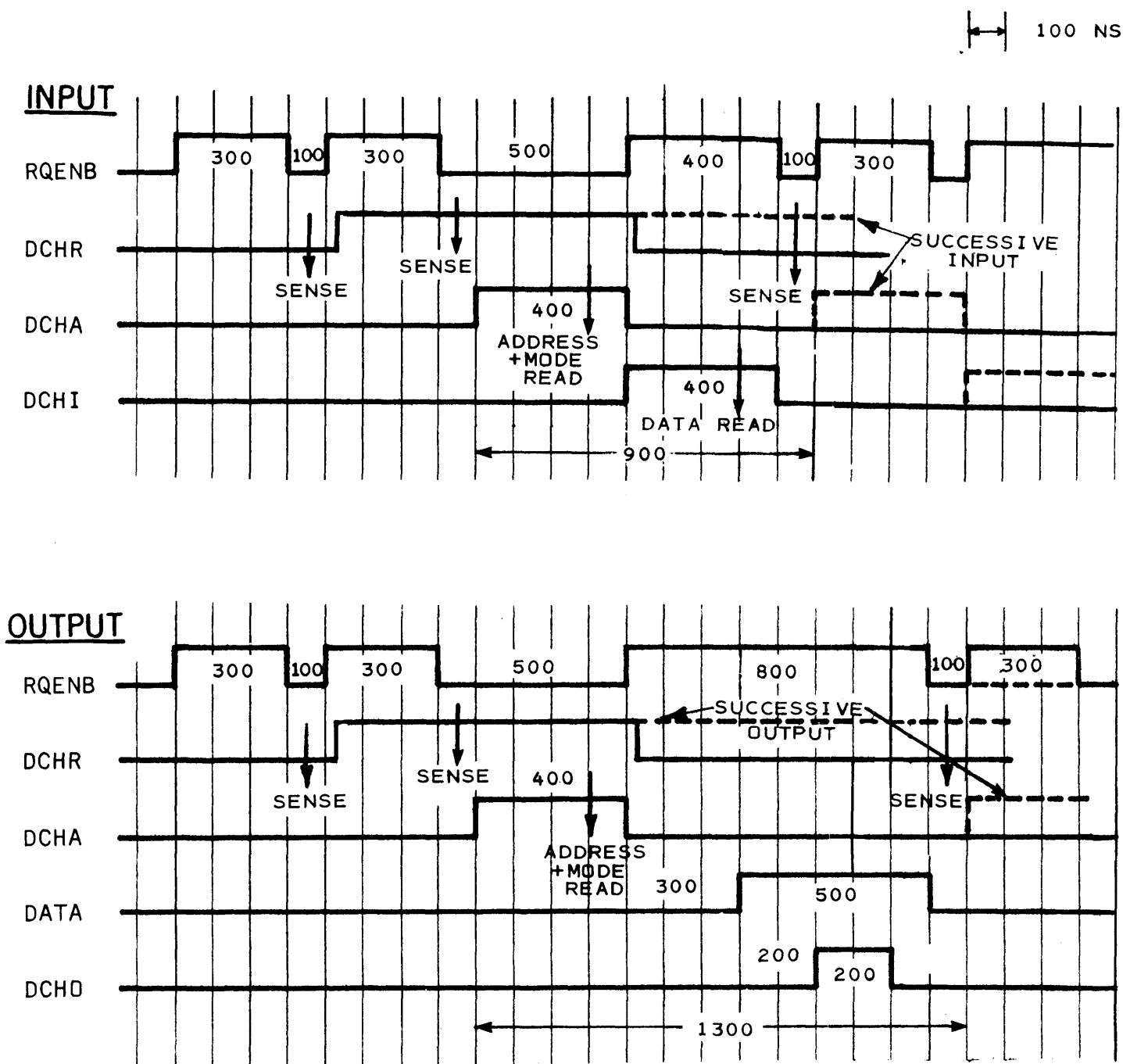
FIGURE 3-6. HIGH SPEED DATA CHANNEL TIMING

3.7 INPUT/OUTPUT CONNECTORS

Four connector sockets are found on the rear of the POINT 4 processor chassis. Figure 3-7 illustrates the positions of these connectors on the processor backplane.

3.7.1 Power Connectors

Two of these sockets receive power cables carrying power supply voltages from the POINT 4 Power Supply to the processor chassis. These connector sockets are:

Socket	Mounting Position
12-pin	Top center of the POINT 4 backplane
20-pin	Upper right side of POINT 4 backplane

Instructions for the connection of these cables between the POINT 4 Power Supply chassis and the POINT 4 Processor chassis are found in Section 4.3 of this manual.

3.7.2 External I/O Device Connector

I/O Bus signals from external peripheral controllers (i.e. controllers not housed inside the POINT 4 chassis) are carried to the processor via a 50-conductor cable. The receptacle for this cable is located on the lower right-hand side of the POINT 4 backplane. Pin assignments for this connector are shown on Figure 3-8.
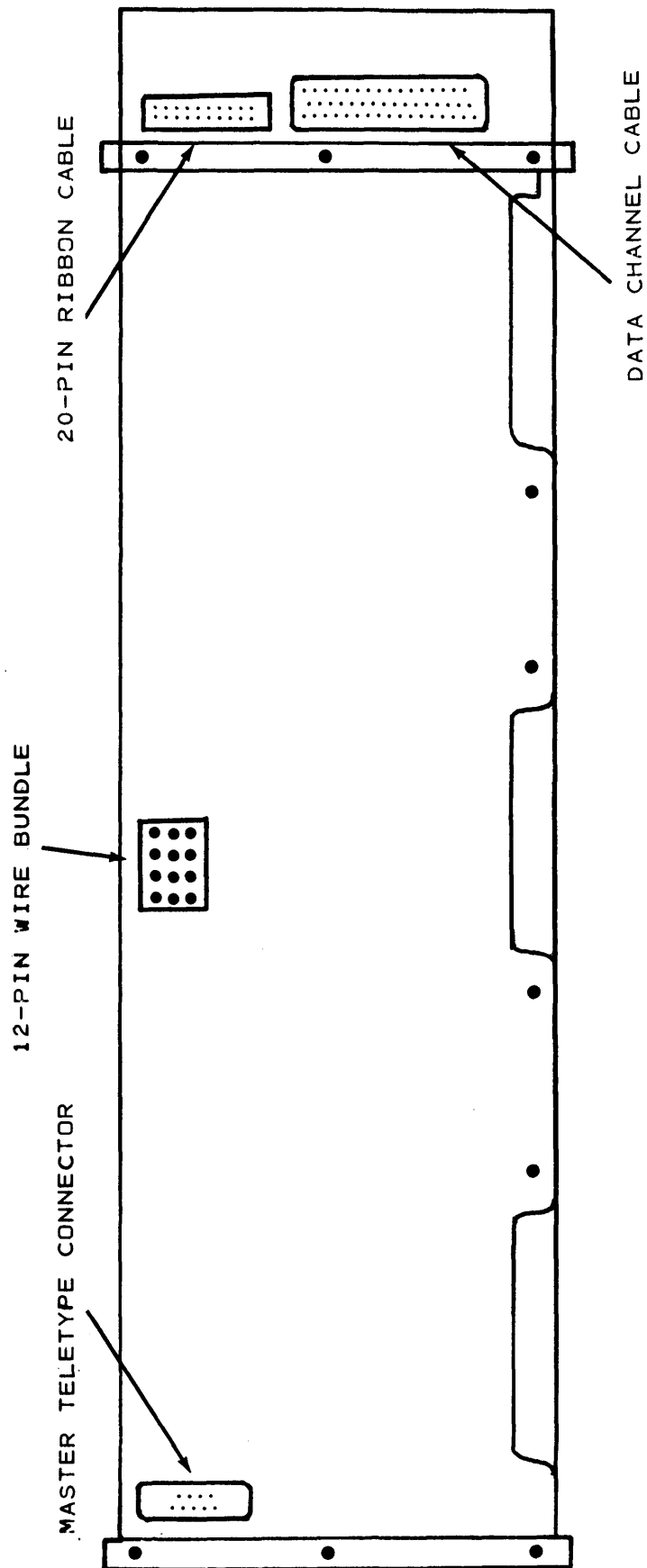
FIGURE 3-7. INPUT/OUTPUT CONNECTOR LOCATIONS

20-PIN RIBBON CABLE

DATA CHANNEL CABLE

12-PIN WIRE BUNDLE

MASTER TELETYPE CONNECTOR

Pin Number	Signal Name	Backplane Reference
1	GND	A1,2
2	CLR+	A50
3	DATA0-	B62
4	DATA1-	B65
5	DATA2-	B82
6	DATA3-	B73
7	DATA4-	B61
8	DATA5-	B57
9	DATA6-	B95
10	DATA7-	B55
11	DATA8-	B60
12	DATA9-	B63
13	DATA10-	B75
14	DATA11-	B58
15	DATA12-	B59
16	DATA13-	B64
17	DATA14-	B56
18	DATA15-	B66
19	DATIA+	A44
20	DATIB+	A42
21	DATIC+	A54
22	DATOA+	A58
23	DATOB+	A56
24	DATOC+	A48
25	DCHA-	A60
26	DCHI+	B37
27	DCHM0-	B17
28	**	B21
29	DCHO+	B33
30	DCHPIO-	JUMPER SAVER
31	DCHR-	JUMPER SAVER
32	DS0-	A72
33	DS1-	A68
34	DS2-	A66
35	DS3-	A46
36	DS4-	A62
37	DS5-	A64
38	INTA+	A40
39	INTPIO-	JUMPER SAVER
40	INTR-	JUMPER SAVER
41	IOPLS+	A74
42	IORST+	A70
43	MSKO-	A38
44	**	B38
45	RQENB-	B41
46	SELB-	A82
47	SELD-	A80
48	STRT+	A52
49	+5V	A3,4
50	GND	A1,2

** Not used by POINT 4 CPU.

Figure 3-8. EXTERNAL I/O DEVICE CONNECTOR PIN ASSIGNMENT

3.7.3 Teletype/Master CRT Connector

The master teletype or CRT can be interfaced to the POINT 4 via a
cable connected at the upper left-hand corner of the backplane.
To use this connector, the controller must be in CPU slot 2
(second from top). Pin assignments for master teletype (CRT)
interface are:

Pin Number	Signal Name	Backplane Reference
1	N/C	–
2	N/C	–
3	–5V	A6
4	TTYIN	2B69
5	N/C	–
6	TTYOUT	2A85
7	+15V	A10
8	STPBIT–	2A87
9	GND	A1,2

SECTION IV

INSTALLATION

4.1 ENVIRONMENTAL REQUIREMENTS

The location in which the POINT 4 computer will be used must be
evaluated prior to installation, to ensure that all power and
cooling requirements are met. The following preinstallation
considerations are necessary:

Power Requirements: The POINT 4 requires a power source of 115
     VAC, 47 to 63 Hz with 5 amperes maximum current draw; or a
     220 VAC, 47 to 63 Hz power source with 2.5 amperes current
     draw. In addition to power requirements for the POINT 4,
     consideration must be made of power resources and electrical
     outlets to handle all peripheral devices to be used with the
     processor.

Temperature Requirements: The POINT 4 requires an environment
     with an adequate temperature control system to maintain a
     recommended 20 to 30 degrees Celsius. Maximum operating
     range is between 0 and 50 degrees Celsius.

Enclosure Requirements: The POINT 4 is packaged in a 7-slot
     chassis, measuring 5.25 inches in height. The power supply
     module for the POINT 4 is mounted in a separate chassis
     measuring 5.25 inches. This provides for flexibility of
     installation and helps isolate heat and noise from the
     processor. Both the processor chassis and power supply
     chassis are designed to be mounted in a standard 19 inch
     equipment rack. The units require sufficient free space
     around the chassis to allow for cooling air flow.

4.2 UNPACKING INSTRUCTIONS

The POINT 4 computer is shipped from the factory in accordance
with configuration requirements for each customer. A complete
test and inspection procedure is followed prior to shipment.
However, it is recommended that each unit be inspected for
completeness and shipping damage prior to installation. Before
shipping cartons are opened, they should be inspected for evidence
of damage due to dropping, puncturing or crushing. If damage is
evident, contact the carrier and the Educational Data Systems
(EDSI) Sales Representative for further instructions.

Unpacking the Carton: The POINT 4 is packaged in a double wall
        corrugated carton. Styrofoam packing inserts surround the
        chassis. The CPU board is packaged in a cardboard carton
        and placed on top of the chassis. See Figure 4-1 for
        illustration of POINT 4 processor chassis packaging. Figure
        4-2 illustrates power supply chassis packaging.

Container Contents: Each item removed from the carton should be
        checked against the packing slip. Inspect all items for
        damage, including cable connectors. Contact the EDSI Sales
        Representative if items are damaged or broken.

BOX WITH CPU BOARD
(SITS ON TOP OF CHASSIS)

