NOTE

This manual documents the Model 9000A-Z8000 and its assemblies at the revision levels identified in Section 7. If your instrument contains assemblies with different revision letters, it will be necessary for you to either update or backdate this manual. Refer to the supplemental change/errata sheet for newer assemblies, or to the backdating information in Section 7 for older assemblies.

9000A-Z8000

Interface Pod

Instruction Manual

P/N 716035
MAY 1984

FLUKE®

WARRANTY

Notwithstanding any provision of any agreement the following warranty is exclusive:

The JOHN FLUKE MFG. CO., INC., warrants each instrument it manufactures to be free from defects in material and workmanship under normal use and service for the period of 1-year from date of purchase. This warranty extends only to the original purchaser. This warranty shall not apply to fuses, disposable batteries (rechargeable type batteries are warranted for 90-days), or any product or parts which have been subject to misuse, neglect, accident, or abnormal conditions of operations.

In the event of failure of a product covered by this warranty, John Fluke Mfg. Co., Inc., will repair and calibrate an instrument returned to an authorized Service Facility within 1 year of the original purchase; provided the warrantor's examination discloses to its satisfaction that the product was defective. The warrantor may, at its option, replace the product in lieu of repair. With regard to any instrument returned within 1 year of the original purchase, said repairs or replacement will be made without charge. If the failure has been caused by misuse, neglect, accident, or abnormal conditions of operations, repairs will be billed at a nominal cost. In such case, an estimate will be submitted before work is started, if requested.

THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS, OR ADEQUACY FOR ANY PARTICULAR PURPOSE OR USE. JOHN FLUKE MFG. CO., INC., SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER IN CONTRACT, TORT, OR OTHERWISE.

If any failure occurs, the following steps should be taken:

1. Notify the JOHN FLUKE MFG. CO., INC., or nearest Service facility, giving full details of the difficulty, and include the model number, type number, and serial number. On receipt of this information, service data, or shipping instructions will be forwarded to you.

2. On receipt of the shipping instructions, forward the instrument, transportation prepaid. Repairs will be made at the Service Facility and the instrument returned, transportation prepaid.

SHIPPING TO MANUFACTURER FOR REPAIR OR ADJUSTMENT

All shipments of JOHN FLUKE MFG. CO., INC., instruments should be made via United Parcel Service or "Best Way"* prepaid. The instrument should be shipped in the original packing carton; or if it is not available, use any suitable container that is rigid and of adequate size. If a substitute container is used, the instrument should be wrapped in paper and surrounded with at least four inches of excelsior or similar shock-absorbing material.

CLAIM FOR DAMAGE IN SHIPMENT TO ORIGINAL PURCHASER

The instrument should be thoroughly inspected immediately upon original delivery to purchaser. All material in the container should be checked against the enclosed packing list. The manufacturer will not be responsible for shortages against the packing sheet unless notified immediately. If the instrument is damaged in any way, a claim should be filed with the carrier immediately. (To obtain a quotation to repair shipment damage, contact the nearest Fluke Technical Center.) Final claim and negotiations with the carrier must be completed by the customer.

The JOHN FLUKE MFG. CO., INC, will be happy to answer all applications or use questions, which will enhance your use of this instrument. Please address your requests or correspondence to: JOHN FLUKE MFG. CO., INC., P.O. BOX C9090, EVERETT, WASHINGTON 98206, ATTN: Sales Dept. For European Customers: Fluke (Holland) B.V., P.O. Box 5053, 5004 EB, Tilburg, The Netherlands.

*For European customers, Air Freight prepaid.

John Fluke Mfg. Co., Inc., P.O. Box C9090, Everett, Washington 98206

Table of Contents

continued on page ii

TABLE OF CONTENTS, *continued*

(continued on page iii)

TABLE OF CONTENTS, *continued*

List of Tables

List of Illustrations

Section 1
Introduction

NOTE

It is assumed that the user of this manual is familiar with the basic operation of one of the 9000 series Micro-System Troubleshooters as described in the 9000 series Operator Manuals.

1-1. PURPOSE OF THE INTERFACE POD

The 9000A-Z8000 Interface Pod (hereafter referred to as the Pod) interfaces any Fluke 9000 series Micro-System Troubleshooter (hereafter referred to as the Troubleshooter) to equipment using one of the Z8000 family of microprocessors. The Troubleshooter services printed circuit boards, instruments, and systems employing microprocessors. The 9000A-Z8000 Interface Pod adapts the general purpose architecture of the Troubleshooter to the specific architecture of the Z8000 microprocessor family. The Pod adapts such microprocessor-specific functions as pin layout, status/control functions, interrupt handling, timing, and memory and I/O addressing.

The 9000A-Z8000 Interface Pod can accommodate all four members of the Z8000 family: Z8001, Z8002, Z8003, and Z8004. An adapter is provided for use with the 40-pin Z8002 and Z8004 versions. A switch on the Pod selects between Segmented Memory Z8001 and Z8002 versions, and Virtual Memory Z8003 and Z8004 versions. Unless otherwise specified, references to the "Z8000" in this manual refer to any of the Z8000 family of microprocessors.

1-2. DESCRIPTION OF POD

Figure 1-1 shows the communication between the Troubleshooter, the Pod, and the Unit-Under-Test (hereafter referred to as the UUT). Cables connect the Pod to the Troubleshooter via a front-panel connector and to the UUT through the microprocessor socket.

The external features of the Pod is shown in Figure 1-2.

Internally, the Pod consists of a pair of printed circuit board assemblies mounted within an impact-resistant case. The Pod contains a Z8000 family microprocessor along with the supporting hardware and control software that is required to do the following:

1. Perform handshaking with the Troubleshooter.
2. Receive and execute commands from the Troubleshooter.
3. Report UUT status to the Troubleshooter.
4. Allow the Pod microprocessor to operate the UUT.

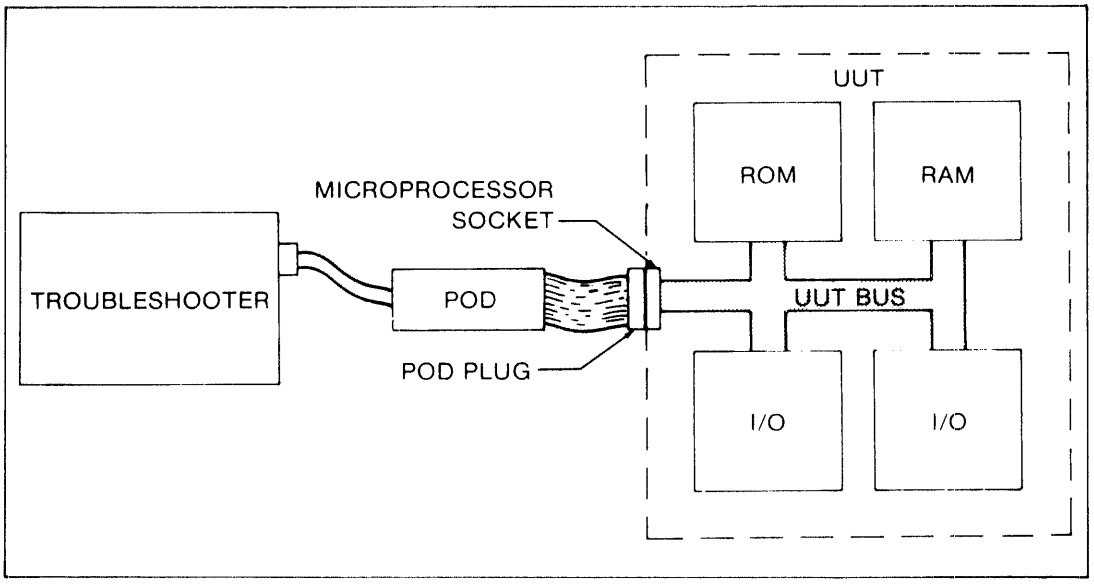


Figure 1-1. Communication Between the Troubleshooter, the Pod, and the UUT

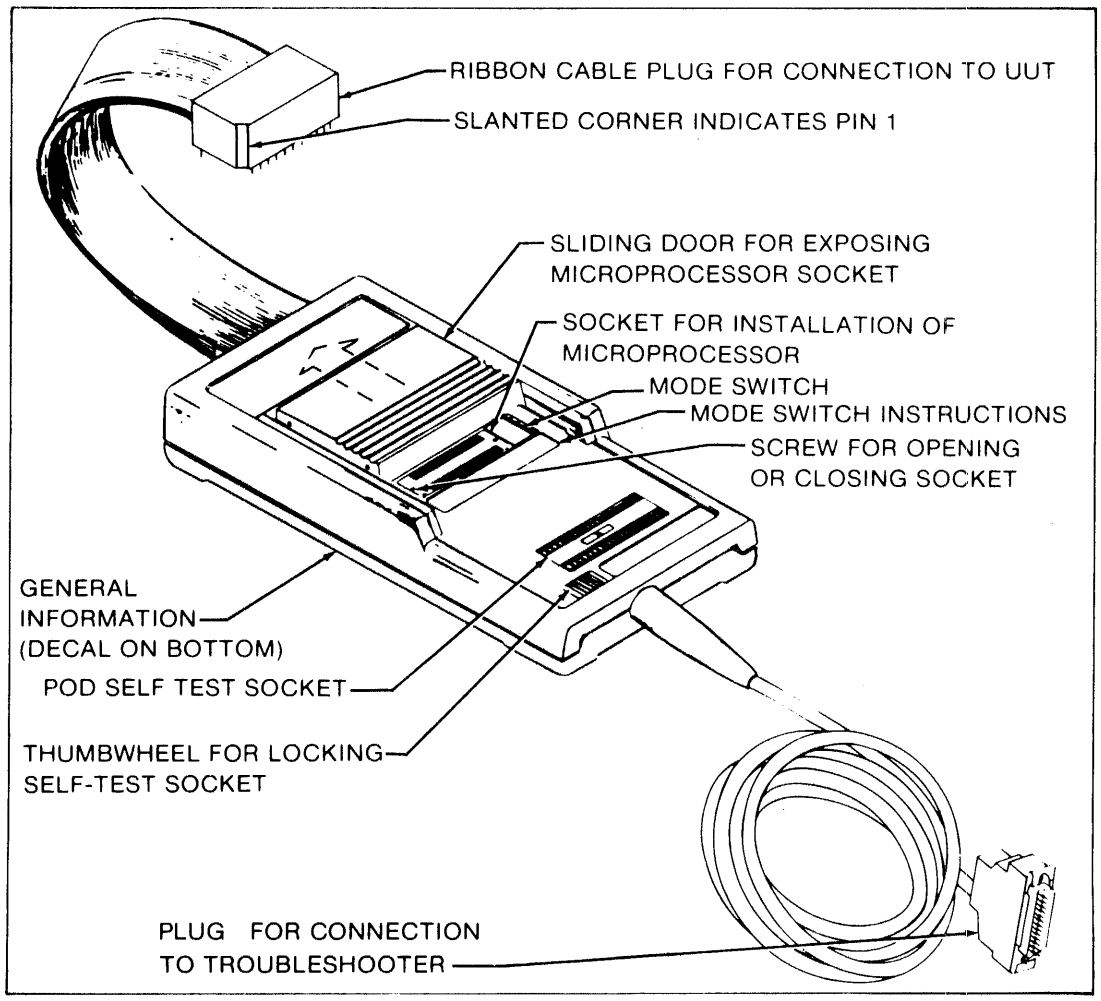


Figure 1-2. External Features of the Z8000 Interface Pod

The Troubleshooter supplies operating power (+5V) for the Pod. The UUT provides the external clock signal required by the Pod for operation. Using the UUT clock signal allows the Troubleshooter and Pod to function at the designed operating speed of the UUT (up to 10 MHz).