STYROFOAM INSERT →

STYROFOAM
INSERTS

DOUBLE WALL
CORRUGATED CARTON
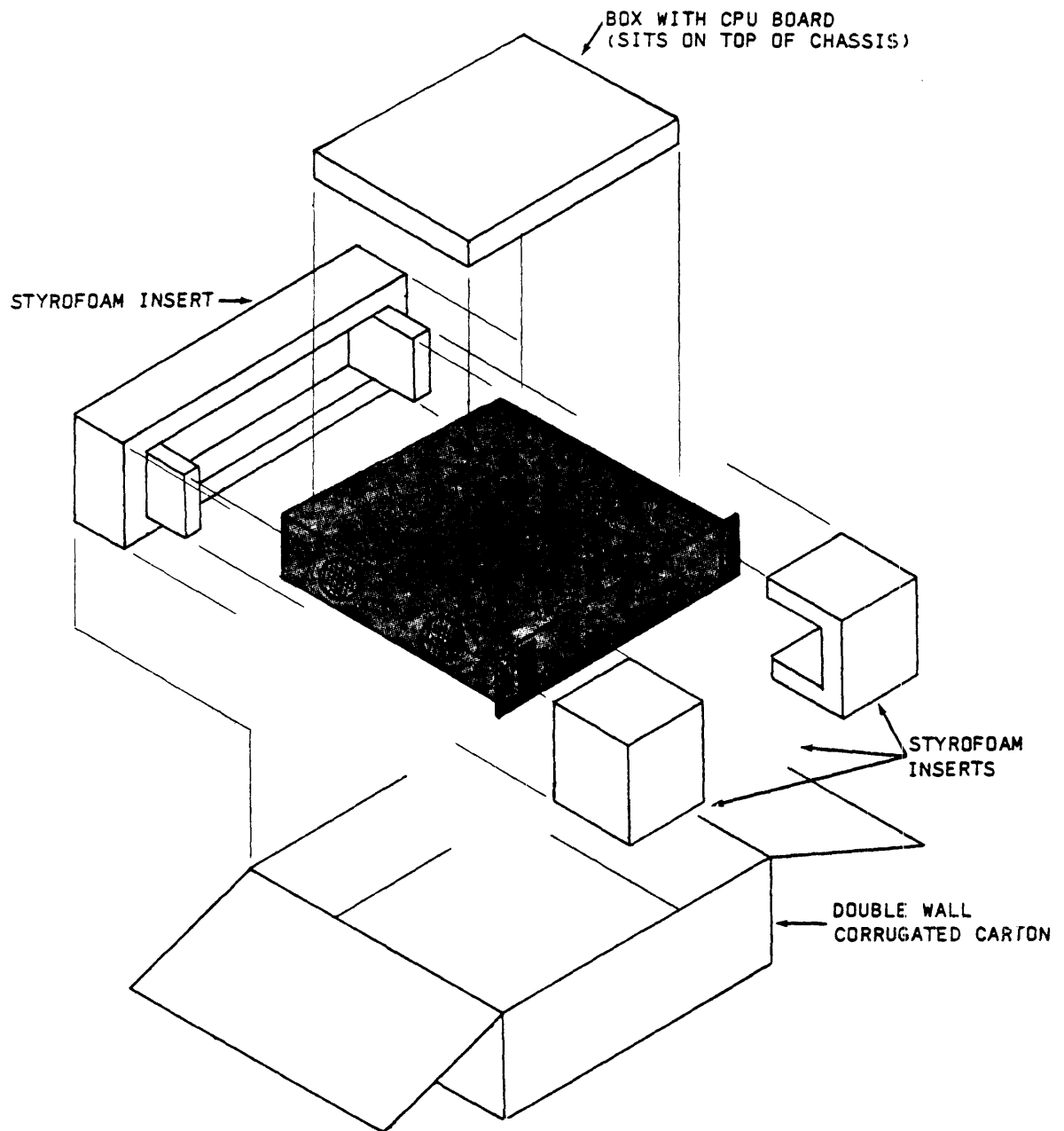
FIGURE 4-1. POINT 4 PROCESSOR CHASSIS PACKAGING
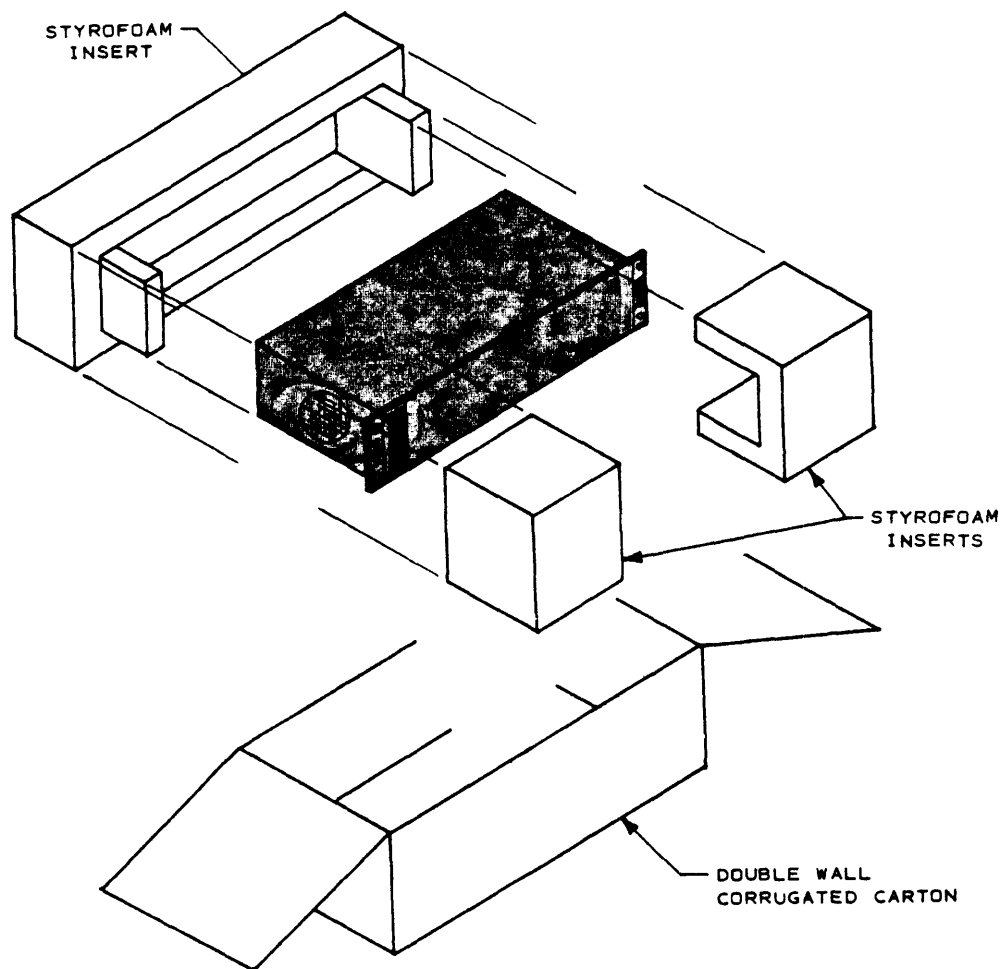
FIGURE 4-2. POINT 4 POWER SUPPLY PACKAGING

4.3 PROCESSOR AND POWER SUPPLY MOUNTING

The standard 19 inch chassis for both the processor and the power supply are mounted according to the following general procedures. Procedures may vary from enclosure to enclosure.

1. The Front Panel: The POINT 4 processor and power supply front panels are removed by snapping off. There are no screws or hinges holding them in place. Removal of the front panel will reveal mounting slots on each side of the chassis.

2. The Chassis: The chassis may be supported both front and rear by shelf angles fastened to the cabinet rails, or by similar brackets. Once the enclosure has been prepared the chassis will slide onto the shelf angles. Bolt the chassis flange directly to the cabinet rails. The 5.25 inch chassis has two mounting slots on each side. See Figure 4-3 for chassis mounting slot locations.

CAUTION

Care must be taken not to crimp, or in any way damage the cables connected to the processor chassis while mounting the chassis.

3. The Printed Circuit Boards: The slots on the 7-slot processor chassis are numbered 1-7 from top to bottom, with the processor/memory board always occupying the top slot. PC boards are mounted component side up. A typical configuration of boards would be:

   - CPU/Memory Board
   - Disk Controller
   - Tape Controller
   - EDSI Mighty Mux (8-port)
   - Mighty Mux 4-port synchronous expansion board
   - Mighty Mux 16-port asynchronous expansion board
   - EDSI Micro-N microprogrammable processor

   Figure 4-4 illustrates a typical board configuration. However, the jumper-free backplane allows for flexibility in board organization, without the necessity to jumper data channel interrupt priority around unused slots.

   EDSI-manufactured PC boards are shipped in separate cartons, except for the CPU board which is mounted in a carton inside the CPU shipping carton (see Figure 4-1). Printed circuit boards should be added to the chassis with caution, making sure the card edge connectors slide smoothly into the backplane sockets.
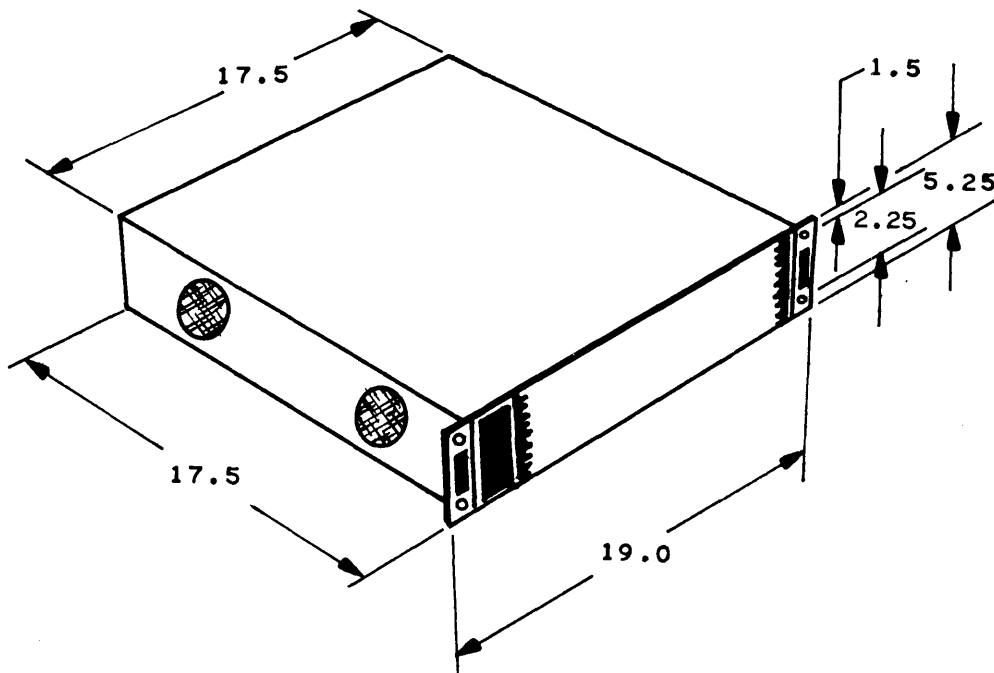
PROCESSOR CHASSIS
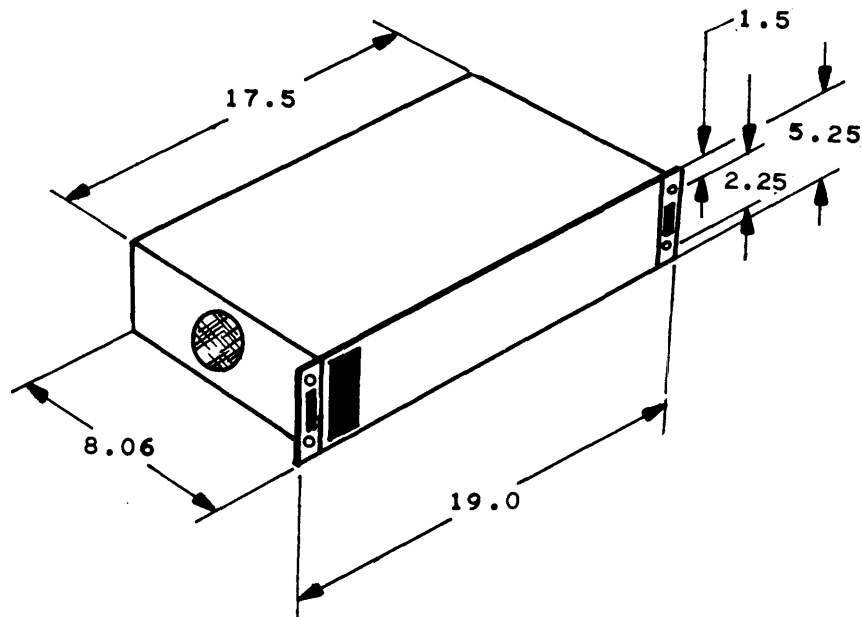
POWER SUPPLY CHASSIS

FIGURE 4-3. POWER SUPPLY CHASSIS AND PROCESSOR CHASSIS MOUNTING SLOTS
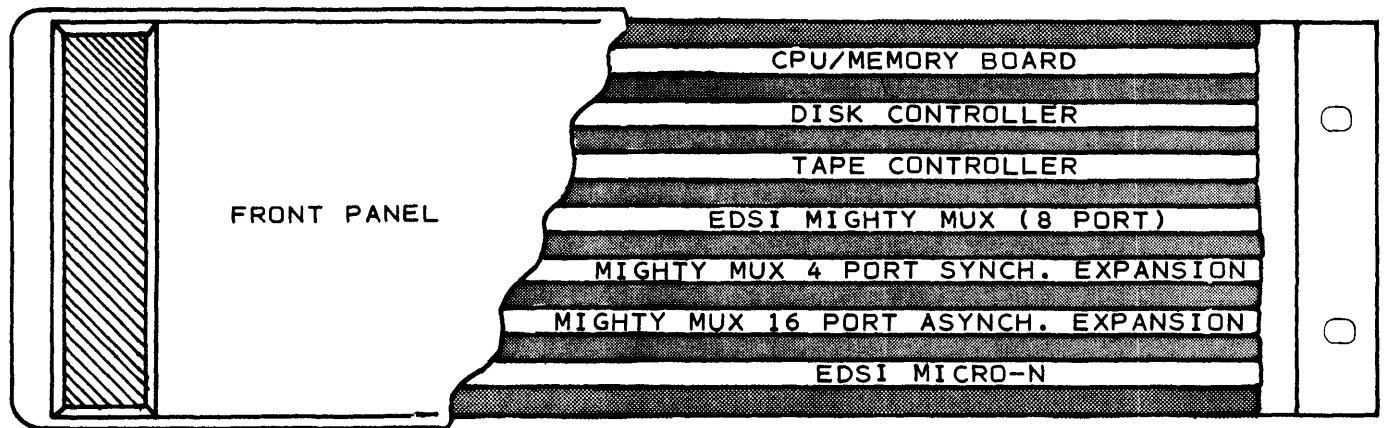
FIGURE 4-4. TYPICAL POINT 4 BOARD CONFIGURATION

The boards shown in the figure (top to bottom) are:
- CPU/MEMORY BOARD
- DISK CONTROLLER
- TAPE CONTROLLER
- EDSI MIGHTY MUX (8 PORT)
- MIGHTY MUX 4 PORT SYNCH. EXPANSION
- MIGHTY MUX 16 PORT ASYNCH. EXPANSION
- EDSI MICRO-N

FRONT PANEL

4. Power Supply Cabling: The POINT 4 processor chassis comes
   with a 6 foot long cable assembly attached to the rear of
   the chassis. This cable assembly consists of a ribbon cable
   and a wire bundle strapped together. Processor chassis
   connections are made at the factory. Check to ensure that
   these cables are properly connected to the chassis.
   Mounting positions on the processor chassis are illustrated
   in Figure 4-5. Connectors and their mounting positions are:

Connector	Mounting Position
12 Pin-Wire Bundle	Top center of the POINT 4 backplane
20 Pin-Ribbon Cable	Upper right side of the POINT 4 backplane
3 Sheathed Power Wires	Connected inside of the fan module

The free ends of the cable assembly must be connected to the
rear of the power supply. Mounting positions on the power
supply chassis are illustrated in Figure 4-6. The
connectors and their mounting positions are as follows:

Connector	Mounting Position
25-Pin Ribbon Cable	Rear upper left-hand corner of power supply chassis
15-Pin Wire Bundle	Rear lower left-hand corner of power supply chassis

12-PIN WIRE BUNDLE
MOUNTING POSITION

20-PIN RIBBON CABLE
MOUNTING POSITION

ENTRY POINT FOR FAN WIRING

COMING FROM POWER CABLE
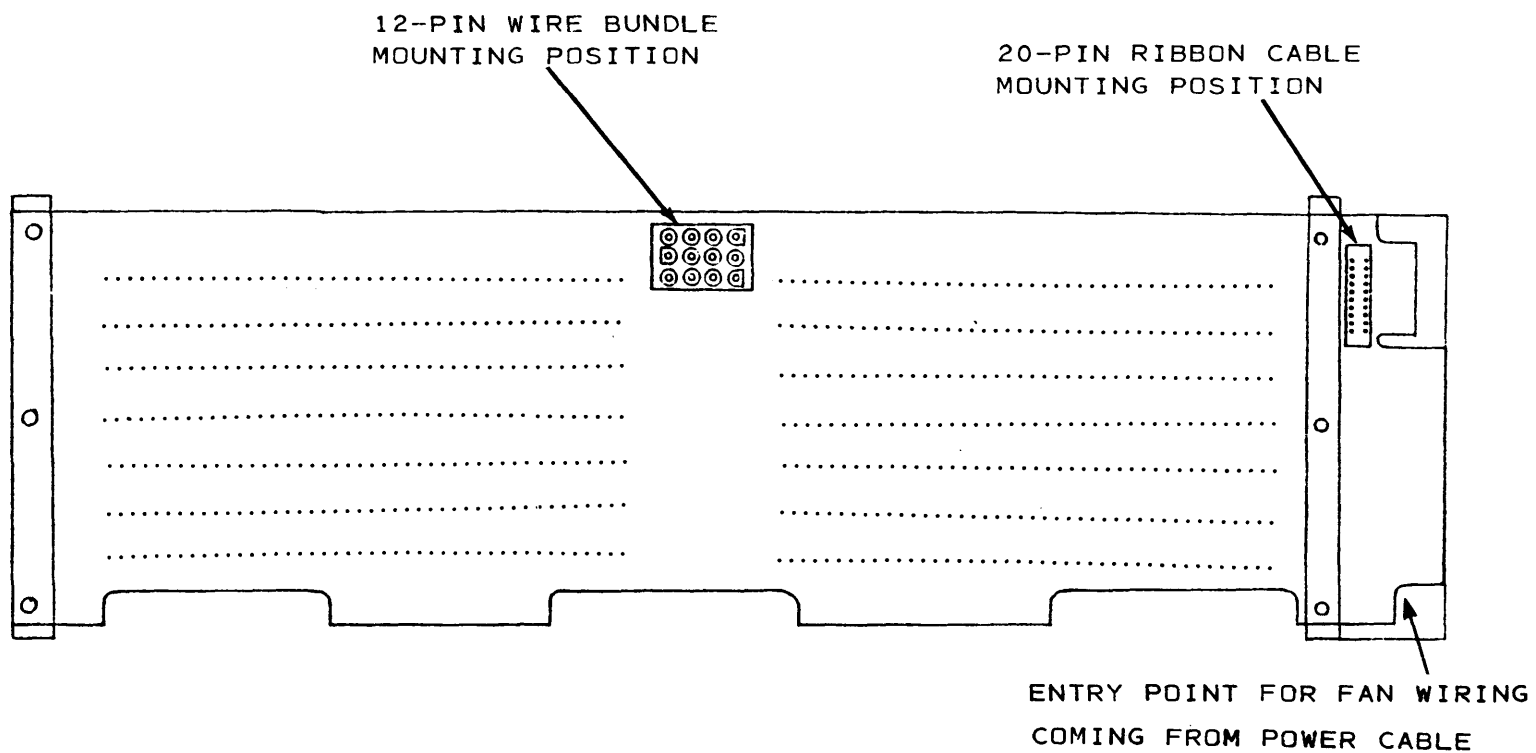
FIGURE 4-5.   POINT 4 PROCESSOR CHASSIS BACKPLANE
CONNECTOR MOUNTING POSITIONS