Logic level detection circuits are provided on each line to the UUT. These circuits allow detection of bus shorts, stuck-high, or stuck-low conditions, and any bus drive conflict (two or more drivers attempting to drive the same bus line).

Over-voltage protection circuits are also provided on each line to the UUT. These circuits guard against Pod damage which could result from the following:

1. Incorrectly inserting the ribbon cable plug in the UUT microprocessor socket.

2. UUT faults which place potentially-damaging voltages on the UUT microprocessor socket.

The over-voltage protection circuits guard against voltages of +12V to -7V on any one pin. Multiple faults, especially of long duration, may cause Pod damage.

A power level sensing circuit constantly monitors the voltage level of the UUT power supply (+5V). If UUT power rises above or drops below an acceptable level the Pod notifies the Troubleshooter of the power fail condition.

The 48-pin zero-insertion force Self Test Socket provided on the Pod enables the Troubleshooter to check Pod operation. The ribbon cable plug must be connected to the Self Test Socket during self test operation. The ribbon cable plug should also be inserted into this socket when the Pod is not in use to provide protection for the plug.

1-3. SPECIFICATIONS

Specifications for the Pod are listed in Table 1-1.

Table 1-1. Z8000 Pod Specifications

ELECTRICAL PERFORMANCE	
Power Dissipation	5 watts max.
Maximum External Voltage	-7V to +12V may be applied between ground and any ribbon cable plug pin continuously.
MICROPROCESSOR SIGNALS	
Input Low Voltage	-03V min., 0.8V max.
Input High Voltage	2.0V min., 5.0V max.
Output Low Voltage	0.5V max. at rated current
Output High Voltage	2.4V min. at -400 ua
Tristate Output Leakage Current	±0.02 mA typical, +0.1 to -0.2 mA max.
Input Current	
CLK	-1.2 mA max.
All Other Input Lines	-.6 mA max.

Table 1-1. Z8000 Pod Specifications (cont)

TIMING CHARACTERISTICS	
Maximum External Clock Frequency	10.0 MHz typical
Insertion Delays to Z8000 Signals	
Input Signals	12 ns typical
Output Signals	15 ns typical
UUT POWER DETECTION	
Detection of Low Vcc Fault	Vcc <+4.5V
Detection of High Vcc Fault	Vcc >+5.5V
Pod Protection from UUT Low Power	Vcc <+3.3V
GENERAL	
Size	5.7 cm H x 14.5 cm W x 27.1 cm L (2.2 in H x 5.7 in W x 10.7 in L)
Weight	1.5 kg (3.3 lbs)
Environment	
STORAGE	-40°C to +70°C, RH <95% non-condensing
OPERATING	0°C to +40°C, RH <95% non-condensing +40°C to +50°C, RH <75% non-condensing

Section 2
Installation and Self Test

2-1. INTRODUCTION

The procedures for connecting the Pod to the Troubleshooter, performing the Pod Self Test, and connecting Pod to the UUT are given in the following paragraphs.

2-2. INSTALLING THE MICROPROCESSOR IN THE POD

A microprocessor must be installed in the Pod to prepare it for testing a UUT.

Note

The Pod is supplied with a Z8001 microprocessor rated for 6 MHz operation. You will need to replace the microprocessor if your application requires a faster Z8001 or if your UUT uses a Z8002, Z8003, or Z8004.

The Pod socket is not designed for repeated insertions. It is not meant to test a new CPU with each tested assembly.

To install a microprocessor in the Pod, perform the following steps:

1. If the Pod is already connected, remove power from the UUT and the Troubleshooter.

2. Select a microprocessor to use in the Pod, either the one out of the UUT, or another of the same type.

3. Open the sliding door on the top of the Pod (shown in Figure 2-1) to expose the Pod microprocessor socket. Open the socket contacts by using a screwdriver to turn the screw at the end of the socket. Turn the screw counterclockwise to open the socket.

4. If a Z8002 or Z8004 is used, insert the microprocessor into a 40-48 pin adapter before installing it into the Pod. Insert the chosen microprocessor into the socket, aligning pin 1 to the marked position. Close the socket contacts by turning the screw clockwise. Close the sliding door.

5. Set the Processor Select Switch to the correct position (as shown on the Pod decal) for Segmented Memory or Virtual Memory devices.

2-3. CONNECTING THE POD TO THE TROUBLESHOOTER

1. Remove power from the Troubleshooter.

2. Using the round shielded cable, connect the Pod to the Troubleshooter at the location shown in Figure 2-1. Secure the connector using the sliding collar.

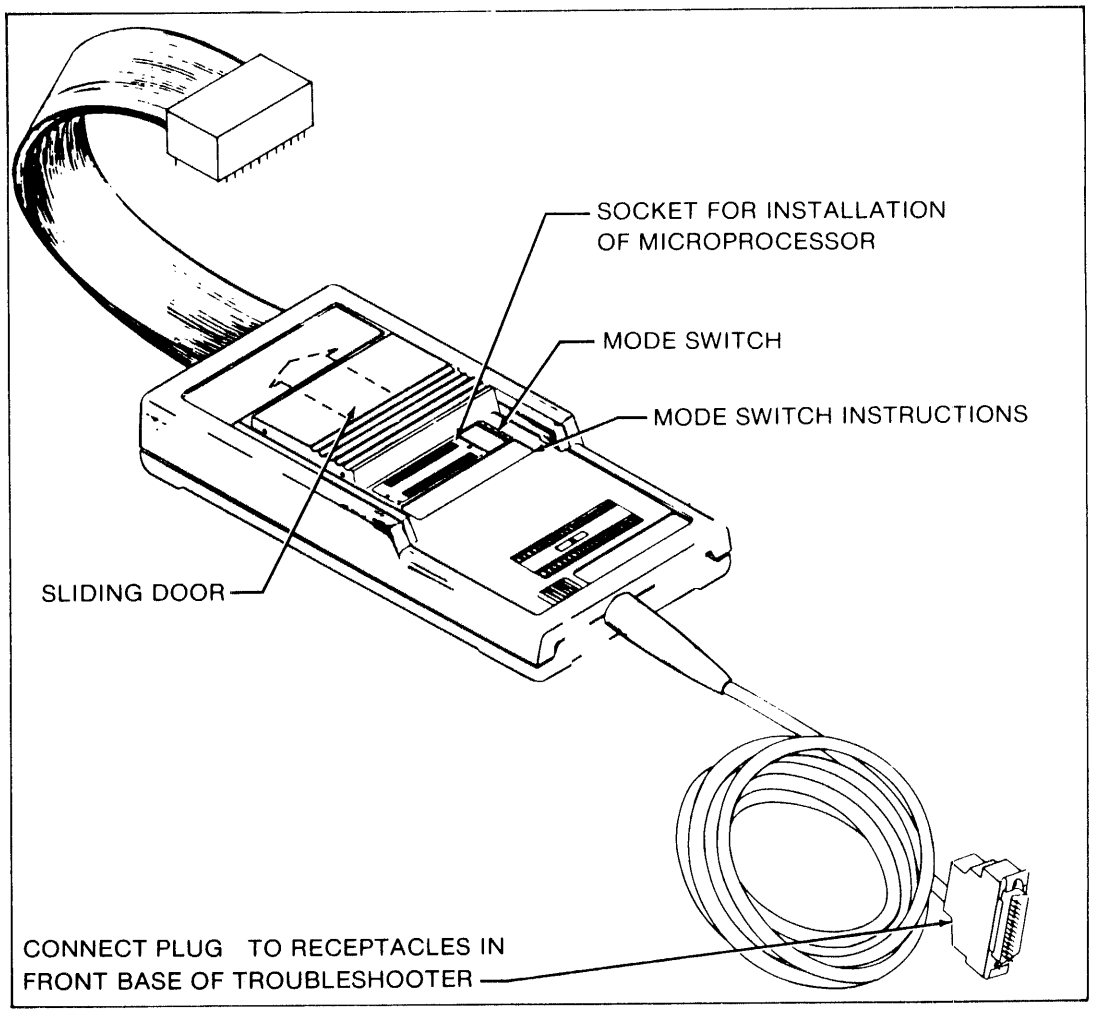


Figure 2-1. Location of Microprocessor Socket and Mode Switches

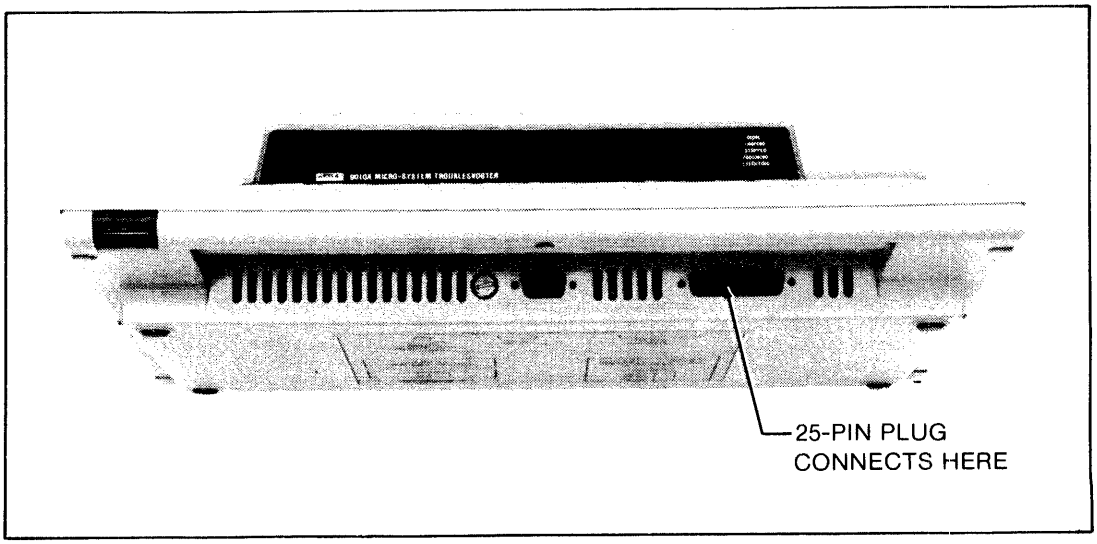


Figure 2-2. Connection at Interface Pod to Troubleshooter

2-4. PERFORMING THE POD SELF TEST

To perform the Pod Self Test, perform the following steps:

1. Make sure that a 6 MHz or greater clock speed microprocessor is installed in the Pod microprocessor socket, and that the Processor Select Switch is set according to the Pod decal.