FUSE BOX

AC PLUG

25-PIN RIBBON CABLE
MOUNTING POSITION

15-PIN WIRE BUNDLE
MOUNTING POSITION
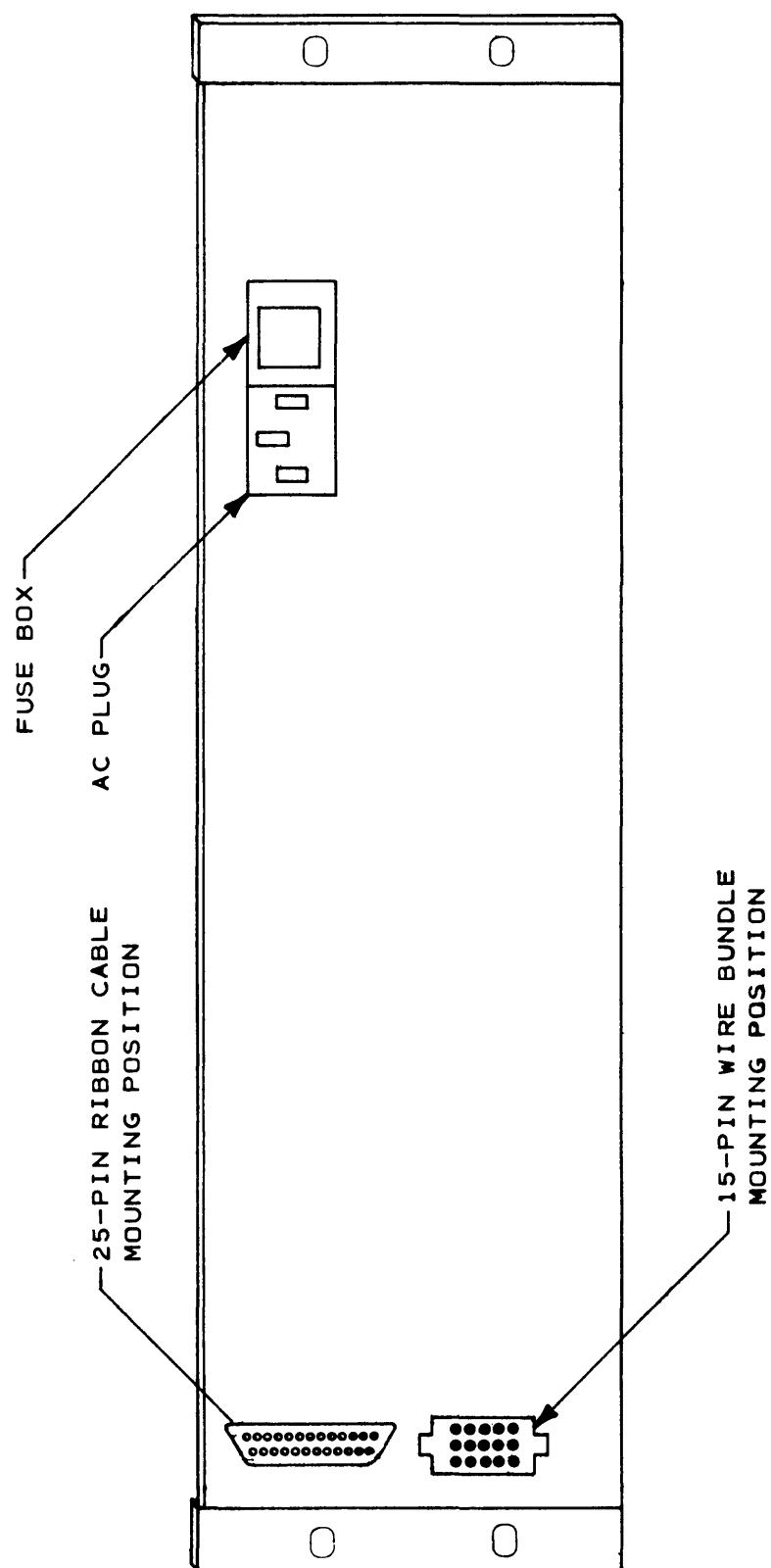
FIGURE 4-6. POWER SUPPLY CHASSIS REAR CONNECTOR MOUNTING POSITIONS

4.4 POWERING-UP THE SYSTEM

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

1. Before connecting the 6-foot ᴀC power cable to the AC power source, turn the processor Mini-panel key switch to the OFF position. See Section 5.3 for processor Mini-panel switch positions.

2. Plug the AC power cable into the AC socket. The AC IN light emitting diode on the power supply Mini-panel should be illuminated. See Section 5.2 for power supply Mini-panel light emitting diode positions.

    If the AC IN light emitting diode does not illuminate, check the following:

    *   Check to ensure that the AC power is properly connected
    *   Check to ensure that the power control cable is properly connected to the processor chassis (See Section 4.3 for cable connection instructions)
    *   Check to insure that the fuse in the power supply fuse box is installed and good

    If the AC IN light emitting diode still does not illuminate, do not proceed. Return the power supply for repair.

3. Turn the Mini-panel key switch to the STANDBY position. If the power supply contains battery backup, the +5 BU, the -5 BU, and the +12 BU light emitting diodes should be illuminated. See Section 5.2 for positions of light emitting diodes on the power supply Mini-panel.

4. Turn the Mini-panel key switch to the ON position. The POWER OK light emitting diode on the processor Mini-panel should illuminate indicating that power is available to the processor chassis. The power supply Mini-panel light emitting diodes should all be illuminated, indicating that all voltages are in tolerance. If any of the light emitting diodes on the power supply Mini-panel do not illuminate, one of the following conditions exists:

    *   The power supply for that voltage is out of tolerance
    *   The light emitting diode is bad

4.5 DIAGNOSTIC CHECKS

4.5.1 Diagnostic Capabilities

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

The Self-Test diagnostic contains the following tests:

1. Halt Instruction Test
2. Compare Instruction Test
3. ALU and Data Bus Test
4. ALU Source Operand Test
5. Exhaustive ALU Instruction Test
6. Page 0 and Base 3 Addressing Modes Test
7. Relative, Base 2, and Indirect Addressing Modes Test
8. Auto Indexing Test
9. Limited I/O Instruction Test
10. Multi-level Indirect Addressing Test
11. Check of 64K Addressing Capability (if installed)
12. Worst-case Memory Test of all Memory Locations

4.5.2 Self-Test Operating Procedures

The following are general procedures for starting the POINT 4 Self-Test. For details on tests executed, expected results, and halt interpretations see the POINT 4 Diagnostics Manual.

4.5.2.1 Normal Self-Test Operation

1. If the front panel has been re-installed, remove the panel again by snapping it off.

2. Locate the Self-Test switch on the front edge of the CPU board. See Figure 4-7 for Self-Test switch location. Press this switch and simultaneously press the APL switch on the processor Mini-panel. This will cause loading of the Self-Test Program into main memory.

3. The first test will verify operation of the Halt instruction. The CPU will halt and the Run light will go out. The CONT switch on the processor Mini-Panel must be pressed to cause the Self-Test to continue execution.

4. After a few preliminary tests the message "EDS POINT 4 SELF-TEST" will be displayed on the master terminal (if a master terminal is in use).
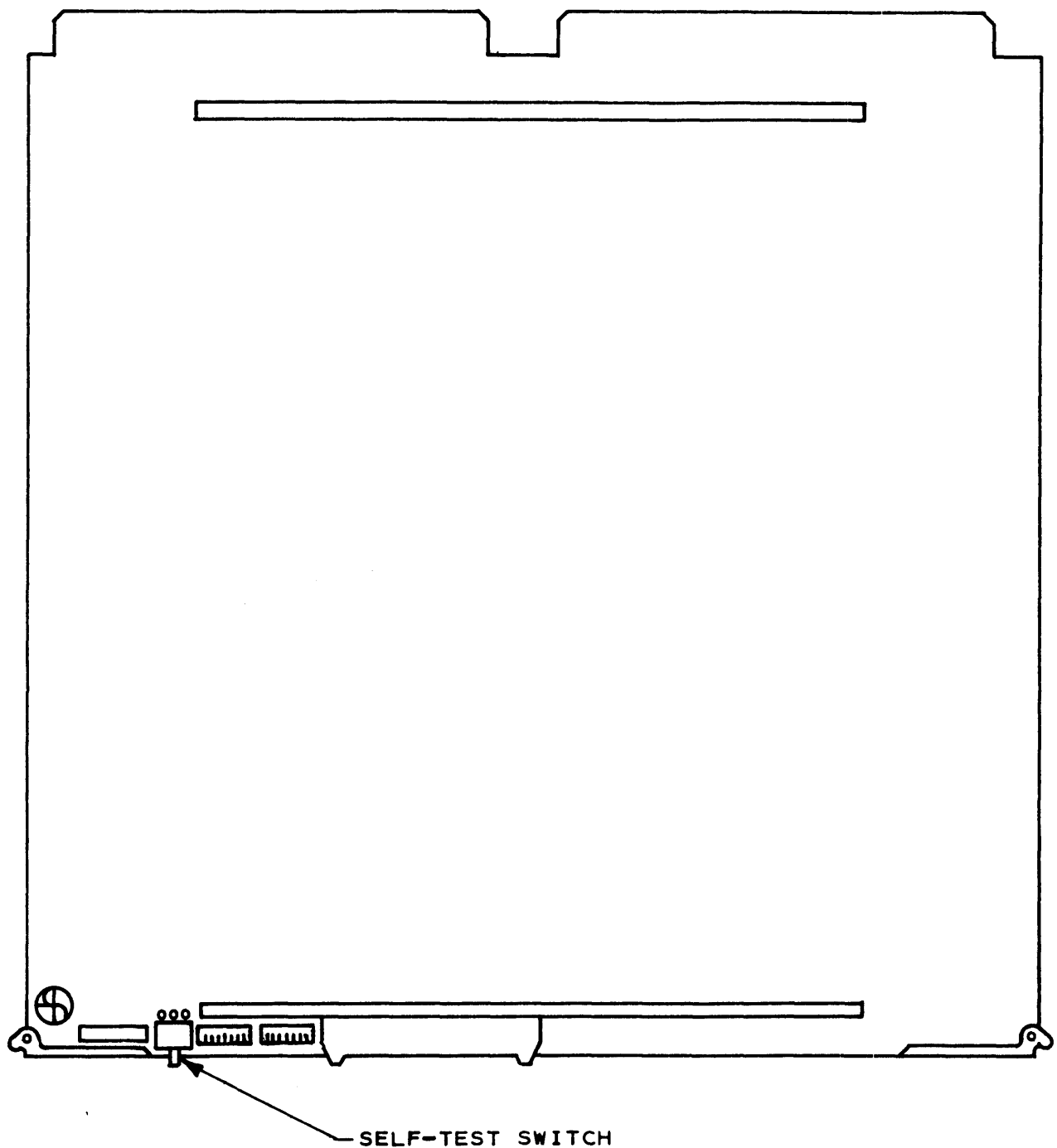
SELF-TEST SWITCH

FIGURE 4-7. POINT 4 SELF-TEST SWITCH LOCATION

5. After testing of all instructions, addressing modes and basic CPU operations, the message "32K CPU OK" or "64K CPU OK" will be displayed on the master terminal, depending on the memory capacity installed.

6. After completion of the worst-case memory test, the message "MEMORY OK" will by typed on the master terminal.

7. Thereafter, each time the Self-Test repeats, a "V" will be displayed on the master terminal.

4.5.2.2 How to Interpret a Halt

If Self-Test halts (RUN light goes out) after the initial Halt, an error is indicated. Refer to the POINT 4 Diagnostic Manual for detailed information on each step of the diagnostic programs, for diagnostic program listings, and for interpretations of halts at various locations.

4.5.2.3 The Self-Test as a Continuous Reliablity Test

Note that the Self-Test may be used either as a one-pass hardware verifier or as a continuous reliability test.

- If a one-pass execution is desired, wait until the first "V" appears, then press the STOP switch.

- If a continuous test is desired, return the front panel to position without pressing the STOP switch. The Self-Test program will continue running until the STOP switch is pressed.

SECTION V

OPERATING PROCEDURES

5.1 GENERAL

This section contains descriptions and explanations of
capabilities and operating procedures for the power supply
Mini-panel and the three types of control units available on the
POINT 4 computer. These control units are: the processor
Mini-panel, the detachable Operator Control Unit, and the Virtual
Control Panel. Controls and indicators are outlined and specific
procedures for performing common types of operations are given.