2. If a 40-pin adapter is attached to the ribbon cable plug, remove the adapter before proceeding.

3. Open the pins of the Pod Self Test socket by turning the adjacent thumbwheel. Insert the ribbon cable plug into the socket. Close the socket using the thumbwheel.

4. Turn the power on and press the BUS TEST key on the Troubleshooter to initiate the Pod Self Test.

If the Troubleshooter displays the message *POD SELF TEST Z8000 OK,* the Pod is operating properly.

If the Troubleshooter displays any message other than *POD SELF TEST Z8000 OK,* the Pod may not be operating properly. Make sure the Pod ribbon cable plug is properly positioned in the self test socket and try the Self Test again.

For information about Pod troubleshooting and repair, refer to Section 6.

2-5. CONNECTING THE POD TO THE UUT

WARNING

TO PREVENT POSSIBLE HAZARDS TO THE OPERATOR OR DAMAGE TO THE UUT, DISCONNECT ALL HIGH-VOLTAGE POWER SUPPLIES, THERMAL ELEMENTS, MOTORS, OR MECHANICAL ACTUATORS WHICH ARE CONTROLLED OR PROGRAMMED BY THE UUT MICROPROCESSOR BEFORE CONNECTING THE POD.

Connect the Pod to the UUT as follows:

1. Be sure that power is removed from the UUT.

2. Disconnect UUT analog outputs or potentially hazardous UUT peripheral devices as described in the warning at the beginning of this section.

3. If necessary, disassemble the UUT to gain access to the UUT microprocessor socket. If the UUT microprocessor is still in the socket, remove the microprocessor.

4. Turn the Pod self test socket thumbwheel to release the Pod plug, and remove the Pod plug from the self test socket.

5. Insert the Pod plug into the UUT microprocessor socket, using the proper adapter if the UUT uses a 40-pin microprocessor. Make sure the slanted corner of the Pod plug is aligned with pin 1 of the UUT microprocessor socket.

6. Reassemble the UUT using extender boards if necessary.

CAUTION

The Pod contains active protection circuits. To avoid damage to the Pod, turn the Troubleshooter power on before applying power to the UUT.

7. Apply power to the UUT.

Section 3
Microprocessor Data

3-1. INTRODUCTION

This section contains microprocessor data which may be useful during operation of the Troubleshooter. This information includes descriptions of Z8000 signals and pin assignments.

3-2. MICROPROCESSOR SIGNALS

Table 3-1 lists all of the Z8000 microprocessor signals and provides a brief description of each signal. Refer to the microprocessor manufacturer's literature for complete information.

Table 3-2 is a summary of the Z8000 microprocessor signal activity.

Figures 3-1 through 3-4 show the Z8000 family pin assignments.

Table 3-1. Signal Descriptions

SIGNAL NAME	DESCRIPTION
$\overline{\text{ABORT}}$	The Abort line is used in conjunction with the $\overline{\text{SAT}}$ line to interrupt instructions before they are completed. (Available on the Z8003 and Z8004 only.)
AD0-AD15	These 16 tri-state multiplexed Address/Data lines are used to address memory and for Input/Output. The lines contain address information when the Address Strobe ($\overline{\text{AS}}$) line rises and data when the Data Strobe ($\overline{\text{DS}}$) line rises.
$\overline{\text{AS}}$	The rising edge of this Address Strobe line indicates valid addresses.
$\overline{\text{BUSACK}}$	When this Bus Acknowledge line is Low, the CPU has relinquished control of the bus.
$\overline{\text{BUSREQ}}$	The Bus Request line is driven Low to request the bus from the CPU.
$\overline{\text{DS}}$	The rising edge of the Data Strobe line indicates valid data available on the multiplexed Address/Data ($\overline{\text{AD}}$) lines.
$\overline{\text{MREQ}}$	Memory Request is a tri-state output that indicates that a memory address is present on the address/data bus.

Table 3-1. Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
$\overline{MI}$, $\overline{MO}$	Multi-Micro In and Multi-Micro Out form part of a daisy-chain that allows sharing resources in a multi-microprocessor system.
$\overline{NMI}$	A falling edge on the Non-Maskable Interrupt line requests a non-maskable interrupt. $\overline{NMI}$ has priority over the Vectored and Non-Vectored Interrupts.
$\overline{NVI}$	The Non-Vectored Interrupt line initiates a non-vectored interrupt.
CLK	The System Clock is a single-phase, five-volt time base.
$\overline{RESET}$	The Reset line resets the CPU.
R/$\overline{W}$	Read/Write indicates that the CPU is performing a read or write operation with memory or I/O.
$\overline{SAT}$	The Segment Page Address Translation Trap line is activated by a Memory Management Unit (MMU) to interrupt the CPU while a program or data in secondary storage is moved into main memory. (Available on the Z8003 only.)
$\overline{SEGT}$	The Segment Trap is asserted by the Memory Management Unit (MMU) to interrupt the CPU when the MMU encounters a segment trap. (Available on the Z8001 only.)
SN0-SN6	The Segment Number lines provide a segment number for use by a Memory Management Unit. (Available on the Z8001 and Z8003 only.)
ST0-ST3	These Status lines indicate the CPU status (refer to Table 3-3, Z8000 CPU Status Codes).
$\overline{STOP}$	The Stop line is usually used to single-step instructions.
$\overline{VI}$	This line requests a Vectored-Interrupt.
$\overline{WAIT}$	The Wait line tells the CPU that an I/O device or the memory is not ready for a transfer of data.
B/$\overline{W}$	Byte/Word specifies the nature of the 16-bit information on the address/data bus.
N/$\overline{S}$	Normal/System Mode indicates the CPU's present operating mode.

Table 3-2. Signal Summary

SIGNAL NAME	MNEMONIC	INPUT/ OUTPUT	ACTIVE STATE	DRIVER
Abort	$\overline{\text{ABORT}}$	input	low	tri
Address/Data	AD0-AD15	output	high	tri
Address Strobe	$\overline{\text{AS}}$	output	low	tri
Bus Acknowledge	$\overline{\text{BUSACK}}$	output	low	
Bus Request	$\overline{\text{BUSREQ}}$	input	low	tri
Memory Request	$\overline{\text{MREQ}}$	output	low	
Multi-Micro In	$\overline{\text{MI}}$	input	low	
Multi-Micro Out	$\overline{\text{MO}}$	output	low	
Non-Maskable Interrupt	$\overline{\text{NMI}}$	input	low	
Non-Vectored Interrupt	$\overline{\text{NVI}}$	input	low	
System Clock	CLK	input	—	
Reset	$\overline{\text{RESET}}$	input	low	tri
Read/Write	R/$\overline{\text{W}}$	output	high/low	
Segment Page Address Translation Trap	$\overline{\text{SAT}}$	input	low	
Segment Trap	$\overline{\text{SEGT}}$	input	low	tri
Segment Number	SN0-SN6	output	high	tri
Status	ST0-ST3	output	high	
Stop	$\overline{\text{STOP}}$	input	low	
Vectored Interrupt	$\overline{\text{VI}}$	input	low	
Wait	$\overline{\text{WAIT}}$	input	low	tri
Byte/Word	B/$\overline{\text{W}}$	output	high/low	tri
Normal/System Mode	N/$\overline{\text{S}}$	output	high/low	

NOTE: Driver Definitions: tri - Tri-state Driver
TTL = TTL Compatible (input or output)

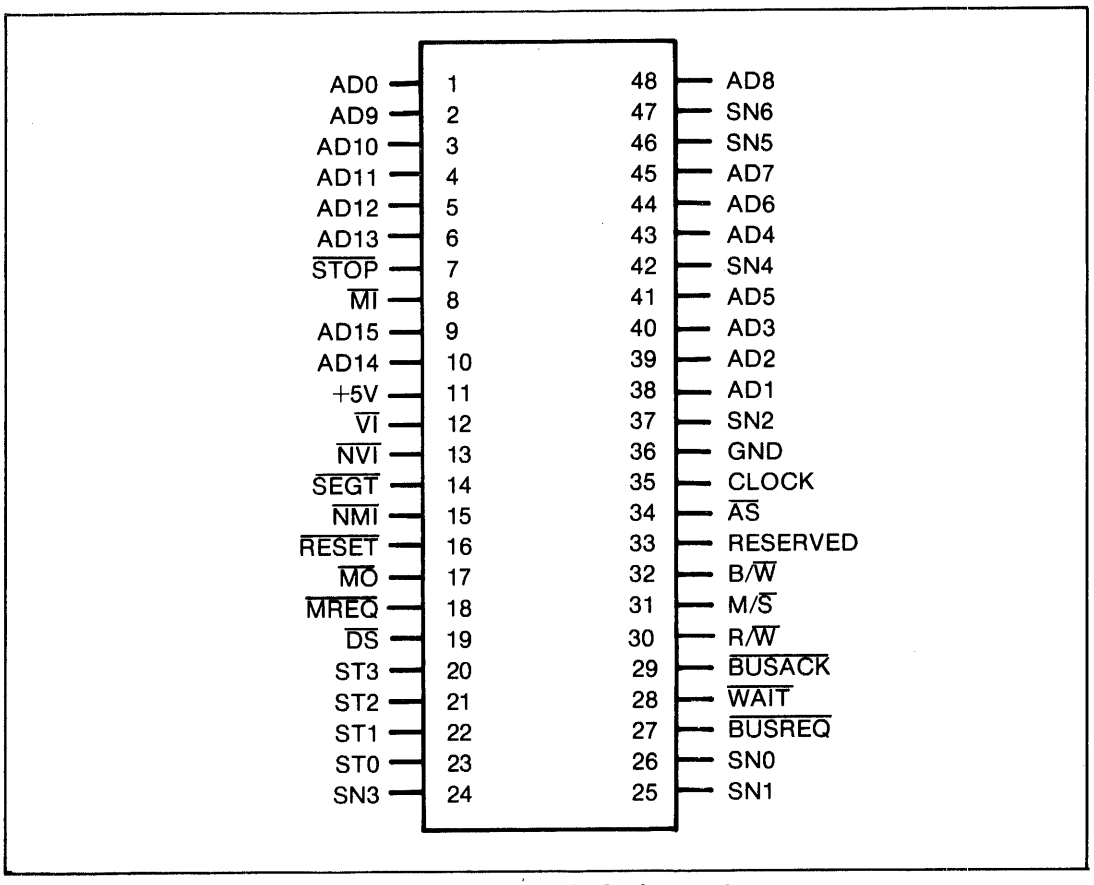


Figure 3-1. Z8001 Pin Assignments

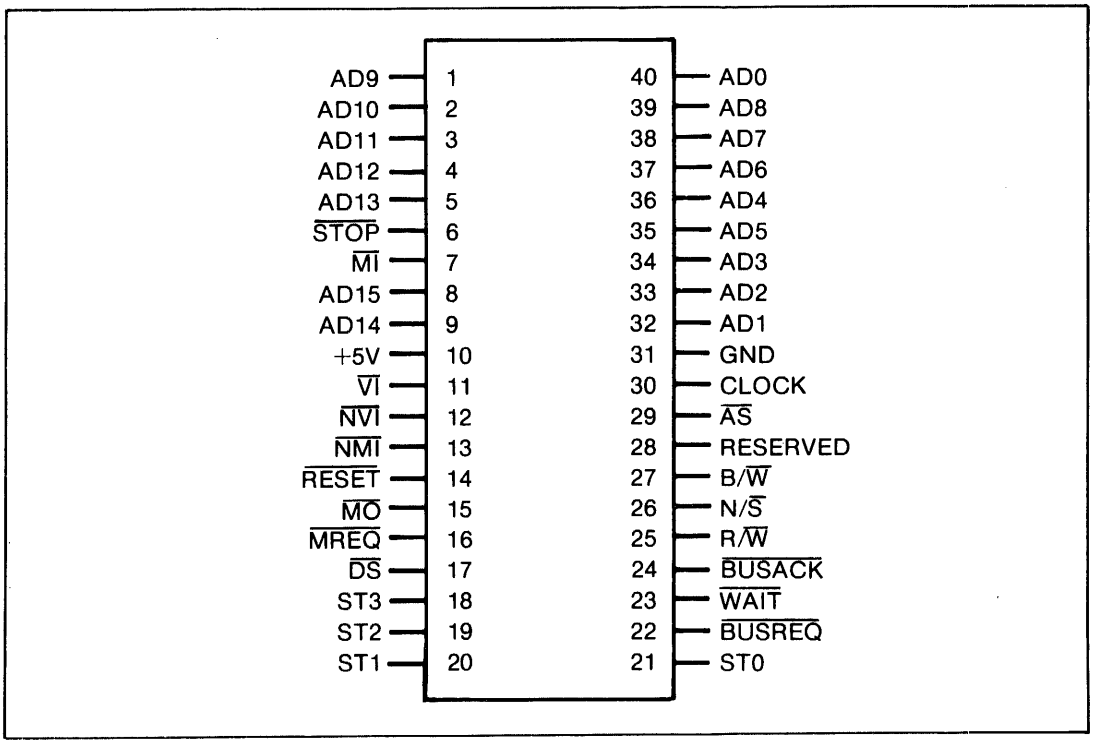


Figure 3-2. Z8002 Pin Assignments

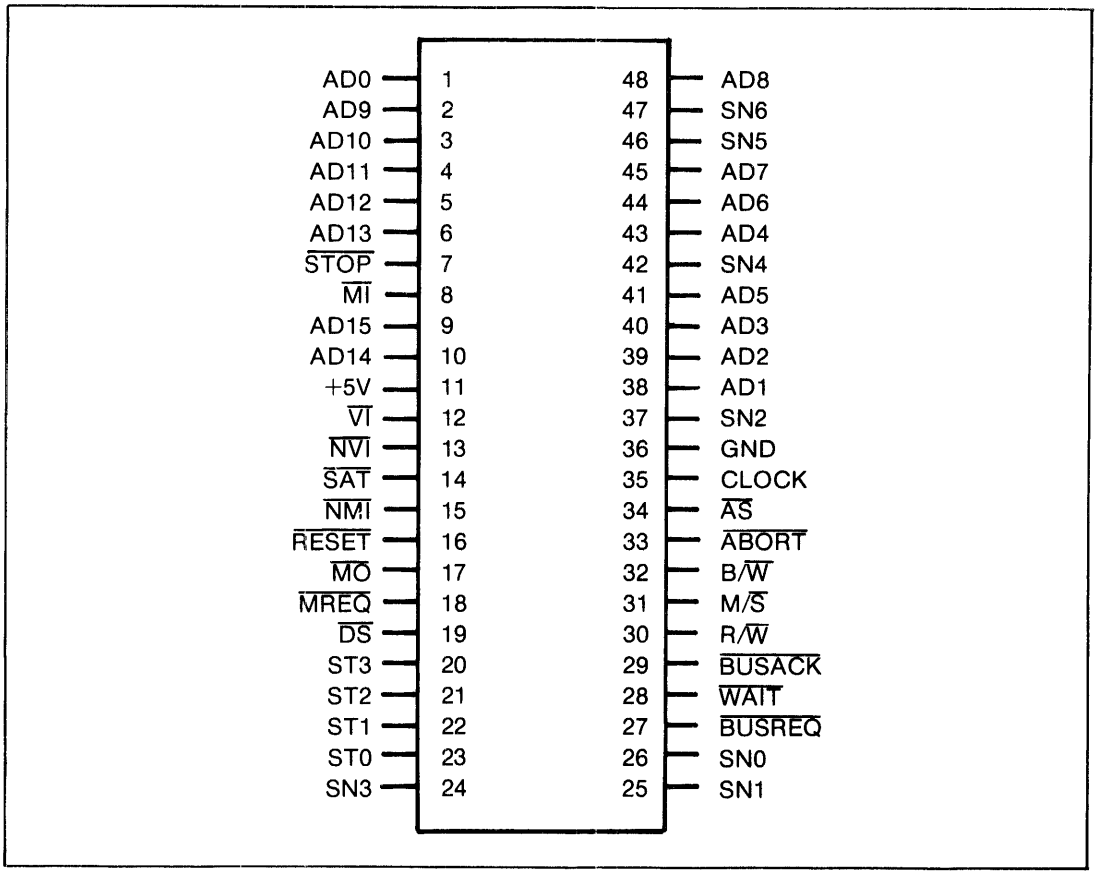


Figure 3-3. Z8003 Pin Assignments

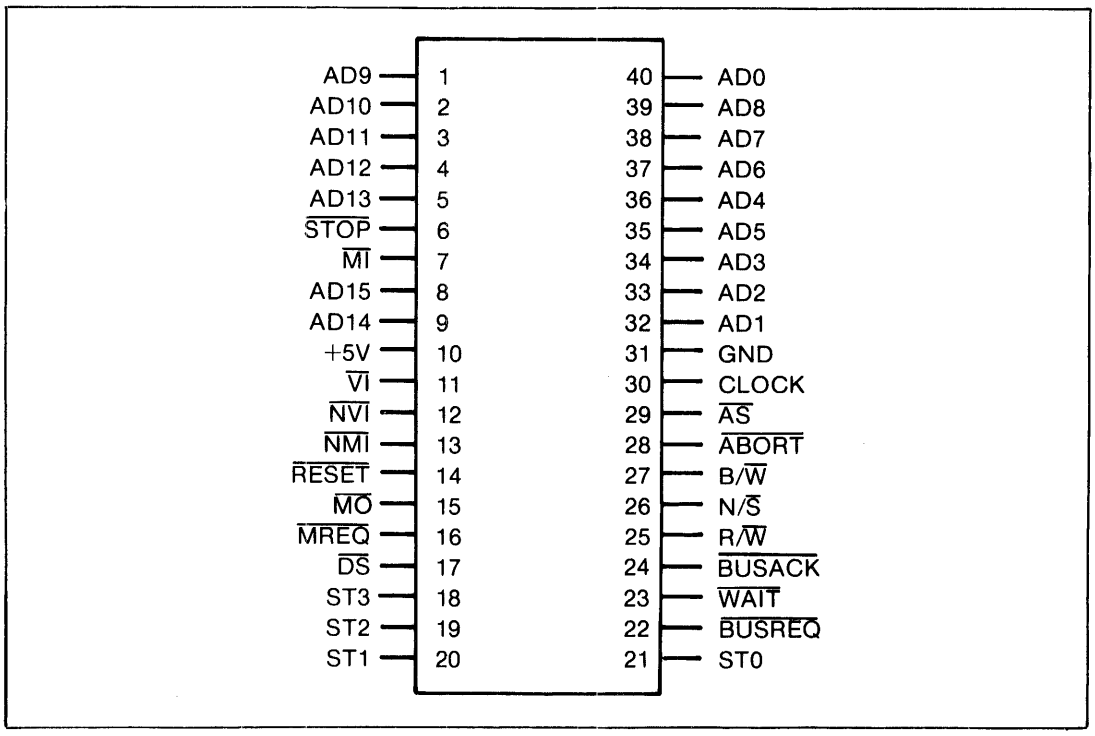


Figure 3-4. Z8004 Pin Assignments

Section 4
Operating Information

4-1. INTRODUCTION

This section contains information which pertains to operating the Troubleshooter with Z8000-based systems. This additional information complements the information in the Troubleshooter Operator and Programming manuals, and covers such items as the following:

- Address space assignment
- Special address functions
- Characteristics of Z8000 memory addressing
- Definitions and bit assignment of status lines
- Definitions of forcing and interrupt lines
- Definitions and characteristics of user-writable control lines
- Bit assignments of control lines
- Interrupt handling
- Characteristics of Bus Test, Learn, and Run UUT
- Marginal UUT problems

4-2. GETTING STARTED

After the Pod is connected to the Troubleshooter and installed in the UUT, you may see the message *POD TIMEOUT - ATTEMPTING RESET* displayed by the Troubleshooter as soon as any Pod operation is attempted. This message usually appears because the UUT is asserting a forcing status line: either the $\overline{\text{BUSREQ}}$ (Bus Request), or $\overline{\text{WAIT}}$ lines. Manually resetting the UUT may remove the problem, but it may be necessary to disable the status input using the Troubleshooter Setup function. Setting the corresponding Setup messages *SET - ENABLE xxxx?* to NO disables the offending line.

If the status line remains faulty and you attempt another operation, the message *ACTIVE FORCE LINE - LOOP?* appears. Pressing the MORE key allows you to see which line is causing the message to appear. You can disable reporting of this error and continue operation by setting the Setup message *SET - TRAP ACTIVE FORCE LINE?* to NO. For more information about enable lines, refer to a later section titled User Enablable Status Lines. For more information about forcing lines, refer to a later section titled Forcing Lines.