5.2 POWER SUPPLY "MINI-PANEL"

The POINT 4 power supply chassis houses a set of light emitting
diode indicators for monitoring power supply voltages and battery
backup voltages. The power supply Mini-panel is located on the
letthand side of the POINT 4 power supply chassis. Figure 5-1 is
an illustration of the power supply Mini-panel indicators.
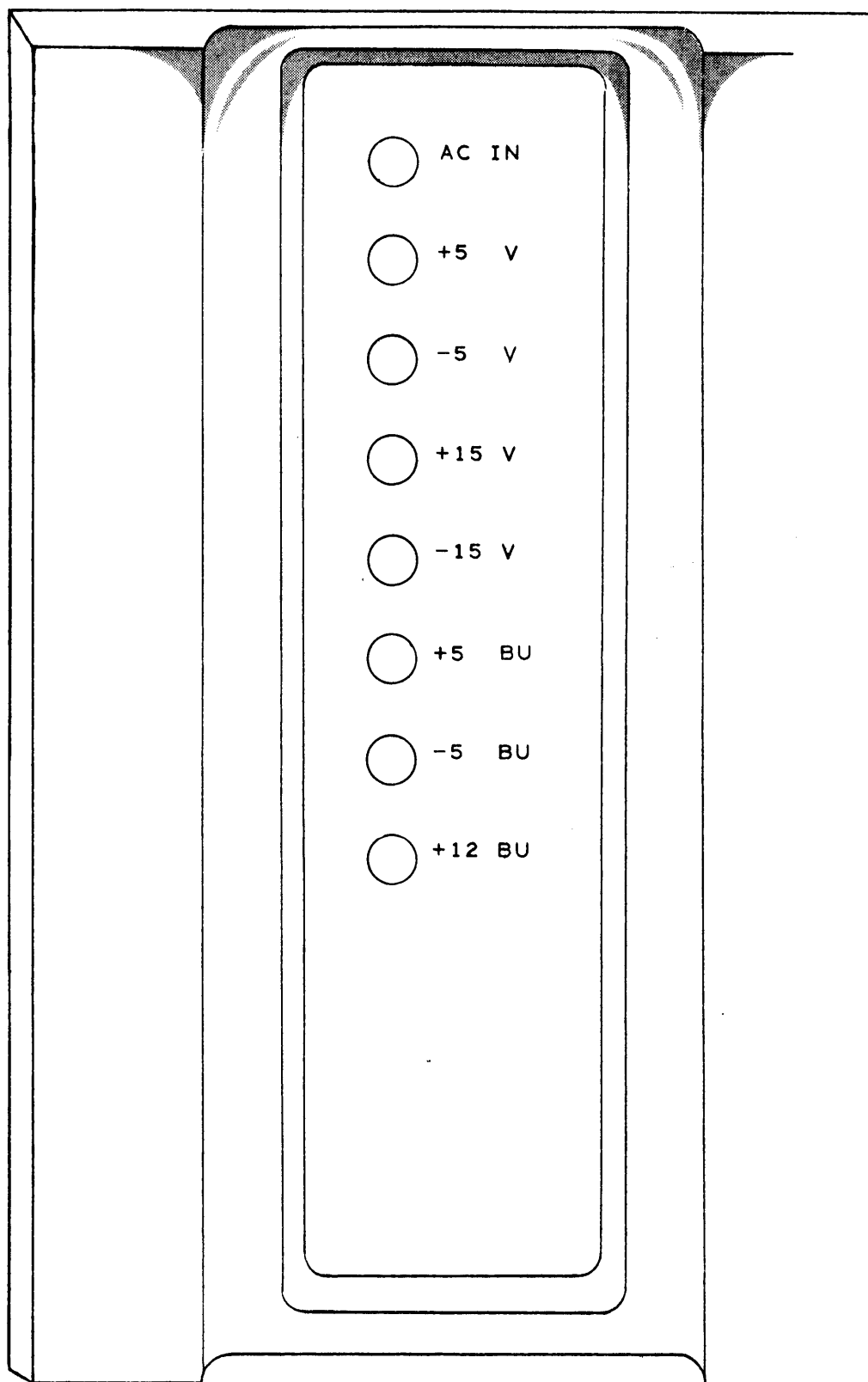
FIGURE 5-1. POWER SUPPLY MINI-PANEL

The Mini-panel indicators when illuminated indicate the following:

Indicator	Meaning
AC IN	Indicates that AC power has been applied to the power supply unit.
+5V	The +5V output voltage is in tolerance. This output voltage is available for user applications.
-5V	The -5V output voltage is in tolerance. This output voltage is available for user applications.
+15V	The +15V output voltage is in tolerance. This output voltage is available for user applications.
-15V	The -15V output voltage is in tolerance. This output voltage is available for user applications.
+5 BU	The +5V battery backup supply is in tolerance. This output voltage is available only to the POINT 4 CPU/memory board. If the battery backup unit is not installed, this light means the same as the +5V light above.
-5 BU	The -5V battery backup supply is in tolerance. This output voltage is available only to the POINT 4 CPU/memory board. If the battery backup unit is not installed, this light means the same as the -5V light above.
+12 BU	The +12V battery backup supply is in tolerance. This output voltage is only available to the POINT 4 CPU/memory board. The +12V supply is operational with or without the battery backup unit.

5.3 PROCESSOR "MINI-PANEL"

The POINT 4 processor chassis houses essential controls and indicators for basic processor control functions. The controls and indicators for this processor Mini-panel are located on the lefthand side of the POINT 4 chassis (see Figure 1-1). There are three types of operating functions on the Mini-panel: processor monitoring indicators, program execution controls and power controls and indicators. Figure 5-2 is an illustration of the processor Mini-panel controls and indicators.
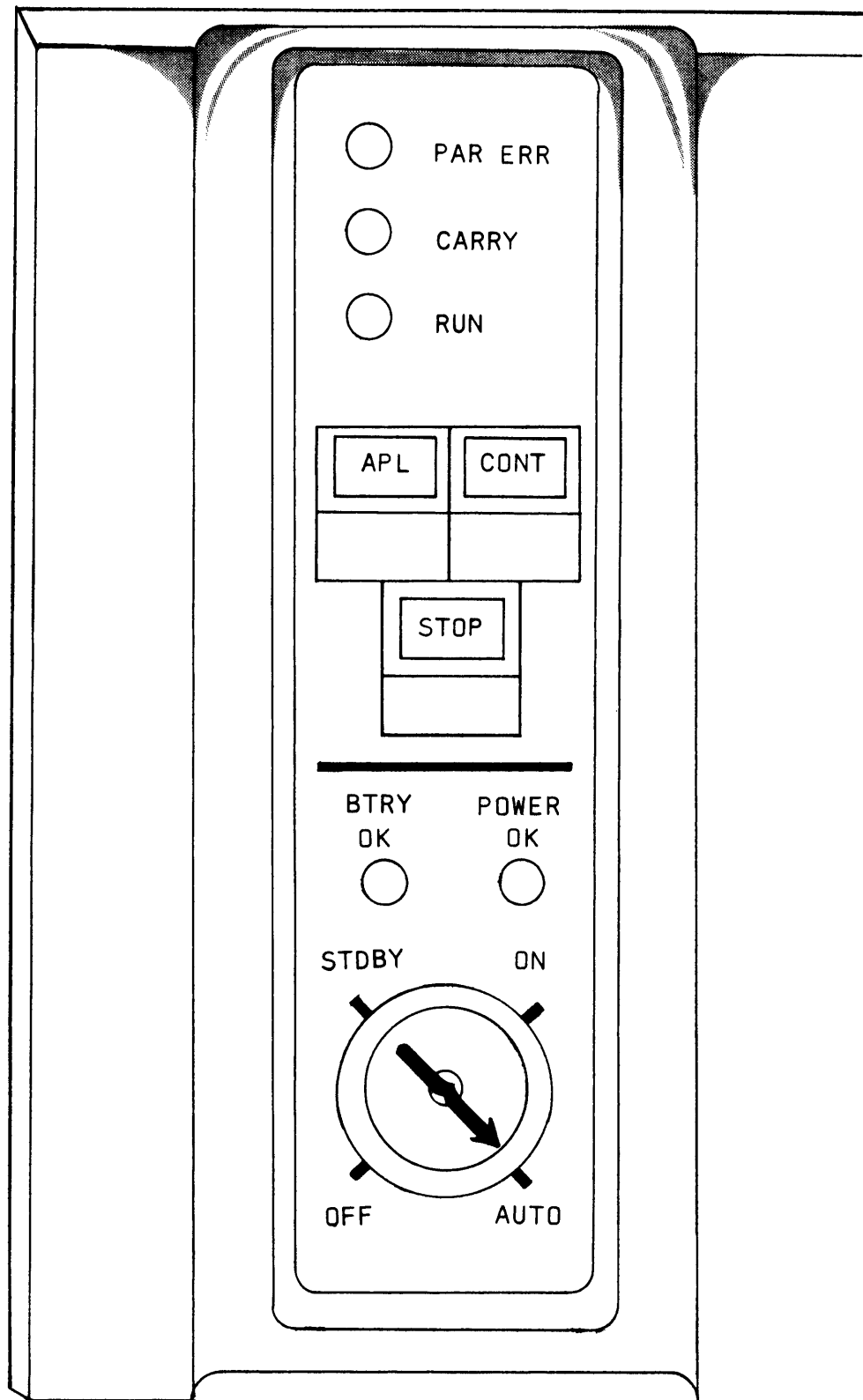
FIGURE 5-2. POINT 4 PROCESSOR MINI-PANEL

5.3.1 Power Controls

The lower half of the Mini-panel is devoted to power control and monitoring. POWER ON is controlled by a four-position key-operated rotary switch. This switch controls the following four 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. The power-fail auto-restart feature (available with the battery backup option) is disabled, and the processor must be started manually whenever AC power is turned ON.
AUTO	Maintains processor power on and places the Mini-panel in the Panel-Off Mode. In this mode all controls on the Mini-panel are disabled, and all indicators remain active. Disabling of the controls prevents interference with the operation of the CPU. The power-fail auto-restart feature is enabled if the battery backup option has been installed. This feature causes the processor to resume operation automatically upon AC power restoration following power failure.
STDBY	Standby: Turns the processor off, leaving the battery backup unit (if installed) operational. Voltages for $\pm$5V and $\pm$15V are removed from the chassis but +5 BU, -5 BU and +12 BU are maintained for CPU slot use only.
OFF	Disables the battery backup unit. This switch position is used for storage or extended power off conditions. The battery charger is kept on as long as the power supply is plugged in.

Also provided are Light Emitting Diode (LED) indicators, which illuminate to indicate an active state, for AC power and battery backup power.

The POWER OK LED can be interpreted as follows:

Power Supply AC IN	Processor Chassis POWER OK	Interpretation
OFF	OFF	Power supply not connected to AC
ON	OFF	Power supply operational but power not available to processor chassis. This condition (if keyswitch is in ON or AUTO) indicates that one of the power supply voltages is out of tolerance (see Power Supply Mini-Panel) or that there is a problem in the cabling between the processor and the power supply.
ON	ON	All power supply voltages are in tolerance and available to the processor chassis

If the Battery Backup option has been installed in the Power Supply chassis, the BATTERY OK light emitting diode will be operational. The BATTERY OK LED has the following interpretations, depending on the state of the AC IN LED on the power supply:

Power Supply AC IN	Processor Chassis BTRY OK	Interpretation
OFF	OFF	AC Power source is off and the Battery Backup Unit has been fully discharged
OFF	ON	The Battery Backup Unit is supplying +5V BU, -5V BU and +12V BU to the CPU/memory board to maintain the contents of memory
ON	OFF	The Battery Backup Unit batteries are being recharged but are not fully charged. The LED indicator will flash ON and OFF as the batteries are reaching full charge.
ON	ON	The Battery Backup Unit batteries are fully charged

See Figure 5-2 for positions of the key-operated rotary switch and the POWER ON and BTRY ON indicators.

5.3.2 Processor Operation Monitoring Indicators

In addition to the power monitoring indicators discussed above, the Mini-panel has three LED indicators performing the following functions:

Indicator	Function
PAR ERR	Indicates that a parity error has occured in a memory read operation. The LED illuminates when the parity error condition is active. The LED indicates that the processor has come to a halt pending operator action. Pressing the CONTinue switch or the APL switch will cause the processor to resume operation and turn off the parity error light. This LED is operational only if the parity error option is installed.
CARRY	Indicates the current state of the processor carry flag. The LED illuminates when the carry flag is set to a 1.
RUN	Indicates that the processor is in normal operation, with one instruction after another being executed. When the processor is stopped, the light goes off.

See Figure 5-2 for locations of the operation monitoring LED indicators.

5.3.3 Program Execution Controls

Three momentary contact switches are available to control program execution in the processor. These switches are enabled in the Panel-On Mode (key switch set to ON position) and disabled in the Panel-Off Mode (key switch set to AUTO postion). These switches are:

Switch	Function
STOP	This switch stops processor operation before executing the next instruction. The processor finishes the current instruction, fetches the next instruction and then stops. The Program Counter points to the next instruction to be executed. NOTE If the processor is caught in an infinite indirect addressing loop which will prevent completion of the instruction, the STOP control will not work. The APL switch must then be used to stop the processor and load MANIP for debugging (see Section 5.5 for use of MANIP).
CONT	This switch causes program execution to resume, starting at the address contained in the Program Counter.
APL	The Automatic Program Load (APL) switch performs two separate functions: 1) The APL switch loads the contents of an octal debugger/manipulator PROM into the top 1000 (octal) words of memory. The debugger/ manipulator is used for access to accumulators and memory. It allows examination and deposit of data for operation monitoring and control. It optionally allows loading of system software from disc or other DMA devices. See Section 5.5 for debugger/manipulator program commands. 2) The APL switch may also be used in combination with the Self-Test switch, located on the front edge of the CPU circuit board, to load the contents of the Self-Test PROM into memory. The Self-Test program performs a complete check of hardware functions and executes a worst-case memory test. It can be used either as a hardware verifier or as a continuous reliability test. See Section 4.5 for a description of the self-test diagnostics. If the CPU is running, it is necessary to press the STOP control before pressing the APL switch. However, if the processor is performing a multi-level indirect addressing instruction, the APL switch will cause the processor to stop without use of the STOP switch.

See Figure 5-2 for locations of the program execution controls.

5.4 OPERATOR CONTROL UNIT (OPTIONAL)

In addition to the basic controls and indicators on the POINT 4
processor chassis, an Operator Control Unit is available which
enhances operator access to the processor. This detachable
control unit can be extended via ribbon cable to any convenient
working surface. The compact control unit contains all switches
and indicators necessary to monitor and control the processor.
(See Figure 1-5 for a photograph of the Operator Control Unit.)

The Operator Control Unit measures 6.12 inches wide, 3.62 inches
high and 1.10 inches deep. It may be mounted on the front panel
at the slot provided in the center of the front panel, or extended
to a convenient working surface by mounting a 6-foot ribbon cable
to the processor PC board.

5.4.1 Operator Control Unit Capabilities

The Operator Control Unit is designed to aid the computer operator
in detecting possible problems in the system and debugging these
problems, and for entering program and data changes into the
system. The operator can monitor activity on the system via eight
light emitting diode indicators. These indicators monitor the
following processor functions: data channel, programmable control
store, high-speed interprocessor bus, 64K word addressing, program
execution, interrupt enabling, carry condition and parity error
detection. Octal displays allow the operator to observe the
contents of memory and accumulators as well as the entries made
via the data entry switches.

Switches provide the operator with the ability to enter data,
access memory to examine and deposit into it, access accumulators
to examine and deposit into them, enable 64K word addressing, and
to control program execution. In addition, an APL switch is
provided.

FIGURE 5-3. OPERATOR CONTROL UNIT
CONTROLS AND INDICATORS

5.4.2 CPU State Indicators

Eight Light Emitting Diode (LED) indicators are used to monitor processor operation, illuminating when the function is active. The following chart describes the function of each LED indicator. See Figure 5-3.

LED Indicator	Function
PAR ERR	The parity error indicator illuminates when a parity error has occurred during a memory read operation. This indicator is enabled only when the parity error detection option has been chosen. The light indicates that the processor has come to a halt pending operator action. The RESET control must be pressed to enable use of the other control unit switches when a parity error has occurred. Pressing the CONTinue button will then cause the processor to resume operation, regardless of the previous parity error. The error may be investigated and/or corrected through use of the examine and deposit capabilities of the Operator Control Unit.
ION	Interrupts are enabled. This LED indicates that the processor will respond to interrupt requests from peripheral devices.
DCH	Data channel is active. This LED indicates that a data channel direct memory access transfer is currently taking place.
HIP	High-speed Interprocessor Bus (HIP) is active. This indicator is enabled only when the HIP option has been included in the system. This LED indicates that communications are taking place between two POINT 4 processors in the system.
PCS	Programmable Control Store (PCS) is active. This indicator is enabled only when the PCS option has been included in the system. This LED indicates that the PCS is currently being used for execution of special macro instructions.

LED Indicator	Function
64K	The 64K Word Addressing Mode is enabled. This indicator is enabled only when 64K words of memory have been installed in the system. The LED indicates that the 64K Word Addressing Mode is enabled rather than the 32K Word Addressing Mode.
RUN	Indicates that the processor is in normal operation with one instruction after another being executed. When the processor is halted, the LED goes off.
CARRY	Indicates the current state of the processor carry flag. The LED is illuminated when the carry flag is set to a 1.