NOTE

Operating the Pod with the status lines disabled will cause UUT errors if the Z8000 microprocessor is required to WAIT or allow DMA accesses while under test.

If the message *POD TIMEOUT - ATTEMPTING RESET* remains after you disable the enablable lines, the problem may be that the UUT is not supplying a clock to the Pod. If the clock is working properly, perform a Pod self test as described in Section 2.

If the Troubleshooter displays an *ACTIVE FORCE LINE* message during the performance of BUS TEST on a properly functioning UUT, it may be necessary to change the Bus Test address using the Setup function of the Troubleshooter, or it may be necessary to inhibit reporting of forcing line errors by using the Setup function of the Troubleshooter, or by using the forcing line error mask special address. Refer to the Forcing Line Error Mask description under the Special Features of the Z8000 Pod in this section.

4-3. ADDRESS SPACE ASSIGNMENT

4-4. Introduction

All of the Z8000 family of microprocessors have 16 multiplexed address lines (AD0 - AD15) which allow direct addressing of 64K bytes of memory. In addition, the Z8001 and Z8003 versions have seven segment lines (SN0-SN6) which select one of 128 64K address segments, allowing a total addressing range of 8M bytes. The Z8000 can use seven data types, from 32-bit long words to individual bits.

4-5. Address Mapping

In order to allow the user to easily enter the complex address descriptions for the Z8000, a simplified address descriptor is used for specifying addresses via the Troubleshooter.

Addresses for the segmented versions of the Z8000 microprocessor (Z8001 and Z8003) are normally defined by an Offset (an address within a 64K byte block) and a Segment (one of 128 possible memory blocks).

64K Byte Offset Addresses

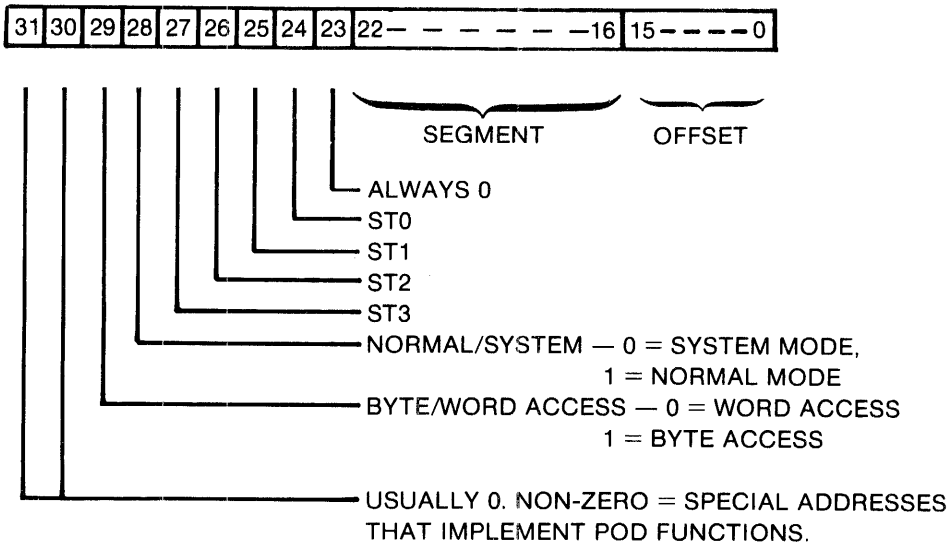
Address offsets (and addresses for non-segmented versions) are specified by bits 0-15 of the address. This provides an address offset range of 0000- FFFF. The address offset is put on lines AD0-AD15 during a bus access by the Pod.

Address Segment Notation

The Z8001 and Z8003 have additional output lines that can be used to switch between different segments of memory.

To simplify segment address components using the Troubleshooter, the seven segment bits appear in bits 16-22 of the address.

As an added convenience for the operator, the high byte of the address designates Normal/$\overline{\text{System}}$ mode and Byte/$\overline{\text{Word}}$ operation. There are four status lines, ST0-ST3, that indicate Pod operation in the stack, program, or data space.



For example, the address

1962 A77E

shows an address offset of A77E in memory segment 62. The CPU's program counter registers will receive the value 6200 A77E. The upper byte indicates system mode operation and using word accesses in the stack space. These status elements are described below.

Word and Byte Accesses

The Z8000 family of microprocessors provides for both word and byte accesses on the microprocessor bus. The Troubleshooter makes specifying word or byte accesses convenient for the operator by using a single bit of the address. Bit 29 of the address will be sensed by the Pod and the B/$\overline{W}$ (BYTE/$\overline{WORD}$) line to the UUT will be set accordingly. If bit 29 is zero, then the B/$\overline{W}$ signal will be low during the bus cycle, resulting in a word access. For byte accesses, bit 29 is set to a one.

The Z8000 Pod accepts only even addresses for word accesses. If odd addresses are specified for word access, the Troubleshooter defaults to the next lower (even) address and displays an error message.

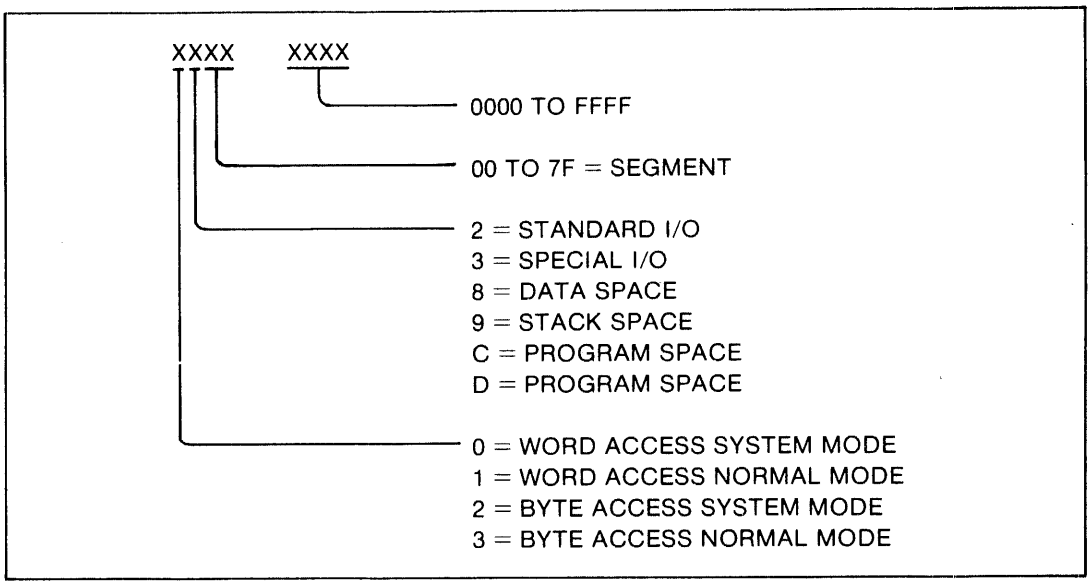
Normal/System Mode

The Z8000 microprocessors provide a control signal (N/$\overline{S}$) to the UUT that indicates the CPU's operating mode. This signal can be controlled through setting or clearing bit 28 in the address. If bit 28 is a one, then the Pod sets the N/$\overline{S}$ line high, indicating the normal mode to the UUT. The system mode is indicated by setting bit 28 to zero.

Status Line

Bits 24-27 are four status lines that denote current UUT system status. Not all combinations of the four status lines, the Normal/$\overline{System}$ status line, and the Byte/$\overline{Word}$ status line will occur. All possible status code combinations are shown in Table 4-1, Status Codes.

Table 4-1. Status Codes



Special Addresses

In addition to the regular address spaces, the Pod recognizes special addresses that are used to access information in the Pod or to cause the Pod to perform special functions. The special addresses are of the form F000 00XX. These special functions include indirect vectoring of Run UUT, Quick looping, Quick RAM and ROM tests, interrupt handling, and other miscellaneous controls. These functions are discussed in the section titled Special Functions of the Z8000 Pod.

For example, the address

F000 0016

is a special Pod address containing the state of the $\overline{MO}$ output line.

4-6. STATUS/CONTROL LINES

4-7. Introduction

The Troubleshooter classifies the signals at the microprocessor pins into four categories; address, data, status, and control. Address and data are multiplexed on the same lines. The A/D (Address/Data) lines contain address information on the rising edge of the Address Strobe line, and valid data on the rising edge of the Data Strobe line. Status lines are inputs to the microprocessor. They provide the CPU with critical status information about the system. Control lines are outputs from the microprocessor. They are the means with which the microprocessor can control other devices in the system using bus transactions.

The Pod permits the operator to monitor the state of the status lines and to manipulate the control lines from the Troubleshooter mainframe. The following paragraphs describe these capabilities.

4-8. Status Line Bit Assignments

When a Read Status operation is performed, the Troubleshooter displays the logic levels of each of the status lines in binary form, where "1" indicates a logic high, and a "0" indicates a logic low. To determine which digits correspond to specific status lines, refer to Table 4-2 or the Pod decal (on the bottom of the Pod).

For example: If a Read Status operation is performed and there are no active status lines and no flags set, the Troubleshooter displays:

READ @ STS = 0000 0111 1110 1111 OK

If a WAIT is pending, a Read Status operation will display:

READ @ STS = 0000 0111 1110 1011 OK

Table 4-2. Status and Control Line Bit Assignments

BIT	STATUS LINES	BIT	CONTROL LINES
15	$\overline{\text{SAT}}$ ACKNOWLEDGE FLAG	15	—
14	$\overline{\text{VI}}$ ACKNOWLEDGE FLAG	14	—
13	$\overline{\text{NVI}}$ ACKNOWLEDGE FLAG	13	—
12	$\overline{\text{NMI}}$ ACKNOWLEDGE FLAG	12	—
11	TIMEOUT FLAG	11	$\overline{\text{AS}}$
10	$\overline{\text{MI}}$	10	$\overline{\text{MO}}$
9	* $\overline{\text{RESET}}$	9	N/$\overline{\text{S}}$
8	** $\overline{\text{VI}}$	8	$\overline{\text{MREQ}}$
7	** $\overline{\text{NVI}}$	7	R/$\overline{\text{W}}$
6	** $\overline{\text{NMI}}$	6	B/$\overline{\text{W}}$
5	** $\overline{\text{SAT}}$	5	$\overline{\text{DS}}$
4	POWER FAIL FLAG	4	ST3
3	* $\overline{\text{ABORT}}$ (Z8003, Z8004 ONLY)	3	ST2
2	* *** $\overline{\text{WAIT}}$	2	ST1
1	* $\overline{\text{STOP}}$	1	ST0
0	* *** $\overline{\text{BUSREQ}}$	0	* $\overline{\text{BUSACK}}$

* FORCING LINES
** INTERRUPT LINES
*** USER ENABLEABLE

* USER WRITEABLE

The active low level at bit 2 indicates that the $\overline{\text{WAIT}}$ line is active. Note that most status lines are active low; the exceptions are flags at bits 4 and 11-15.

NOTE

When displaying status line error information (or other error information), the Troubleshooter displays the faulty lines as ones and good lines as zeroes rather than showing logic levels.

NOTE

The flag bits 4 and 11-15 do not represent particular Z8000 signals, but are generated within the Pod to indicate significant events to the Troubleshooter operator. (Refer to the section titled Status Lines Generated by the Pod.)

4-9. User-Enablable Status Lines

The Z8000 has two inputs ($\overline{\text{BUSREQ}}$ and $\overline{\text{WAIT}}$) which the operator can individually enable or disable using the Troubleshooter's Setup function. When these inputs are disabled, the UUT-generated signals appearing at these inputs are prevented from affecting the Pod.

For example: A $\overline{\text{WAIT}}$ line stuck at the active low level would cause the Z8000 within the Pod to stop and wait for a device to accept a data transfer, preventing normal Pod/Troubleshooter operation. After disabling this input to the Z8000 using the Setup function of the Troubleshooter, the $\overline{\text{WAIT}}$ signal is prevented from holding up normal Pod operation. Also see the discussion of the Timeout Flag, Paragraph 4-12.

Either of these status lines may be enabled or disabled using the Troubleshooter Setup function. The relevant Setup display message is *SET-ENABLE xxxxxx?* where xxxxxx is either *WAIT* or *BUSREQ*. Pressing the YES key on the Troubleshooter enables the status line; pressing the NO key disables the status line. The default for both lines is YES -- enabled.

NOTE

During Troubleshooter Setup, selecting the message SET-ENABLE xxxxxx? NO prevents the enable line from affecting the operation of the Pod (although the Pod can still detect whether the line is high or low). This differs from selecting the Troubleshooter Setup message SET-TRAP ACTIVE FORCE LINE? NO which does not prevent an enable line from affecting the operation of the microprocessor, but does prevent the active condition of a disabled line from being reported on the Troubleshooter display.

4-10. Status Flags Generated by the Pod

The Z8000 Pod provides several status flags that do not represent particular Z8000 signals. These flags are used to provide helpful information to the operator. The Pod-generated flags are: Power Fail, Timeout, Non-Maskable Interrupt Acknowledge, Non-Vectored Interrupt Acknowledge, Vectored Interrupt Acknowledge, and Segment Page Address Translation Trap Acknowledge,

4-11. POWER FAIL STATUS FLAG

The Power Fail Status Flag is set high by the Pod whenever the UUT power supply voltage drops below 4.5V or rises above 5.5V. This flag is sensed by the mainframe and, if set, causes a *BAD POWER SUPPLY* message to be displayed on the Troubleshooter.

4-12. TIMEOUT FLAG

The Timeout Flag is set high by the Pod whenever a Pod timeout error occurs. It indicates that a UUT access was prematurely aborted by the Pod's watchdog timer. This will occur if the Pod is in the Fast mode (see the description of Special Address F000 0017) and the $\overline{\text{WAIT}}$ line is stuck low.

4-13. NON-MASKABLE INTERRUPT ACKNOWLEDGE FLAG

The Non-Maskable Interrupt Acknowledge Flag is set high by the Pod whenever the CPU processes a Non-Maskable Interrupt.

4-14. NON-VECTORED INTERRUPT ACKNOWLEDGE FLAG

The Non-Vectored Interrupt Acknowledge Flag is set high by the Pod whenever the CPU processes a Non-Vectored Interrupt.

4-15. VECTORED INTERRUPT ACKNOWLEDGE FLAG

The Vectored Interrupt Acknowledge Flag is set high by the Pod whenever the CPU processes a Vectored Interrupt.

4-16. SEGMENT PAGE ADDRESS TRANSLATION TRAP ACKNOWLEDGE FLAG

The Segment Page Address Translation Trap Acknowledge Flag is set high by the Pod whenever the CPU processes a Segment Page Address Translation Trap interrupt.

4-17. Forcing Lines

Forcing lines are a special category of status lines which, when active, can force the microprocessor into some specific state or action.

The following signals are classified as forcing lines on Z8000 microprocessors: $\overline{\text{RESET}}$, $\overline{\text{WAIT}}$, $\overline{\text{STOP}}$, $\overline{\text{BUSREQ}}$, and on the Z8003 and Z8004 only, $\overline{\text{ABORT}}$. The status bits for these functions are shown on the Pod decal and in Table 4-2.

If one of these lines is asserted, the Troubleshooter displays the error message *ACTIVE FORCE LINE (@ aaaa)-LOOP?*. The *ACTIVE FORCE LINE* error message helps isolate status lines which are not functioning properly.

Notice that two of the forcing lines, $\overline{\text{WAIT}}$ and $\overline{\text{BUSREQ}}$, are user-enablable lines. If these user-enablable lines are disabled (via the Setup function or a Special Address), their inputs to the Pod microprocessor are disabled, but the Pod continues to monitor their condition; if they are asserted, the Pod reports to the Troubleshooter that a forcing line is active. If these lines are enabled, they are not considered forcing lines, even when they are active, and no *ACTIVE FORCE LINE* message will be displayed.

4-18. Interrupt Lines

Interrupt inputs to the Z8000 consist of the four status lines $\overline{\text{NMI}}$, $\overline{\text{NVI}}$, $\overline{\text{VI}}$, and $\overline{\text{SEGT}}$($\overline{\text{SAT}}$). The Pod will enable these interrupt lines and gather interrupt information if interrupts have been enabled using the Troubleshooter SETUP function. For more detail, refer to the section titled Interrupt Handling.

NOTE

The reporting of interrupt request lines is disabled at power on. Reporting of active interrupt lines is enabled by selecting the Troubleshooter Setup function message SET-TRAP ACTIVE INTERRUPT? and pressing the YES key.

4-19. User-Writable Control Lines

The Z8000 has a control line which the Troubleshooter can set high or low with the Write Control function. This feature is used by Bus Test to check a line which cannot be toggled by normal read and write operations. It is also useful for helping troubleshoot these lines. The Write Control function is described in the following paragraphs as it pertains to the Z8000 Pod. Note that the Write Control function only sets a line low (active) for one UUT bus cycle, just long enough to verify that it can be driven.

The Write Control and Data Toggle Control Troubleshooter functions require the entry of binary digits to specify the desired level of each user-writable control line.

The one user-writable control line in the Z8000 Pod is $\overline{\text{BUSACK}}$. To drive the $\overline{\text{BUSACK}}$ line low, use a *WRITE @ CTL = 0* command.

4-20. Control Line Bit Assignments

When performing a Bus Test or various other Troubleshooter functions, the Troubleshooter may detect that one or more control lines are not drivable. For example, the Troubleshooter might detect that the $\overline{\text{DS}}$ line is not drivable. The Troubleshooter will then display the message *CTL ERR 00000000 00100000-LOOP?*. The zeros and ones correspond to the bit numbers assigned to the control lines as listed in Table 4-2 and on the label on the back of the Pod. Bit 5 is set to 1 because the $\overline{\text{DS}}$ line was detected as not drivable. All error messages that pertain to non-drivable control lines use the same bit number assignments as listed in Table 4-2.

4-21. SPECIAL FEATURES AND CONTROLS OF THE Z8000 POD

The Z8000 Pod offers several special functions which enhance its usefulness. These special functions reside in the Pod rather than the Troubleshooter and are accessed by reading or writing to special addresses outside the standard address space of the Z8000 microprocessor. The special addresses are listed in Table 4-3.

4-22. QUICK FUNCTIONS

The Pod can perform three "quick" functions: the Quick-Looping Read and Write, the Quick RAM Test, and the Quick ROM Test. As their names imply, the advantage of the Quick functions is that they execute faster than the corresponding mainframe functions (Looping Read and Write, RAM Test and ROM Test). The software routines that control Quick functions reside in the Pod and not in the Troubleshooter, reducing communication overhead and greatly reducing execution time. The special addresses are listed in Table 4-3.

Table 4-3. Special Addresses

ADDRESS	DESCRIPTION
F000 0000	Read $\overline{\text{NMI}}$ acknowledge word
F000 0001	Read $\overline{\text{NVI}}$ acknowledge word
F000 0002	Read $\overline{\text{VI}}$ acknowledge word
F000 0003	Read $\overline{\text{SAT}}$ acknowledge word
F000 0004	Fast-looping read/write at last address
F000 0005	Read/write default high address
F000 0006	Read/write fast RAM increment
F000 0007	Read/write fast RAM start high address
F000 0008	Read/write fast RAM start offset
F000 0009	Read/write fast RAM end high address
F000 000A	Read/write fast RAM end offset
F000 000B	Read/write fast ROM start high address
F000 000C	Read/write fast ROM start offset
F000 000D	Read/write fast ROM end high address
F000 000E	Read/write fast ROM end offset
F000 000F	Read fast RAM error high address/ ROM checksum
F000 0010	Read fast RAM error low address/ inactive ROM bits
F000 0011	Read/write refresh enabled
F000 0012	Read/write refresh rate
F000 0013	Read/write transparent read high address
F000 0014	Read/write transparent read offset
F000 0015	Read/write runuut FCW
F000 0016	Read/write state of $\overline{\text{MO}}$ output
F000 0017	Read/write state of fast mode
F000 0018	Read/write state of continuous interrupt flag
F000 0019	Read last address high errors (no dummy read)
F000 001A	Read last address low errors (no dummy read)
F000 001B	Read last data drivability (no dummy read)
F000 001C	Read last control errors (no dummy read)
F000 001D	Read last forcing line errors (no dummy read)
F000 001E	Read last status (no dummy read)
F000 001F	Read last error summary (no dummy read)
F000 0020	Read/write control drivability mask
F000 0021	Read/write forcing line reporting mask
F000 0022	Read selftest error, write selftest disable

Appendix A in this manual lists a Troubleshooter program that makes the Pod's Quick functions operate, from the perspective of the operator, like standard Troubleshooter functions. Using this program, the operator selects the desired functions, then is prompted for parameters in same manner as the standard Troubleshooter functions. This method may be preferable for some uses over the normal method of loading individual special addresses that is described in subsequent paragraphs.

The program is presented in two forms: as a standard Troubleshooter program, and as a source program for the optional 9010 Language Compiler. The 9010 Language Compiler program is available for several common mainframe computers and controllers. Contact Fluke Customer Service for details.