5.4.3 Octal Displays

Two separate six-digit octal displays are provided, one for address (ADDRESS) and one for data (DATA). These are LED displays and are located at the upper right-hand corner of the control unit (see Figure 5-3 for an illustration of Operator Control Unit displays, indicators, and switches). The contents of these displays are controlled by the examine switches and the data entry switches on the control unit.

The address display (ADDRESS) is always equal to the Program Counter (PC). While in RUN, ADDRESS monitors the program by continuously displaying the program counter. When the processor is halted, ADDRESS displays the PC where execution was stopped. Any control unit operation in which an address is entered, incremented, or decremented (EXAM, EXAM NEXT, etc.) will also change the PC and ADDRESS simultaneously.

5.4.4 Data Entry and Processor Manipulation Controls

All data entry and processor control buttons on the Operator
Control Unit are completely sealed, laminated membrane switches.
These buttons are used to monitor and enter data into memory and
accumulators and to control program execution. They serve as
useful system monitoring and problem debugging tools for the
operator or for system programmers.

Eight membrane switches are provided for data entry in octal form.
These buttons are numbered 0, 1, 2, 3, 4, 5, 6, 7 enabling the
operator to make 6-digit octal entries for deposit in accumulators
or memory and entry of memory addresses. The octal digits entered
are shifted into the DATA display. The destination of the octal
entry is controlled by the accumulator deposit buttons for the
accumulators, by EXAM for memory addresses and by the DEPosit and
DEPosit NEXT switches for memory data. The number in the DATA
display may be read by the CPU via use of a READS -,CPU
instruction.

The 16-bit computer word is divided into six octal digits. The
most significant bit of the sixteen bits forms the most
significant octal digit. Since this digit consists of only one
bit, it can have the values 0 and 1 only. Therefore, if a value
greater than 1 is shifted into the most significant bit, that
value will be truncated to a 1-bit number as follows:

- any even number --> 0
- any odd number   --> 1

A CLEAR DATA button is provided to clear "DATA" to 000000.
Corrections may also be made by entering zeros followed by the
correct octal digits. The most significant digit will be shifted
off the left end of the display as new digits are entered. See
Figure 5-3 for positions of the data entry and CLEAR DATA buttons.

Processor access and program execution controls consist of 20
sealed membrane switches. These controls can be grouped by basic
functions into three types: memory access, accumulator access and
program execution. See Figure 5-3 for positions of these controls
on the Operator Control Unit.

Memory access controls (disabled while the processor is in RUN mode) are:

Control	Function
EXAM	Examine: The octal value in the DATA display is moved into the ADDRESS display. The contents of memory at the memory location in the ADDRESS display is then displayed in the DATA display. The PC is set equal to the ADDRESS display. Note that pressing EXAM twice in a row corresponds to indirect addressing.
EXAM NEXT	Examine Next: The address in the ADDRESS display is incremented by 1 and the content of the new memory address is displayed in the DATA display. The PC is set equal to the incremented memory address which is displayed in the ADDRESS display.
EXAM PREV	Examine Previous: The address in the ADDRESS display is decremented by 1 and the content of the new memory address is displayed in the octal DATA display. The PC is set equal to the decremented memory address which is displayed in the ADDRESS display.
DEP	Deposit: Deposits the value in the DATA display into the memory address displayed in the ADDRESS display. The value in DATA may be a value read from memory, an accumulator, or a value entered via the data entry buttons. The PC is left equal to the ADDRESS display.
DEP NEXT	Deposit Next: The address in the ADDRESS display is incremented by 1 and the value in the DATA display is deposited into the incremented memory address. The value in DATA may be a value read from memory, an accumulator, or a value entered via the data entry buttons. The PC is set equal to the incremented address displayed in ADDRESS.
SET 64K	Enable 64K word Addressing: For systems with 64K words of memory installed, this control enables addresses 100000 through 177777. Memory addresses are 16-bits long for 64K word addressing, instead of 15-bits long as in 32K word addressing. In 64K addressing mode multi-level indirect addressing is not permitted.

Access to Accumulators 0-3 is available through the following examine and deposit controls (disabled when the processor is in the RUN mode):

Control	Function
EXAM A0	Examine Accumulator 0: Displays, in the DATA display, the contents of accumulator 0 (A0).
EXAM A1	Examine Accumulator 1: Displays, in the DATA display, the contents of accumulator 1 (A1).
EXAM A2	Examine Accumulator 2: Displays, in the DATA display, the contents of accumulator 2 (A2).
EXAM A3	Examine Accumulator 3: Displays, in the DATA display, the contents of accumulator 3 (A3).
DEP A0	Deposit in Accumulator 0: Deposits in accumulator 0 (A0) the value in the DATA display. The value in DATA may be a value read from memory or an accumulator, or a value entered via the data entry buttons.
DEP A1	Deposit in Accumulator 1: Deposits in accumulator 1 (A1) the value in the DATA display. The value in DATA may be a value read from memory or an accumulator, or a value entered via the data entry buttons.
DEP A2	Deposit in Accumulator 2: Deposits in accumulator 2 (A2) the value in the DATA display. The value in DATA may be a value read from memory or an accumulator, or a value entered via the data entry buttons.
DEP A3	Deposit in Accumulator 3: Deposits in accumulator 3 (A3) the value in the DATA display. The value in DATA may be a value read from memory or an accumulator, or a value entered via the data entry buttons.

The remaining six controls are used for control of program execution. They are:

Control	Function
RESET	Resets all I/O devices on the system to an idle state and clears the processor ION flag and 64K addressing mode.
STOP	This switch stops processor operation before executing the next instruction. The processor finishes the current instruction, fetches the next instruction and then stops. The Program Counter and the ADDRESS display point to the next instruction to be executed. The contents of the DATA display is not changed. NOTE If the processor is caught in an infinite indirect addressing loop, which will prevent completion of the instruction, the STOP control will not work. The RESET switch must then be used to stop the processor.
START	Moves the contents of the DATA display to the Program Counter (PC). The processor then begins normal operation by fetching and executing the instruction at the address just loaded into the PC. The RUN light will illuminate.
CONT	This switch causes program execution to resume, starting at the address contained in the Program Counter (and in the ADDRESS display).
INST STEP	Causes the processor to execute one instruction cycle beginning at the address displayed in the ADDRESS display. When the instruction cycle is completed, the processor stops. The ADDRESS display contains the new value of the program counter. The information displayed in the DATA display depends on the type of instruction, as follows: (see next page)

Control	Function	

INST STEP (cont.)	Instruction	Data Displayed
	LDA	Data that was fetched from memory
	STA	Data that was written into memory
	ISZ, DSZ JMP or JSR	The octal value of the next instruction
	Arithmetic/ Logic	Operand in Destination Accumulator after instruction execution is complete
	Input/ Output	Data that was transferred (except in skip instructions, which display the data in A0)

APL   The Automatic Program Load (APL) switch performs two separate functions:

1) The APL switch loads the contents of an octal debugger/manipulator PROM into the top 1000 words of memory. The debugger/manipulator serves as a virtual control panel and operates through the master terminal. It offers several higher-level functions than the operator control unit, such as memory searches, moves, byte addressing, virtual addressing, etc. It also allows loading of system software from disc or other DMA devices. See Section 5.5 for a description of debugger/manipulator commands.

2) The APL switch may also be used in combination with a Self-test switch, located on the front edge of the CPU circuit board, to load the contents of the Self-test PROM into memory. The Self-test program performs a complete check of hardware functions and executes a worst-case memory test. It can be used either as a hardware verifier or as a continuous reliability test. See Section 4.5 for a description of Self-test operation.

If the CPU is running, it is necessary to press the STOP control before pressing the APL switch.

5.5 "VIRTUAL" CONTROL PANEL

The POINT 4 has the ability to do all front panel operations plus many extra system monitoring functions from a master terminal. This feature is designed for use by programmers to debug system problems and to manipulate the contents of registers and memory. The feature is implemented in a system program called "MANIP" which is loaded into RAM from a PROM when the APL switch is pressed.

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 ":" which is also preceded by an octal parameter) and ending with a RETURN.

The following functions are provided (the number in parentheses indicates the number of parameters required for that particular function):

Code	Function	Parameters Required
A	Type initial PC, accumulators and carry flip-flop	(0)
C	Change accumulator or carry flip-flop	(2)
D	Dump (octal, word or byte)	(1 or 2)
E	Enter octal into sequential locations	(1 or 2)
F	Set up an address offset	(0, 1 or 2)
I	Input ASCII string on master terminal	(1 or 2)
J	Jump with accumulators and carry restored	(1 or 2)
K	Store a constant in a block of memory	(3)
M	Move a block in memory	(3)
N	Search memory for not-equal (with mask)	(3 or 4)
O	Output ASCII string on master terminal	(1 or 2)
P	Program load from disc	(0 or 1)
R	Read paper tape from TTY or PTR	(0 or 1)
S	Search memory for constant using a mask	(3 or 4)
V	Verify paper tape (TTY or PTR)	(0 or 1)
X	Calculate a checksum for a block of memory	(2)
Y	Set up an output delay after each RETURN (for proper scrolling)	(1)
:	Examine or deposit into a specified location	(2)

---

*For those who are familiar with EDSI'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, and (4) Ctrl H/Ctrl A (backspace) capability. MANIP occupies only 1000 (octal) words of memory, while DBUG occupies 3000 (octal) words of memory.

These functions are described in detail in the following
subsections.

MANIP normally occupies the following memory locations:

Memory Size	Location of MANIP
32K words	77000 through 77777
64K words	177000 through 177777

Location 77000 or 177000 respectively is reserved for saving the
initial value of the program counter (PC), that is, the value of
PC where the CPU had halted 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 (except in I mode). 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 in entering an octal value, type a few
    zeros followed by the correct octal number, as MANIP only
    uses the last six octal digits typed in for the octal
    word.

5.5.1 Addressing Modes

For many commands, MANIP allows either word or byte addressing,
using either real memory addresses or "offset" (virtual) memory
addresses based on an offset that has been previously entered (via
F command). MANIP also is designed to allow addressing up to 64K
words of memory. This is accomplished by having two word
addressing modes (real and virtual), and three byte addressing
modes (one virtual plus two real modes: lower 32K and upper 32K).

These modes are invoked by the optional second parameter "a" shown
for commands D, E, I, J and O (except that J does not permit byte
addresses). For those commands where no "a" parameter is shown,
the addressing mode (if any) is the same as "omitted" below, i.e.,
word address including offset, if any.

"a" Parameter	Meaning
omitted	word address, including offset, if any
0	word address, absolute
1	byte address, using offset, if any
2	byte address, lower 32K
3	byte address, upper 32K

5.5.2 Command Descriptions

A MANIP command consists of a single letter which is the command identifier and parameters which specify addressing modes, memory addresses and data input. All parameters must be entered in octal form. The letters x, y, z, a, m, and n are used on the following pages to represent octal parameters. Press the RETURN key after entering any command.

Command & Parameters	Definition
A	Causes the initial value of PC (program counter) saved in the 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 screen.
Cx,y	Change accumulator or carry flip-flop. • If x is 0, 1, 2, or 3, then y is stored as the saved value for accumulator x (A0, A1, A2, A3, respectively). • If x is 4, then the saved value of the carry flip-flop is set to 0 or 1 according as y is 0 or not. • If x is greater than 4 and an address offset has been established (see "F" command), x is interpreted as a real address using the offset previously established, and typed out on the master terminal. The y parameter is not used in this case. • Parameter Description "x" - 1 octal digit 0-7 "y" - 1 octal word
Dx,a	Dump memory in octal, beginning at location x, using addressing mode a. Eight words (or bytes if a byte address mode is used) are typed per line, with the address of the first word (byte) at the beginning of each line. • Parameter Description "x" - an octal number representing a 16-bit memory address "a" - one octal digit (0-3 or blank) representing an addressing mode

Command & Parameters	Definition

Ex,a	Enable entry at address x, using address mode a. The address (changed to a word address if it was a byte address) is printed, followed by a colon; an octal value may then be entered into the memory location, followed by a RETURN. The next address (x+1) will then be printed and opened for entry. Entry may be continued into sequential address locations until ESC is pressed to terminate entry.
	● If no entry is typed in before the RETURN, the present contents of the opened location is typed out in octal form to allow examination of a value before entering. If another RETURN is then typed, again without an entry, the next address will be printed and opened for entry.
	● If "^" is typed instead of RETURN, the previous address is typed and opened. This last feature is convenient for confirming an entry just typed in.
	● Parameter Description
	  "x" – an octal number representing a 16-bit memory address
	  "a" – one octal digit (0-3 or blank) representing a memory addressing mode

Fx,y	Establish an address offset (i.e., a fixed difference between real memory address on the one hand and addresses as entered and listed in MANIP on the other). The difference x-y is added to addresses entered, and subtracted from memory addresses before listing. If y is not entered then it is assumed to be zero. Whenever a nonzero offset is established, an F is typed at the beginning of each line. To revert to a real memory addressing, type F0.
	● Parameter Description
	  "x" – from 1 to 6 digits representing a real memory address
	  "y" – from 1 to 6 digits representing the listing address which is equivalent to the address specified in "x"

F	Save the current offset value, and reinstate previous offset that was in effect before the current one was established. Types the offset being reinstated. This makes it convenient to toggle back and forth between two different offsets (or one offset and real memory addressing).

Command & Parameters	Definition

Ix,a
text
	Input ASCII. Characters following the RETURN are stored in memory starting at location x (using address mode a), two characters per word (with left-right packing). Input is terminated by pressing ESC, which causes a zero byte or word to be stored.
	● Parameter Description
	  "x" - an octal number representing a 16-bit memory address
	  "a" - one octal digit (0-3 or blank) representing a memory addressing mode

Jx,a	Jump to location x (using addressing mode a) with accumulators and carry stored.
	● Parameter Description
	  "x" - an octal number representing a 16-bit memory address
	  "a" - one octal digit (0-3 or blank) representing a memory addressing mode

Kx,y,z	Store the octal constant z in locations x through y, inclusive.
	● Parameter Description
	  "x" - an octal number representing a 16-bit beginning memory address
	  "y" - an octal number representing a 16-bit ending memory address
	  "z" - an octal number representing a constant.

Mx,y,z	Move block in core. Locations x through y, inclusive, are moved to the area starting at location z.
	● The source and destination areas may overlap in either direction without bad effects.
	● May be used to move MANIP itself as long as the destination area does not overlap the source area.
	● Parameter Description
	  "x" - an octal number representing a 16-bit beginning memory address
	  "y" - an octal number representing a 16-bit ending memory address
	  "z" - an octal number representing the 16-bit beginning memory address of the new location

Command & Parameters	Definition

Nx,y,z,m	Search for not-equal (same as Sx,y,z,m except that it searches for a not-equal condition). Search locations x through y, inclusive, for values not equal to constant z. Each word is first ANDed with mask m before comparison with z.
	● If m is not entered it is assumed to be 177777.
	● The use of the mask is best explained by an example: the command Nx,y,0,170000 will search locations x through y for any value greater than 7777. When such a value is found, its address and contents are typed in octal form on the CRT.
	● Parameter Description
	  "x" – an octal number representing a 16-bit beginning memory address
	  "y" – an octal number representing a 16-bit ending memory address
	  "z" – an octal number representing a constant
	  "m" – an octal number representing a mask; or a blank which defaults the value to 177777

Ox,a	Output ASCII. The contents of memory starting at location x (using address mode a) are typed out as text, two characters per word. Output is terminated if a zero byte is encountered.
	● Parameter Description
	  "x" – an octal number representing a 16-bit memory address
	  "a" – one octal digit (0-3 or blank) representing a memory addressing mode

Px	Program load from disc or other DMA devices. Performs standard "bootstrap" APL function (i.e., gives an NIOS instruction with device code x and then idles at location 377 waiting for the disc to overwrite that location). If x is omitted, reads the mini-switches at the front edge of the CPU board and uses their contents as the device code. ● Parameter Description "x" – a 2-digit octal number representing the I/O device from which the program is to be loaded

Command & Parameters	Definition
Rx	Read punched paper tape from the master teletype if x is omitted or is zero, or from the high-speed paper tape reader (device code 12) if x=1. • If a checksum error occurs, further reading is stopped, and the address where the error occurred is typed out. • If the tape contains an end block with a starting address, the computer will jump to the starting address. • Parameter Description "x" - one octal digit representing the following: 0 or blank - master teletype 1 - high-speed paper tape reader
Sx,y,z,m	Search locations x through y, inclusive, for the constant z. Each word is first ANDed with mask m before comparison with z. • If m is not entered, it is assumed to be 177777. • The use of the mask is best explained by an example: the command Sx,y,60025,160077 will search locations x through y for any I/O instruction for device 25. When a comparison is found, its address and contents are typed in octal form on the CRT. • Parameter Description "x" - an octal number representing a 16-bit beginning memory address "y" - an octal number representing a 16-bit ending memory address "z" - from 1 to 6 digits representing an octal constant "m" - from 1 to 6 digits representing an octal mask; or a blank which defaults the value to 177777
Vx	Verify paper tape from TTY (x = 0 or none) or PTR (x = 1). . • If a verification error is found, its address is typed out. • Parameter Description "x" - one octal digit representing the following: 0 or blank - master teletype 1 - paper tape reader

Command & Parameters	Definition

Xx,y	Calculate and type the checksum over memory locations x through y. Utilizes a "revolving" checksum (using a SUBL 0,1 instruction, with A0 = each word from x through y, and Al = accumulating checksum; initially 0). This insures that if two words in memory are swapped, the swap will be detected by the checksum. Useful for determining if any word in memory has changed.
	● Parameter Description
	  "x" – an octal number representing a 16-bit beginning memory address
	  "y" – an octal number representing a 16-bit ending memory address

Yx	Set up a RETURN delay, required on some CRT's for proper scrolling. After each subsequent carriage return/line feed, MANIP counts up an accumulator from x to 0 before proceeding. For maximum delay set x = 0, for no delay set x = 177777. ● Parameter Description "x" – from 1 to 6 octal digits representing a return delay value

x:y	The octal value y is stored at location x, and the next cell is opened. See the "E" command for more information.
	● Parameter Description
	  "x" – an octal number representing a 16-bit memory address
	  "y" – from 1 to 6 digits representing an octal value

See Appendix D for a summary chart of MANIP commands.

SECTION VI

OPTIONAL FEATURES

6.1   INTRODUCTION

This section covers features which can be added to the POINT 4
computer to enhance its performance. Included are discussions of
architecture, operating instructions and special procedures
related to the following options:

- Battery Backup
- Operator Control Unit
- Data Channel Options
- Programmable Control Store
- High-Speed Interprocessor Bus

6.2   BATTERY BACKUP

The battery backup option provides three features:

- Memory contents protection for at least two hours in case of |
  power failure
- Power Monitor Auto-Restart after resumption of power service
- Standby mode, which removes ±5V and ±15V power but maintains
  +5V BU, -5V BU, and +12V BU power to the CPU slot for memory
  protection

6.2.1   Battery Backup Voltages

The battery unit is maintained in a charged state by the power
supply as long as the unit is plugged in, and AC power is
available. The battery backup module consists of three 6 volt
sealed lead acid batteries connected in series to provide 18 volts
to the main power supply.

From these 18 volts the power supply provides the following voltages to the CPU/memory board and Mini-panel in case of power failure:

Voltage	Supplied to
+ 5V BU	CPU Board Memory Refresh logic, Mini-panel
- 5V BU	For Memory on CPU Board
+12V BU	For Memory on CPU Board

Battery backup voltages can be monitored on the power supply chassis Mini-panel, by the light emitting diodes (LED) labeled as BU voltages. Illumination of these LED indicators means these voltages are in tolerance. If one or more LED indicator is not illuminated one of the following two conditions exists:

AC Power	LED Indicator OFF
ON	Backup power supply(s) is out of tolerance
OFF	Batteries are discharged and memory contents are lost

6.2.2 Power Monitor Auto-Restart Handling

When AC power fails an unmaskable interrupt is produced. A Power-fail interrupt is indicated if the CPU DONE flag is set to a 1. Two Input/Output CPU instructions are associated with a Power-fail interrupt. They are:

Instruction	Function
SKPDN CPU	Skip if a Power-fail interrupt has occurred
SKPDZ CPU	Skip if a Power-fail interrupt has not occurred

One of these instructions is used to enter a routine for handling Power-fail conditions. The routine should do the following:

- Save the accumulators, carry flag, and program counter where the interrupt occurred.
- Do any other cleanup that may be necessary.
- Put an appropriate auto-restart instruction in memory location 0.
- Halt.

While the Power-fail condition exists, the Backup Battery Unit retains the contents of memory for at least two hours (if the batteries were fully charged when the Power-fail occurred).

When AC power is restored, the CPU will begin execution at memory location 0, if the processor Mini-panel key switch is set to AUTO. If the key switch is not set to AUTO, the processor will halt, with the Program Counter equal to 0, pending operator action.

6.3 OPERATOR CONTROL UNIT (OCU)

An extensive discussion of Opertor Control Unit capabilities and
operating procedures is included in Section 5.3. There are,
however, special instructions necessary for attaching the Operator
Control Unit to the processor chassis.

6.3.1 Attachment to the Front of Processor Chassis

The Operator Control Unit can be attached to the center of the
processor front panel for location with the processor. The unit
has an edge connector protruding from the upper rear of the unit.
This connector inserts directly into the slot provided on the
center of the front panel, mating with a socket behind the front
panel. This socket connects via ribbon cable to a socket on the
front edge of the CPU board. See Figure 6-1 for an illustration
of this connection.

6.3.2 Extension of Operator Control Unit

The Operator Control Unit can be extended via ribbon cable to any
convenient working surface. Extension is via a six-foot ribbon
cable. The connector on the rear of the Operator Control Unit
attaches to a socket on one end of the ribbon cable. The ribbon
cable threads through a slot on the lower edge of the center of
the front panel and the other connector connects to the socket on
the front edge of the CPU board. See Figure 6-2 for an
illustration of this attachment.
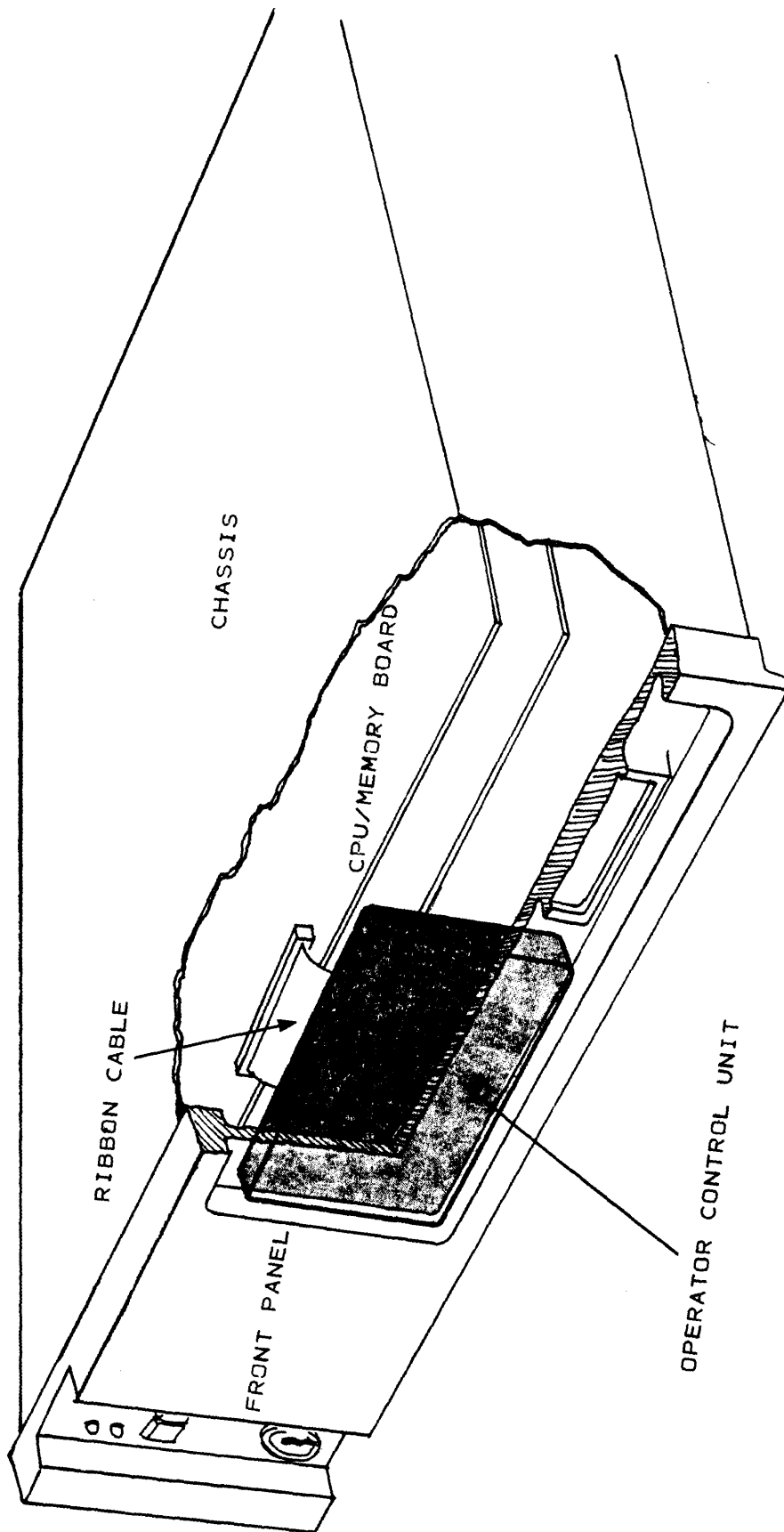
FIGURE 6-1. OPERATOR CONTROL UNIT ATTACHMENT TO FRONT PANEL

Labels in figure:
CHASSIS
CPU/MEMORY BOARD
POINT4
FRONT PANEL
RIBBON CABLE
OPERATOR CONTROL UNIT
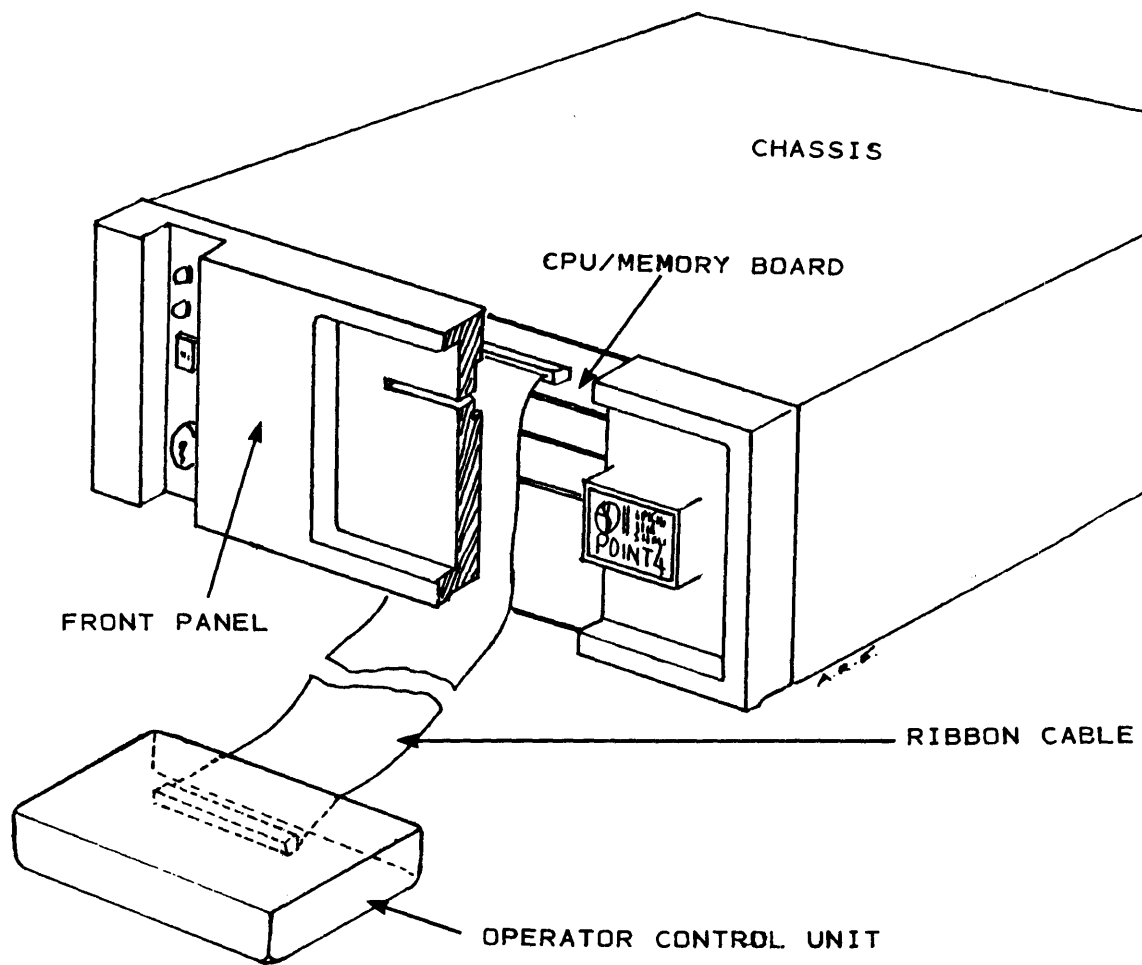
FIGURE 6-2. OPERATOR CONTROL UNIT REMOTE ATTACHMENT

6.4 DATA CHANNEL OPTIONS

Two options are available for Data Channel speed selection. They
are:

        Standard Data Channel
           Input   - 1100 nanoseconds |
           Output - 1700 nanoseconds |

        High-Speed Data Channel
           Input   - 900 nanoseconds |
           Output - 1300 nanoseconds |

6.4.1 Selection of the Data Channel

CPU logic tests input to PIN A93 (top slot) to determine which
speed has been enabled. (See Figure 3-2, Section III for
backplane pin assignments.) If PIN A93 is jumpered to ground the
data channel will operate at Standard Data Channel speeds. If PIN
A93 is left open the data channel will operate at High-speed Data
Channel speeds.

6.4.2 Criterion for Data Channel Speed Selection

The high-speed Data Channel on the POINT 4 does not have nearly as
stringent requirements for controller timing as on some
competitive computers. With the POINT 4, the controller is given
about 200 nanoseconds from the start of DCHA to put its address on
the I/O bus. Similarly, the POINT 4 also allows about 200
nanoseconds from the start of DCHI before it requires the input
data on the I/O bus. (See Section 3.6.3 on Data Channel Transfer
Timing.) In some competitive computers as little as 75
nanoseconds is allowed for putting addresses or data on the I/O
bus.

The result of this relatively long access time to the I/O bus is
that many DMA controllers which cannot operate on a high-speed
data channel with competitive computers can operate on the POINT 4
high-speed Data Channel. Therefore the High-speed Data Channel
should be used in all cases unless the system includes a
controller with specifications that indicate inability to
accommodate these speeds. If in doubt try High-speed. If
operation is then inconsistent, switch to Standard Data Channel by
jumpering PIN A93 to ground. If operation is consistent at the
Standard Data Channel speeds, it will be necessary to maintain
operation at these speeds.