4-23. Quick-Looping Read or Write

The Quick-Looping Read or Write function is useful for enhanced viewing on an oscilloscope that is synchronized to the TRIGGER OUTPUT pulse (available on the Troubleshooter rear panel). If a signal trace on the oscilloscope screen is dim due to a low repetition rate, the Quick-Looping function can increase the repetition rate to make the signal trace much more visible.

NOTE

The Address Sync mode will synchronize the Troubleshooter TRIGGER OUTPUT to the beginning of the bus cycle. The Data Sync mode may be more useful to check for valid data. Refer to the section titled Probe and Oscilloscope Synchronization Modes for details of the available synchronization modes..

To select the Quick-Looping function, first perform a standard read or write operation at the desired address. Then do a *READ* or *WRITE @ F000 0004*. The Pod first performs a read or write operation in the normal manner, reporting to the Troubleshooter any UUT system errors detected (such as *ACTIVE FORCE LINE*, or *CTL ERR*, etc.); then the Pod enters the Quick-Looping mode where the read or write operation is performed many times faster than the ordinary Looping function specified by pressing the LOOP key on the Troubleshooter keyboard. During the Quick Loop, the Pod does not check for any UUT System errors. Quick-Looping continues until the operator selects another operation.

For example, if the operator specifies the operation *READ @ 1800 0000*, the *READ @ F000 0004* will perform a looping read operation at the address 00 0000 (with status code bits of 1000 and the CPU in the Normal mode). If the operator specifies the operation *WRITE @ 1802 00FE = 8C17, WRITE @ F000 0004 ENTER ENTER* will perform a looping write operation at address 02 00FE (with status code bits of 1000), writing the data 8C17.

The Quick-Looping function may be used for read or write operations at any of the valid Z8000 addresses listed in Table 4-3.

If both error reporting and the Quick-Looping feature are desired, you may apply the ordinary Troubleshooter Looping function to the Quick-Looping read or write, such as *READ @ 1812 3456 ENTER READ @ F000 0004 ENTER LOOP*. The Troubleshooter will command read operations at address 12 3456 at the normal looping speed with full error reporting. For every ordinary read operation, the Pod will interject several Quick-Looping read operations (with no error reporting) which will enhance oscilloscope viewing.

The Special Addresses used with the Quick-Looping function are described in Table 4-5.

Table 4-4. Special Address Summary

ADDRESS	DESCRIPTION
F000 0000	$\overline{NMI}$ Acknowledge Word (paragraph 4-27)
F000 0001	$\overline{NVI}$ Acknowledge Word (paragraph 4-27)
F000 0002	$\overline{VI}$ Acknowledge Word (paragraph 4-27)
F000 0003	$\overline{SAT}$ Acknowledge Word (paragraph 4-27)
F000 0004	Fast Looping Read/Write at Last Address (paragraph 4-23, 4-28)
F000 0005	Default High Address (paragraph 4-29)
F000 0006	Fast RAM/ROM Test Increment (paragraph 4-24, 4-25)
F000 0007	Fast RAM Test Start High Address (paragraph 4-24)
F000 0008	Fast RAM Test Start Offset (paragraph 4-24)
F000 0009	Fast RAM Test End High Address (paragraph 4-24)
F000 000A	Fast RAM Test End Offset (paragraph 4-24) Code — Meaning XX00 — No test requested XXB0 — Busy, R/W test in progress XXB1 — Busy, performing decode test XXC0 — Complete, no errors XXF0 — Fail Read/Write test XXF1 — Failed Address Decode test XX — Indicates the upper byte of the current RAM offset being tested.
F000 000B	Fast ROM Test Start High Address (paragraph 4-25)
F000 000C	Fast ROM Test Start Offset (paragraph 4-25)
F000 000D	Fast ROM Test End High Address (paragraph 4-25)
F000 000E	Fast ROM Test End Offset (paragraph 4-25) Code — Meaning XX00 — No test requested XXB0 — Busy, test in progress

Table 4-4. Special Address Summary (cont)

ADDRESS	DESCRIPTION
	XXC0 Complete, no errors XXC1 Complete, inactive bits XX Indicates the upper byte of the current ROM offset being tested.
F000 000F	Fast RAM Test Error High Address (paragraph 4-25)
	Fast ROM Test Checksum (paragraph 4-24)
F000 0010	Fast RAM Test Error Low Address (paragraph 4-24)
	Fast ROM Test Inactive Bits (paragraph 4-25)
F000 0012	Read Last Test Status (paragraph 4-31)
F000 0013	Run UUT FCW (paragraph 4-32)
F000 0014	Transparent Read High Address (paragraph 4-33)
F000 0015	Transparent Read Offset (paragraph 4-33)
F000 0016	State of $\overline{MO}$ Output (paragraph 4-34)
F000 0017	Refresh Enabled (paragraph 4-35)
F000 0018	Refresh Rate (paragraph 4-36)
F000 0019	State of Fast Mode (paragraph 4-37)
F000 001A	State of Continuous Interrupt Flag (paragraph 4-38)
F000 001B	Last Address High Drivability Error (No Dummy Read) (paragraph 4-39)
F000 001C	Last Address Low Drivability Error (No Dummy Read) (paragraph 4-39)
F000 001D	Last Data Drivability Error (No Dummy Read) (paragraph 4-39)
F000 001E	Last Control Errors (No Dummy Read) (paragraph 4-39)
F000 001F	Last Forcing Line Errors (No Dummy Read) (paragraph 4-39)
F000 0020	Last Status (No Dummy Read) (paragraph 4-39)
F000 0021	Last Error Summary (No Dummy Read) (paragraph 4-39)

Table 4-4. Special Address Summary (cont)

ADDRESS	DESCRIPTION
	SYSTEM FAULT BYTE XXXXXXXX DATA LINES NOT DRIVABLE (BIT 0) ADDRESS LINES NOT DRIVABLE (BIT 1) CONTROL LINES NOT DRIVABLE (BIT 2) FORCING LINES PENDING AND DISABLED (BIT 3) INTERRUPT PENDING (BIT 4) ILLEGAL ADDRESS (BIT 5) SELF TEST (BIT 6) UUT POWER FAILURE (BIT 7) DEFAULT VALUE = FF
F000 0022	Control Drivability Error Reporting Mask (paragraph 4-40)
F000 0023	Forcing Line Error Reporting Mask (paragraph 4-40)
F000 0024	Self Test Diagnostic (paragraph 4-41)
	FFFF — Hex FFFF indicates that the Pod passed the internal self test without any errors being detected.
	ACTIVE FORCE LINE — After receiving an Active Force Line error message, pressing the MORE key on the Troubleshooter will provide a bit map showing the status lines. Refer to Figure 4-2, or the Pod decal, for forcing line bit assignments. A "1" indicates a defective status line.
	CTL ERR — Press the MORE key to display a bit map of the control lines that the Pod self test has determined are probably faulty. A "1" signifies a bad control line.
	ADDR ERR — The MORE key displays a map of the address/data lines that failed a simple read/write test (see discussion below).
	DATA ERR — The MORE key will display a map of the address/data lines that failed a simple drivability test (see discussion below).
	BAD PWR SUPPLY — The Pod has measured an out-of-tolerance power supply voltage.
	1 — The Pod has computed an internal ROM signature that differs from what was expected.
	2 — The Pod has found a Read/Write error in its internal RAM.

Table 4-5. Quick-Looping Functions of the Z8000 Pod

FUNCTION	SPECIAL ADDRESSES AND OPERATIONS	DESCRIPTION OF USE
Quick Looping Write	Write @ XYYY YYYY = ZZZZ	Performs a normal write of data ZZZZ at the address XYYY YYYY. X may only be 0-3 hex.
	Write @ F000 0004 enter enter	Causes the Pod to perform a quick-looping write at the address used in the previous write command. UUT system errors are reported only during the first execution of read or write and not during succeeding executions.
Quick Looping Read	Read @ XYYY YYYY = ZZZZ	Performs a normal read at address XYYY YYYY. X may only be 0-3 hex.
	Read @ F000 0004 enter enter	Causes the Pod to perform a quick-looping read at the address used in the previous read command. UUT system errors are reported only during the first execution of read or write and not during succeeding executions.

4-24. Quick RAM Test

The Quick RAM Test allows the operator to test RAM address blocks more quickly than with the RAM Short test. The Quick RAM Test is considerably faster than the RAM Short test and is almost as rigorous. The Quick RAM test is particularly well suited for programming applications.

The Quick RAM Test consists of two phases; the first test phase is a read-write check, while the second checks address decoding. The read-write check is performed by writing and reading a one and a zero from each bit of each test address to ensure that there are no bits held high or low. After the read-write check is completed, a unique bit pattern has been written to each address. For the address decoding check, the Pod reads each address and compares the read data with the unique word that is expected.

The addressing increment and the starting and ending addresses for the Quick RAM Test are specified in a different manner than for the usual RAM Test. They are entered by writing to the special addresses listed in Table 4-3. The increment (1 for bytes and 2 for words) should be written into F000 0006. To specify the starting address, write the top four digits of the address into special address F000 0007 and the address offset into F000 0008. The top four digits of the ending address segment should be written into F000 0009 and the offset into F000 000A. Either word or byte addresses may be used. The ending address must be greater than the starting address, and both addresses must be even for word addresses. The status code assigned to the beginning address will also be used for the ending address. The address increment value must be even for word addresses.

The Quick RAM Test begins execution as soon as the operator completes the entry of the ending address. During and after execution of the test, the Troubleshooter will not display any information about the progress or results of the test unless requested by the operator. The test may be aborted before completion by selecting another operation.

To determine if the Quick RAM Test is still in progress, or what the test results are, the Troubleshooter operator should perform a *READ @ ENTER* operation (which commands a READ operation at the last entered address). In response, the Pod returns a two-byte word, displayed by the Troubleshooter in hexadecimal format (with leading zeroes suppressed). The lower byte of this word indicates the status of the test or the test results. The status codes and their meanings are shown in Table 4-1. The upper byte of the Pod response shows bits 8 through 15 of the address under test, which allows the operator to monitor the progress of the Pod as it proceeds through the test.

For example, to do a Fast RAM test on a section of memory from XX00 5500 through XX00 7500 with word accesses, use the following procedure:

1. Write the value 2 to special address F000 0006 to ensure that the increment value is set for word accesses,

   WRITE @ F000 0006 = 2

2. Enter the address segment and offset components of the starting address into special addresses F000 0007 and F000 0008 respectively:

   WRITE @ F000 0007 = 0800

   WRITE @ F000 0008 = 5500

3. Enter the segment and offset components of the ending address into special addresses F000 0009 and F000 000A:

   WRITE @ F000 0009 = 0800

   WRITE @ F000 000A = 7510

   Writing the ending offset into location F000 000A will cause the test to begin.