APPENDICES

APPENDIX A

VON NEUMANN MAP OF POINT 4 COMMANDS

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

  0 |   0    0 JMP  0    0
M     |   0    0 JSR  0    1
E     |   0    0 ISZ  1    0
M  |   0    0 DSZ  1    1
.     |   0         1                I          X              DISPLACEMENT
R         LDA               0  -    00  ABS
E                    ACC    1  a    01  REL
F     |   1         0                   10  B2
.         STA       00  A0              11  B3
                    01  A1
                    10  A2
                    11  A3

  0 |   1    1              000  NIO        CTRL
                            001  DIA      XFER  SKP      DEVICE CODE
          I/O               010  DOA     00  -  BN
                            011  DIB     01  S  BZ
                            100  DOB     10  C  DN
                            101  DIC     11  P  DZ
                            110  DOC
                            111  SKP

  1 |
A         ACS      ACD      OPCODE      SH       CY      NL        SK
L
U    00  A0   00  A0   000  COM   00  -    00  -    0 -    000   -
     01  A1   01  A1   001  NEG   01  L    01  Z    1 #    001  SKP
I    10  A2   10  A2   010  MOV   10  R    10  O           010  SZC
N    11  A3   11  A3   011  INC   11  S    11  C           011  SNC
S                     100  ADC                            100  SZR
T                     101  SUB                            101  SNR
R                     110  ADD                            110  SEZ
.                     111  AND                            111  SBN

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

POINT 4 INSTRUCTION REFERENCE CHART

(OCTAL TO SYMBOLIC CONVERTER)

ARITH/LOGIC	MEMORY REF.	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
DEST.	INDIRECT	100 S
0 0	2000 @	200 C
4000 1		300 P
10000 2	ADR. MODE	
14000 3	0 ABS	I/O SKIP
	400 REL	63400 SKPBN
SHIFT	1000 BASE 2	63500 SKPBZ
100 L	1400 BASE 3	63600 SKPDN
200 R		63700 SKPDZ
300 S	DISPLACEMENT	
	0-177 POS.	DEVICE CODE
CARRY	200-377 NEG.	10 TTI
20 Z	EXC.ABS)	11 TTO
40 O		12 PTR
60 C		13 PTP
	SPECIAL ARITH.	14 RTC
NO-LOAD	TESTS	17 LPT
10 #	101014 SKZ	
	101015 SNZ	SPL. CPU INST.
SKIP COND.	101112 SSP	60177 INTEN
1 SKP	101113 SSN	60277 INTDS
2 SZC	102032 SGE	60477 READS
3 SNC	102033 SLS	61477 INTA
4 SZR	102414 SEQ	62077 MSKO
5 SNR	102415 SNE	62677 IORST
6 SEZ	102432 SGR	63077 HALT
7 SBN	102433 SLE	

APPENDIX C
ASCII CODE CHART

7-BIT OCTAL CODE	CHARACTER	7-BIT OCTAL CODE	CHARACTER	7-BIT OCTAL CODE	CHARACTER	7-BIT OCTAL CODE	CHARACTER
000	NUL	040	BLANK	100	@	140	`
001	SOH (CONTROL A)	041	!	101	A	141	a
002	STX (CONTROL B)	042	"	102	B	142	b
003	ETX (CONTROL C)	043	#	103	C	143	c
004	EOT (CONTROL D)	044	$	104	D	144	d
005	ENQ (CONTROL E)	045	%	105	E	145	e
006	ACK (CONTROL F)	046	&	106	F	146	f
007	BEL (CONTROL G)	047	'	107	G	147	g
010	BS (CONTROL H)	050	(	110	H	150	h
011	HT (CONTROL I)	051	)	111	I	151	i
012	LF (CONTROL J)	052	*	112	J	152	j
013	VT (CONTROL K)	053	+	113	K	153	k
014	FF (CONTROL L)	054	,	114	L	154	l
015	CR (CONTROL M)	055	–	115	M	155	m
016	SO (CONTROL N)	056	.	116	N	156	n
017	SI (CONTROL O)	057	/	117	O	157	o
020	DLS (CONTROL P)	060	0	120	P	160	p
021	DC1 (CONTROL Q)	061	1	121	Q	161	q
022	DC2 (CONTROL R)	062	2	122	R	162	r
023	DC3 (CONTROL S)	063	3	123	S	163	s
024	DC4 (CONTROL T)	064	4	124	T	164	t
025	NAK (CONTROL U)	065	5	125	U	165	u
026	SYN (CONTROL V)	066	6	126	V	166	v
027	ETB (CONTROL W)	067	7	127	W	167	w
030	CAN (CONTROL X)	070	8	130	X	170	x
031	EM (CONTROL Y)	071	9	131	Y	171	y
032	SUB (CONTROL Z)	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	F SEP	074	<	134	\	174	|
035	G SEP	075	=	135	]	175	}
036	R SEP	076	>	136	^	176	~
037	U SEP	077	?	137	_	177	DEL

APPENDIX D
VIRTUAL FRONT PANEL COMMANDS

⟶	APL does Auto-Boot if mini switches set to 200 + device code
A	Display PC, A0, A1, A2, A3, and carry.
Cx,y (x<5)	Change accumulator x (or carry if x=4) to value y
Cx (x>4)	Convert real address x to virtual
Dx*	Dump memory in octal, beginning at address x
Ex*	Enable entry at address x
Fx,y	Establish offset x-y, where x=real memory address, y=virtual (listing) address.
Ix*	Input ASCII (2 characters/word), starting at address x. Press ESC to terminate input.
Jx	Jump to location x, after restoring accumulators and carry.
Kx,y,z	Store constant z in memory locations x through y.
Mx,y,z	Move memory block x through y to location z.
Nx,y,z,m	Search memory x through y for not-equal to z, using mask m (optional)
Ox*	Output memory in ASCII, starting at location x, until a 0 byte is encountered.
Px	Program load from DMA device code x. If x omitted reads switches.
R/R1	Read binary paper tape from TTY (R) or PTR (R1).
Sx,y,z,m	Search memory x through y for the value z, using mask m (optional).
V/V1	Verify paper tape, from TTY (V) or PTR (V1).
Xx,y	Calculate checksum over x through y.
Yx	Set up a delay after each CR/LF. x=0 for maximum delay; x=177777 for none.
x:y	Enter value v at address x, and open next cell for entry.
↑	Open previous cell for entry.

*Address x may be followed by mode designator 0,1,2, or 3:

Mode	Meaning
None	Word address, including "F" offset, if any
0	Word address, absolute
1	Byte address, using offset, if any
2	Byte address, lower 32K
3	Byte address, upper 32K

APPENDIX E

PROGRAMMING EXAMPLES

I. NUMBER HANDLING

Generating Numbers

There are 6 numbers that can be generated with a single
instruction:

```
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
```

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 what instructions to use to test for the 20 different sets
of numbers. The skip condition can be changed from a zero-test to
a non-zero-test to obtain the complement 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
NEGO	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

	INSTRUCTION	TESTS FOR
1 VALUE	MOV 0,0,SZR	0
	INCZL 0,0,SZR	77777
	NEGZL 0,0,SZR	100000
	COM 0,0,SZR	-1
2 VALUES	MOVZR 0,0,SZR	0,1
	COMZR 0,0,SZR	-1,2
	MOVZL 0,0,SZR	0,100000
	COMZL 0,0,SZR	77777,-1
32K	MOVL 0,0,SZC	≥0
	INCL 0,0,SZC	-1≤x≤77776
	NEGL 0,0,SZC	-77777≤x≤0
	MOVR 0,0,SZC	EVEN
	ADDL 0,0,SZC	2D BIT = 0
32K +1	MOVZR 0,0,SEZ	EVEN OR 1
	NEGZR 0,0,SEZ	EVEN OR -1
	INCZR 0,0,SEZ	ODD OR 0
	COMZR 0,0,SEZ	ODD OR -2
32K +1	MOVZL 0,0,SEZ	≥0 OR 100000
	COMZL 0,0,SEZ	<0 OR 77777
	NEGZL 0,0,SEZ	≤0

The number columns of the matrix (from 0 upward): 0, 1, 2, 3, …, 37776, 37777, 40000, 40001, …, 77776, 77777, 100000, 100001, …, 137776, 137777, 140000, 140001, …, 177774=-4, 177775=-3, 177776=-2, 177777=-1. Dots indicate which numbers are contained in each instruction's test set.

II. BIT TESTING

Any bit in a word can be tested in at most three instructions, without requiring another accumulator. Three bit positions can be tested in 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 the instructions to use.

```
    0 | 1   2   3 | 4   5   6 | 7   8   9 | 10  11  12 | 13   14   15
   ┌───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┐
   │ 1 │ 1 │ 2 │ 2 │ 3 │ 3 │ 3 │ 2 │ 2 │ 2 │ 2 │ 3 │ 3 │ 3 │ 2 │ 1 │
   └───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┘
```

                                                              SKO 0,0

                                                         MOVR 0,0
                                                         SKO 0,0

                                                    MOVR 0,0
                                                    MOVR 0,0
                                                    SKO 0,0

                                               ADDS 0,0
                                               ADDL 0,0
                                               ADDL 0,0,SNC

                                          ADDS·0,0
                                          ADDL 0,0
                                          SSN 0,0

                                     ADDS 0,0
                                     ADDL 0,0,SNC

                                MOVS 0,0
                                ADDL 0,0,SNC

                           MOVS 0,0
                           SSN 0,0

                      MOVS 0,0
                      SKO 0,0

                 MOVS 0,0
                 MOVR 0,0
                 SKO 0,0

            ADDL 0,0
            ADDL 0,0
            ADDL 0,0,SNC

       ADDL 0,0
       ADDL 0,0
       SSN 0,0

    ADDL 0,0
    ADDL 0,0,SNC

 ADDL 0,0
 SSN 0,0

 ADDL 0,0,SNC

SSN 0,0
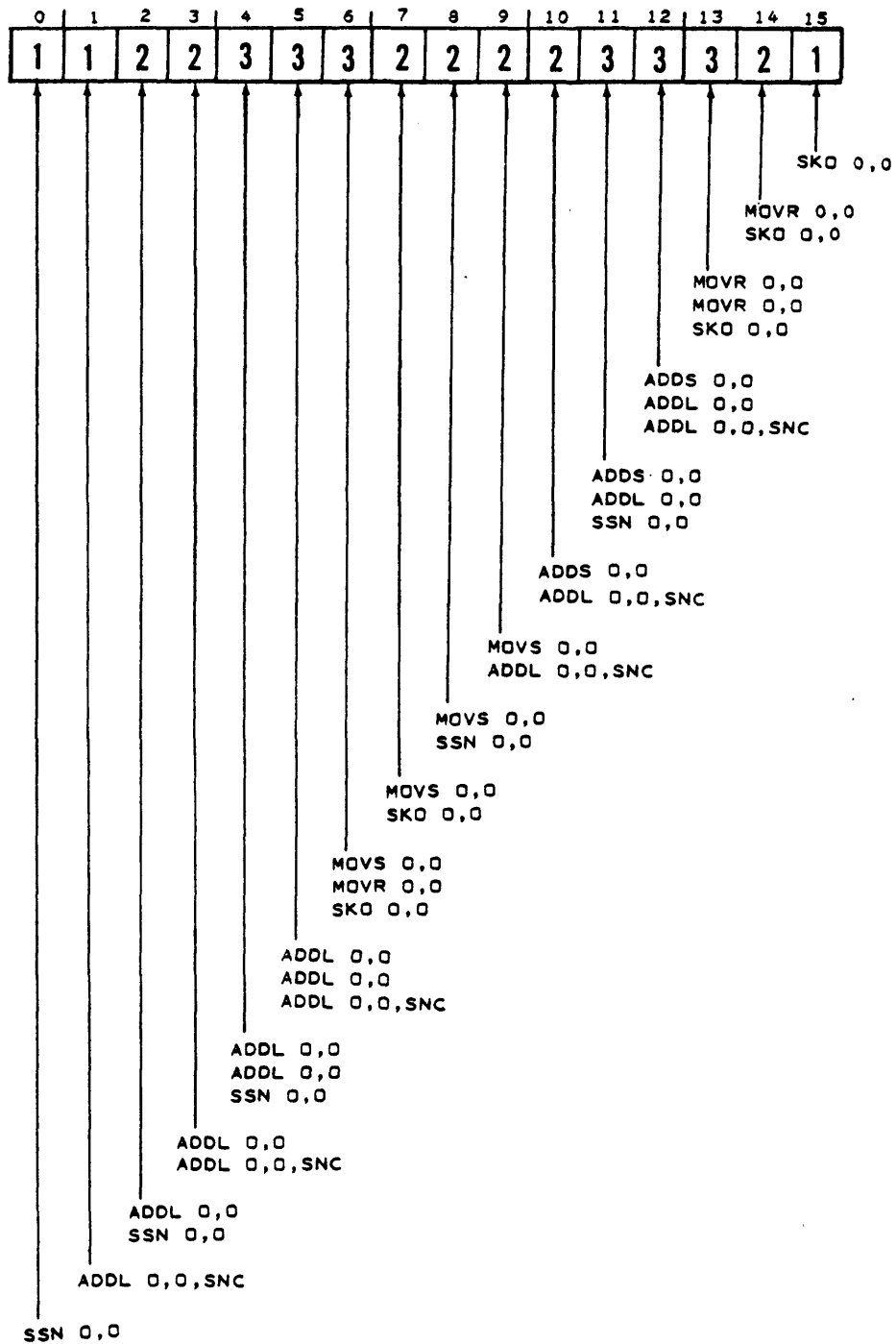
FIGURE E-3.   HOW TO TEST FOR ANY BIT IN A WORD

III. ACCUMULATOR HANDLING

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
```

To "Exclusive-OR" Two Accumulators

This routine takes advantage of 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·A1̄
COM     0,0
AND     0,1     ;A0̄·A1
ADD     2,1     ;A0⊕A1 = A0·A1̄ + A0̄·A1
```

To "Decrement" An Accumulator

Use the following routine to decrement an accumulator:

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

IV. 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.

V.   I/O PROGRAMMING FOR THE MASTER TERMINAL

The assembler listings on the following pages show examples of how
to do input and output to a Master Terminal (Teletype or CRT),
using the standard Device Code 10/11 type of controller. They
also illustrate byte handling and interrupt handling conventions.
Refer also to Section 3.

; PROGRAMMING EXAMPLES

            1000   .LOC   1000

; INPUT/OUTPUT ROUTINES

; MASTER TERMINAL INPUT - ACCEPTS CHARACTER INTO A0

```
    1000  63610       SKPDN   TTI      ;IS THERE ANY INPUT AVAILABLE ?
    1001    777       JMP     .-1      ;  NO, KEEP WAITING
    1002  60610       DIAC    0,TTI    ;YES, ACCEPT IT & CLEAR TTI DONE FLAG
```

;   NOTE: FOR PAPER TAPE READER USE DIAS TO START READING NEXT CHARACTER

; MASTER TERMINAL OUTPUT - ASSUMES OUTPUT CHARACTER IS IN A0

```
    1003  63511       SKPBZ   TTO      ;TTO STILL BUSY FROM A PREV. OUTPUT ?
    1004    777       JMP     .-1      ;  YES, WAIT FOR IT TO FINISH
    1005  61111       DOAS    0,TTO    ;  NO, OUTPUT THE CHAR. AND START TTO
```


; 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, A2 = PRESERVED

```
    1006   25400 TYPE: LDA    1,0,3    ;PICK UP 2 ASCII CHARACTERS
    1007  175420       INCZ   3,3      ;ADV. RETURN PNTR; C=BYTE CNTR
    1010   20411 TYPE2:LDA    0,C377L  ;LEFT-BYTE MASK
    1011  123705       ANDS   1,0,SNR  ;EXTRACT A BYTE - IS IT 0 ?
    1012    1400       JMP    0,3      ;  YES, RETURN TO CALLER
    1013   63511       SKPBZ  TTO      ;  NO, WAIT FOR TTO NOT BUSY
    1014    777        JMP    .-1
    1015   61111       DOAS   0,TTO    ;OUTPUT THE CHARACTER
    1016  125362       MOVCS  1,1,SZC  ;SWAP THE 2 ASCII CHAR.; CK. BYTE CNT
    1017    771        JMP    TYPE2    ;  TYPE THE SECOND CHARACTER
    1020    766        JMP    TYPE     ;  GET 2 MORE CHARACTERS TO TYPE

    1021  177400 C377L:177400         ;OCTAL 377 IN LEFT BYTE
```

; SUBROUTINE TO TYPE A NUMBER IN OCTAL FORM

; INITIAL CONDITIONS: Al = NUMBER TO BE TYPED
; CALLING SEQUENCE:
;   JSR   TPOCT
;   RETURNS HERE
; RETURN CONDITIONS: Al = 0, A2 = PRESERVED, C = 1

```
1022   20413  TPOCT:LDA    0,C.BIT
1023  101120  TPOC2:MOVZL  0,0          ;PRESET CARRY (MSB:1, OTHERS:0)
1024  125105        MOVL   1,1,SNR      ;LEFT-SHIFT A BIT OUT OF Al INTO C
1025    1400        JMP    0,3          ;RETURN WHEN PUSHER BIT IS GONE
1026  101103        MOVL   0,0,SNC      ;ASSEMBLE ASCII DIGIT; COMPLETE ?
1027     775        JMP    .-3          ;   NO, GET MORE BITS
1030   63511        SKPBZ  TTO          ;WAIT FOR TTO NOT BUSY
1031     777        JMP    .-1
1032   61111        DOAS   0,TTO        ;OUTPUT THE ASCII DIGIT
1033   20403        LDA    0,C.OCT
1034     767        JMP    TPOC2        ;CONTINUE THE LOOP

1035  140014  C.BIT:140014  ;CONST. TO STRIP OFF 1 BIT & CNVT. TO ASCII
1036   10003  C.OCT:010003  ;CONST. TO STRIP OFF 3 BITS & CNVT. TO ASCII
```

; 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

```
    1037 131220 GETBY:MOVZR  1,2      ;CONVERT BYTE ADDRESS INTO WORD ADDRESS
    1040  21000       LDA    0,0,2    ;FETCH WORD CONTAINING DESIRED BYTE
    1041 101003       MOV    0,0,SNC  ;DO WE WANT LEFT BYTE ?
    1042 101300       MOVS   0,0      ;  YES, SWAP THE WORD
    1043  30403       LDA    2,C377   ;RIGHT BYTE MASK
    1044 143400       AND    2,0      ;MASK THE RIGHT BYTE
    1045  1400        JMP    0,3      ;RETURN

    1046    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

```
    1047  54414 PUTBY:STA    3,PUTBR  ;SAVE RETURN ADDRESS
    1050 131220       MOVZR  1,2      ;FORM WORD ADDRESS FROM BYTE ADDR.
    1051  34775       LDA    3,C377   ;GET MASK FOR RIGHT HALF
    1052 163403       AND    3,0,SNC  ;MASK GIVEN BYTE; GOES IN LEFT HALF ?
    1053 101301       MOVS   0,0,SKP  ;  YES, SWAP THE BYTE
    1054 175300       MOVS   3,3      ;  NO, SWAP THE MASK
    1055  25000       LDA    1,0,2    ;FETCH THE WORD WHERE BYTE IS TO GO
    1056 167400       AND    3,1      ;MAKE ROOM FOR THE BYTE
    1057 107000       ADD    0,1      ;INSERT THE BYTE
    1060  45000       STA    1,0,2    ;PUT THE WORD BACK
    1061 145100       MOVL   2,1      ;RESTORE A1
    1062   2401       JMP    @PUTBR   ;RETURN

    1063      0 PUTBR:0                ;SAVE RETURN ADDRESS
```

- PAGE 4 -

; SIMPLE, SINGLE-LEVEL 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   40422  INTSV:STA     0,INTS0    ;\
    2001   44422        STA     1,INTS1    ; \
    2002   50422        STA     2,INTS2    ;  \ SAVE ACCUMULATORS
    2003   54422        STA     3,INTS3    ;  / AND CARRY
    2004  101100        MOVL    0,0        ; /
    2005   40421        STA     0,INTSC    ;/
    2006   30421        LDA     2,.INTV    ;POINTER TO INTERRUPT VECTOR TABLE
    2007   61477        INTA    0          ;GET CODE OF INTERRUPTING DEVICE
    2010  113000        ADD     0,2        ;COMPUTE DESIRED INTERRUPT VECTOR
    2011    7000        JSR     @0,2       ;JUMP TO INDICATED SERVICE ROUTINE

    2012   20414  INTSR:LDA     0,INTSC    ;RETURN FROM SERVICE ROUTINE
    2013  101200        MOVR    0,0        ; \
    2014   20406        LDA     0,INTS0    ;  \ RESTORE ACCUMULATORS
    2015   24406        LDA     1,INTS1    ;  / AND CARRY
    2016   30406        LDA     2,INTS2    ; /
    2017   34406        LDA     3,INTS3    ;/
    2020   60177        INTEN              ;RE-ENABLE INTERRUPTS
    2021    2000        JMP     @0         ;RETURN TO INTERRUPTED PROGRAM


    2022      0  INTS0:0                   ;SAVE A0
    2023      0  INTS1:0                   ;SAVE A1
    2024      0  INTS2:0                   ;SAVE A2
    2025      0  INTS3:0                   ;SAVE A3
    2026      0  INTSC:0                   ;SAVE CARRY
    2027   2100  .INTV:INTVT               ;POINTER TO INTERRUPT VECTOR TABLE


           2100          .LOC    2100      ;INTERRUPT VECTOR TABLE
U   2100   2100  INTVT:IS00               ;INTERRUPT SERVICE ADDRESS FOR DEVICE 00
U   2101   2101        IS01               ;FOR DEVICE 01
                       ;ETC.
```

; NOTE: MANY OF THE INTERRUPT SERVICE VECTORS MAY POINT TO THE SAME
;   DEFAULT SERVICE ROUTINE

INDEX

BTRY OK (processor LED) - 5.3.1; Fig. 5-2
BUSY flag - 2.6.1; 2.6.2

C (carry) flag - 2.5
C (change) MANIP command - 5.5.2; App. D
C (clear) control pulse:
  Regular I/O Instructions - 2.6.1
  Special I/O Instructions - 2.6.2; 3.6.1
C (complement) carry flag - 2.5.3.2
Carry flag equals one code - 2.5.3.2
Carry flip-flop - 5.5.2
Carry instruction codes - 2.5.3.2
CARRY processor LED - 5.3.2; Fig. 5-2
CARRY LED (OCU) - 5.4.2; Fig. 5-3
Carry Operations - 2.5.1.1; 2.5.1.2; Fig. 2-2; Fig. 2-3
Carry States (COUT, CNEW, CSEL, CRES) - 2.5.1.1; 2.5.1.2; Fig. 2-2
Change Command (MANIP) - 5.5.2; App. D
Chassis:
  Processor - 1.3.2; 1.4.1.2; 4.3
  Power Supply - 1.3.2; 1.4.1.5
  Mounting - 4.3; Fig. 4-3
Checksum - 5.5.2
Circuit board configuration - 4.3; Fig. 4-4
Circuit board mounting - 4.3
Clear control pulse - 2.6.1; 2.6.2; 3.6.1
CLEAR DATA control (OCU) - 5.4.4; Fig. 5-3
Clock rate - 1.3.1
CLR+ backplane signal - Table 3-1; 3.5.1; 3.5.2; 3.6.1
COM instruction - 2.5.2; Table 2-2; App. A; App. B
Complement carry code - 2.5.3.2
Complement instruction - 2.5.2; App. A; App. B
Compatible software - 1.2.5
Control instruction field - 2.6.1; 2.6.2
Control panel - see Front Panel; Mini-panel
CONT control (OCU) - 5.4.4; Fig. 5-3
CONT switch (processor) - 5.3.2; Fig. 5-2
Cooling - 1.4.1.2
Correction using MANIP - 5.5
Correction using OCU - 5.4.4
CPU/Memory board - 1.4.1.1; Fig. 1-4
CTRL instruction field - 2.6.1; 2.6.2
Customer support - 1.2.6
CY (carry) instruction field - 2.5

D (displacement) instruction field - 2.4.1
D (dump) MANIP command - 5.5.2; App. D
Data bus - 3.5.1

Voltage - 5.2; Fig. 5-1
Indirect addressing - 2.4.1.2
  32K mode - 2.4.1.2
  64K mode - 2.4.1.2
Input ASCII MANIP command - 5.5.2; App. D
Input/Output:
  Control logic - 1.4.1.1
  Control pulse - 2.6.1; 2.6.2; 3.6.1
  Device buffers - 2.6.1.1; 3.3
  Functions - 2.6.1; 2.6.2
  Transfer timing - 3.6
Input/output Bus interface signals - 3.5; 3.5.1; Table 3-1
Input/output connectors - 3.7; 4.3; Fig. 3-7
Input/output skip instructions - 2.6.1.2; App. A; App. B
Input/output transfer instruction - 2.6.1.1; App. A; App. B
Input to accumulator from buffer instructions - 2.6.1; App. A; App. B
Installation - 4.2; Fig. 4-1; Fig. 4-2
Instruction:
  Execution times - 2.7
  Overhead - 2.7
  Register - 1.4.1.1
Instruction formats:
  Arithmetic and Logical - 2.5
  Input/output - 2.6
  Memory Reference - 2.4
  Summary - Fig. 2-1
Instruction Types - 2.3
  Arithmetic and Logical - 2.5
  Input/Output - 2.6
  Memory Reference - 2.4
INST STEP control (OCU) - 5.4.4; Fig. 5-3
INTA+ I/O bus signal - 3.5.1; 3.5.2; Table 3-1
INTA a (DIB a, CPU) instruction - 2.6.2; App. B
INTDS (NIOC CPU) instruction - 2.6.2; App. B
INTEN (NIOS CPU) instruction - 2.6.2; App. B
Interrupt:
  Control signals - 3.5.1; Table 3-1
  Handling - 3.2.1; 3.6.2
  Mask flag - 2.6.2; 3.2.2
  Request flag - 3.2.1
  Sequence - 3.2.1
  Timing - 3.6.2; Fig. 3-4
Interrupt and Priority Scheme - 3.2; 3.2.1
INTPIN- I/O bus signal - 3.5.1; 3.5.2; Table 3-1
INTR- I/O bus signal - 3.5.1; 3.5.2; 3.6.3; Table 3-1
ION LED (OCU) - 5.4.2; Fig. 5-3
IOPLS+ I/O bus signal - 3.5.1; 3.5.2; Table 3-1

R (read paper tape) MANIP command - 5.5.2; App. D
R shift code - 2.5.3.1
RAM access time - 1.3.1
READS a (DIA a, CPU) instruction - 2.6.2; App. B
Regular I/O instruction - 2.6.1; App. A; App. B
Relative addressing - 2.4.1.1
RESET control (OCU) - 5.4.4; Fig. 5-3
Right rotate shift code - 2.5.3.1
RQENB - I/O bus signal - 3.5.1; 3.5.2; 3.6.2; 3.6.3; Table 3-1
RUN processor indicator - 5.3.2; Fig. 5-2
RUN indicator (OCU) - 5.4.2; Fig. 5-3

S (search for constant) MANIP command - 5.5.2; App. D
S (start) control pulse - 2.6.1; 2.6.2; 3.6.1
S shift code - 2.5.3.1
SBN skip test code - 2.5.3.4
Search for constant MANIP command - 5.5.2; App. D
Search for not-equal MANIP command - 5.5.2; App. D
Secondary Arithmetic and Logical instruction functions - 2.5.3
SELB- I/O bus signAl - 3.5.1; 3.5.2; Table 3-1
SELD- I/O bus signal - 3.5.1; 3.5.2; Table 3-1
Self-test operation - 4.5.2; 5.3.2; 5.4.4
   Continuous reliability - 4.5.2.3
   Halt interpretation - 4.5.2.2
   Normal - 4.5.2.1
Self-test switch - 4.5.2.1; Fig. 4-7
Set address offset MANIP command - 5.5.2; App. D
Set output delay MANIP command - 5.5.2; App. D
Set 64K addressing mode control (OCU) - 5.4.4; Fig. 5-3
SEZ skip test code - 2.5.3.4
Shift instruction field - 2.5
SH instruction field - 2.5
64K mode indicator (OCU) - 5.4.2; Fig. 5-3
64K word addressing mode - 2.4.1.2; 5.4.4
SK instruction field - 2.5; 2.5.1.1; 2.5.3.4
Skip instruction field - 2.5; 2.5.1; 2.5.3.4
Skip on condition instructions - 2.6.1.2
Skip test codes - 2.5.3.4
Skip testing - App. E-I; Fig. E-1; Fig. E-2
   Flags - 3.5.1; Table 3-1
SKP skip test code - 2.5.3.4
SKPBN instruction - 2.6.1.2; App. A; App. B
SKPBN CPU instruction - 2.6.2; App. A; App. B
SKPBZ instruction - 2.6.1.2; App. A; App. B
SKPBZ CPU instruction - 2.6.2; App. A; App. B
SKPDN instruction - 2.6.1.2; App. A; App. B
SKPDN CPU instruction - 2.6.2; App. A; App. B

SKPDZ instruction - 2.6.1.2; App. A; App. B
SKPDZ CPU instruction - 2.6.2; App. A; App. B
SNC skip test code - 2.5.3.4
SNR skip test code - 2.5.3.4
Special mnemonics (code-77 I/O instructions) - 2.6.2; App. B
Specifications - 1.3.2
STA instruction - 2.4.2.1; App. A; App. B
Standby mode - 4.4; 6.2
STANDBY processor switch position - 1.4.1.5; 4.4; 5.3.1; Fig. 5-2
START control (OCU) - 5.4.4; Fig. 5-3
START control pulse 0 - 2.6.1; 2.6.2; 3.6.1
STBY processor switch position - 5.3.1; Fig. 5-2
STOP processor switch - 5.3.2; Fig. 5-2
STOP control (OCU) - 5.4.4; Fig. 5-3
Store accumulator instruction - 2.4.2.1; App. A; App. B
STRT+ I/O bus signal - 3.5.1; 3.5.2; 3.6.1; Table 3-1
SUB instruction - 2.5.1; App. A; App. B
Subtract instruction - 2.5.1; App. A; App. B
Subtraction operation - 2.5.1.2; Fig. 2-2; Fig. 2-3
Swap shift code - 2.4.3.1
SZC skip test code - 2.5.3.4
SZR skip test code - 2.5.3

Teletype/Master CRT:
  Handling - 3.3.2
  Connector - 3.7.3
Temperature requirements - 1.3.2; 4.1
32K word addressing mode - 2.4.1.2
Timing control - 1.4.1.1; 3.6
Two's complement notation - 2.2
Typical configuration - 1.4; Fig. 1-2

Unconditional skip test code - 2.4.3.4
Unpacking instructions - 4.2; Fig. 4-1; Fig. 4-2

V (verify paper tape) MANIP command - 5.5.2; App. D
Value typeout MANIP command - 5.5.2; App. D
Verify paper tape MANIP command - 5.5.2; App. D
Virtual Control Panel - 1.4.1.3; 5.3.2; 5.4.4; 5.5
  Addressing modes - 5.5.1
  Commands - 5.5.2; App. D
  Operation - 5.5
Voltages
  Available for user applications - 1.4.1.5
  Battery backup voltages - 1.4.1.5; 5.2; Fig. 5-1
  System voltages - 1.4.1.5; 5.2; Fig. 5-1
  Voltage indicators - 5.2; Fig. 5-1

Word length - 1.3.1; 2.2

X instruction field - 2.4.1; 2.4.1.1
X (calculate checksum) MANIP instruction - 5.5.2; App. D

Y (set output delay) MANIP command - 5.5.2; App. D

Z carry code - 2.5.3.2
Zero carry code - 2.5.3.2

: (examine or deposit) MANIP command - 5.5.2; App. D