4. You can check on the test with a looping READ operation.

   READ @ ENTER = XXB0

   This command will do a looping read at the previously specified ending address (F000 000A). The XX portion of the result is the upper eight bits of the address offset currently being tested. In this example, it would have started at 15 and incremented as the test progressed until whatever address was specified for ending the test. Referring to Table 4-2, the status B0 indicates that the Read/Write test is in progress and that there have not been any errors up to this point in the test.

For more information about the test results, the operator may specify read operations at the special addresses listed in Table 4-3. It is a good practice to specify the *READ @ ENTER* first to find out if the test has been completed before reading at any of the special addresses. Unless the test has been completed (or failed), the information contained at the special addresses will pertain to a previous test rather than the current test, and the current test will be aborted.

The Special Addresses used with the Quick RAM test are described in Table 4-6.

Table 4-6. Using the Quick RAM Test

SPECIFYING THE TEST	
Write @ F000 0006 = Z	Specifies the address increment to be used by the quick RAM test. If Z=0, the address increment defaults to 2 (word increments).
Write @ F000 0007 = UUUU	Specifies the upper word UUUU of the start address for the quick RAM test.
Write @ F000 0008 = LLLL	Specifies the lower word of the start address (offset) for the quick RAM test.
Write @ F000 0009 = UUUU	Specifies the upper word UUUU of the ending address for the quick RAM test.
Write @ F000 000A = LLLL	Specifies the lower word of the ending address (offset) for the quick RAM test. Execution of the quick RAM test begins at the completion of this command specifying the ending address.
REQUESTING INFORMATION ABOUT TEST EXECUTION	
Read @ enter	After the test has been specified, the operator may request information about test results by pressing the keys READ ENTER (the address specification is defaulted). The resulting code that is displayed on the Troubleshooter indicates the following: Code — Meaning XX00 — No test requested XXB0 — Busy, R/W test in progress XXB1 — Busy, performing decode test XXC0 — Complete, no errors XXF0 — Fail Read/Write test XXF1 — Failed Address Decode test XX — indicates the upper byte of the current RAM offset being tested.
Read @ F000 000F	High word of the error address.
Read @ F000 0010	Low word of the error address.
Read @ F000 0011	Hex Mask of the bad binary bits from a Read/Write failure.
Read @ F000 0012	Last Test Status

4-25. Quick ROM Test

The Quick ROM Test allows the operator to test ROM address blocks more quickly than with the ordinary ROM Test. When the Quick ROM Test is performed, the Pod obtains a checksum that may be compared with a checksum obtained by performing the Quick ROM Test over the same address block of a known good UUT. Note that this checksum is not the same value as the signature that is obtained with the ordinary ROM Test.

The Quick ROM Test is not as rigorous and reliable as the signature analysis used by the ordinary ROM Test, nor does the Quick ROM Test have as extensive error reporting. However, the Quick ROM Test can detect inactive data bits, and the checksum can be used to detect a faulty ROM device with a high degree of confidence.

The Quick ROM Test is specified in a manner similar to the Quick RAM Test. The top four digits of the starting address are written to special address F000 000B and the offset to F000 000C. The top four digits of the ending address are written to F000 000D and the offset to F000 000E. The test begins as soon as the ending offset is entered. The address increment is written to location F000 0006, with 2 (word increment) being the default.

If no upper address is entered (i.e. just the lower four digits) operation of the test is assumed to occur in program space (status line output = hex D).

Only program space (status output = hex C or D) or data space (status = hex 8) accesses will be performed. If the user specifies a ROM test in I/O space (status = hex 2 or 3) or stack space (status = hex 9), it will be mapped to the data space.

Testing of high and low ROM's separately can be achieved by using a byte type operation (bit 29 set in the starting address), setting the increment to 2, and starting the test on either an even or odd address.

The ending address must be greater than the starting address.

Like the Quick RAM Test, the Quick ROM Test may be aborted by selecting another operation. To determine if the Quick ROM Test is still in progress, or what the test results are, the Troubleshooter operator should perform a *READ @ ENTER* operation (which commands a Read operation at the last entered address). In response, the Pod returns a two-byte word, which is displayed by the Troubleshooter in hexadecimal format (with leading zeroes suppressed). The lower byte of this word indicates the status of the test or the test results. The status codes and their meanings are shown in Table 4-2. The upper byte of the Pod response shows bits 9 through 15 of the address under test; therefore, the operator can monitor the progress of the Pod as the test proceeds.

The Special Addresses used in the Quick ROM test are described in Table 4-7.

4-26. SPECIAL FEATURES OF THE Z8000 POD

The following paragraphs describe special features of the Z8000 Pod that are used by reading and writing special addresses. These special functions are summarized in Table 4-4.

NOTE

Uncontrolled assertion or removal of the processor clock provided to the Pod by the UUT can cause alteration of the contents of the special address locations. To ensure reliable Pod operation, control information should be written to the Pod special addresses after UUT power is cycled or the Pod connector is removed and installed in the UUT socket.

Table 4-7. Using the Quick ROM Test

SPECIFYING THE TEST	
Write @ F000 0006 = Z	Specifies the address increment to be used by the quick ROM test. If Z=0, the address increment defaults to 2 (word increments).
Write @ F000 000B = UUUU	Specifies the upper word UUUU of the start address for the quick ROM test.
Write @ F000 000C = LLLL	Specifies the lower word of the start address (offset) for the quick ROM test.
Write @ F000 000D = UUUU	Specifies the upper word UUUU of the ending address for the quick ROM test.
Write @ F000 000E = LLLL	Specifies the lower word of the ending address (offset) for the quick ROM test. Execution of the quick ROM test begins at the completion of this command specifying the ending address.
REQUESTING INFORMATION ABOUT TEST EXECUTION	
Read @ enter	After the test has been specified, the operator may request information about test results by pressing the keys READ ENTER (the address specification is defaulted). The resulting code that is displayed on the Troubleshooter indicates the following: Code — Meaning XX00 — No test requested XXB0 — Busy, test in progress XXC0 — Complete, no errors XXC1 — Complete, inactive bits XX — indicates the upper byte of the current ROM offset being tested.
Read @ F000 000F	Checksum — not related to the ROM signature that is obtained from the standard ROM test.
Read @ F000 0010	Hex mask indicating inactive bits detected during test.
Read @ F000 0012	Last Test Status.

4-27. Interrupt Acknowledge Words (Addresses F000 0000 - F000 0003)

Any data that may be placed on the data bus during an interrupt acknowledge cycle can be read at these addresses. Reading these addresses resets the respective bit in the status word (see Table 4-2 or the Pod decal).

$\overline{\text{NMI}}$ Acknowledge Word (Address F000 0000)

$\overline{\text{NVI}}$ Acknowledge Word (Address F000 0001)

$\overline{\text{VI}}$ Acknowledge Word (Address F000 0002)

$\overline{\text{SAT}}$ Acknowledge Word (Address F000 0003)

4-28. Fast-Looping Read/Write at Last Address (Address F000 0004)

A Troubleshooter *READ* @ or *WRITE* @ with this address initiates a fast-looping read or write at the last address used for a UUT access. Refer to Quick Functions above for complete information.

4-29. Default High Address (Address F000 0005)

The four hexadecimal digits contained here are used as a default high address byte to reduce the amount of keyboard entries required when working in repetitive address spaces. If only four address digits are entered for a *READ* @ or *WRITE* @ specification, this default High Address will be appended ahead of the entered digits to form a complete address.

The initial default value is 0800 (read data space, system mode, word access, segment 0).

4-30. Fast RAM Test and Fast ROM Test Addresses (Addresses F000 0006 - F000 00011)

These special addresses are used to implement the Fast RAM Test and Fast ROM Test functions. Complete information about these tests is contained in the description of Quick Functions in this section.

Fast RAM/ROM Increment (Address F000 0006)

Fast RAM Start High Address (Address F000 0007)

Fast RAM Start Offset (Address F000 0008)

Fast RAM End High Address (Address F000 0009)

Fast RAM End Offset (Address F000 000A)

Fast ROM Start High Address (Address F000 000B)

Fast ROM Start Offset (Address F000 000C)

Fast ROM End High Address (Address F000 000D)

Fast ROM End Offset (Address F000 000E)

Fast RAM Error High Address/ROM Checksum (Address F000 000F)

Fast RAM Error Low Address/ Fast ROM Inactive Bits (Address F000 0010)

Fast RAM Error Bits (Address F000 0011)

4-31. Last Test Status (Address F000 0012)

This special address contains the last status after a Fast ROM Test or a Fast RAM Test. For example, if a Fast test is accidently stopped, a Read from this address will produce the status, which the operator can use to determine whether the test had completed or not.

4-32. Run UUT FCW (Address F000 0013)

This special address contains a Flag and Control Word (FCW) to be used with some Run UUT operations. If a Run UUT operation is done with an address other than the default (0), the contents of this address is inserted into the CPU's FCW register just before control is transferred. This allows the operator to enable interrupts and control the various mode bits by writing to this special address.

4-33. Transparent Read Address Control (Addresses F000 0014 - F000 0015)

To provide standby activity for the UUT, continuous READ operations are performed on the UUT during the time the Pod is preparing for the next access to the UUT.

The default for this operation is 0800 0000 (data space, word operation, system mode, segment=0, offset=0). The configuration of the transparent read operation may be changed by writing new data to the Transparent Read Addresses described below.

Transparent read operations are sometimes referred to as "dummy" reads.

Transparent Read High Address (Address F000 0014)

The data written to this special address is used as the high address component used in the transparent read operation.

Transparent Read Offset (Address F000 0015)

The data written to this special address is used as the offset component of the address used in the transparent read operations.

4-34. State of $\overline{MO}$ Output (Address F000 0016)

The Least Significant Bit of the data of this address echos the Multi-Micro Out ($\overline{MO}$) line. An operator may define the state of the $\overline{MO}$ line by writing to this address or observe the current state of $\overline{MO}$ by reading this address. The default is high (inactive).

4-35. Refresh Enabled (Address F000 0017)

This address contains a flag used to set the Refresh Enable bit in the CPU's RAM Refresh Counter. A zero value sets the Refresh Enable bit to 0 (Refresh disabled). Any non-zero value sets the Refresh Enable bit to 1 (Refresh enabled). The default is ENABLED.

4-36. Refresh Rate (Address F000 0018)

This address contains a value used to set the rate in the Z8000's RAM Refresh Counter. The value (within the range 0-64) is shifted nine places left and inserted directly into the 6-bit rate constant of the Z8000's memory refresh register. The rate constant determines the amount of time between successive dynamic memory refresh cycles.

The refresh rate is calculated as

4 X value X clock period

The default value is hexadecimal F, which results in a refresh every 16 μ sec using a 4 MHz clock frequency.

4-37. State of Fast Mode (Address F000 0019)

The Fast mode prevents the $\overline{\text{WAIT}}$ line from interfering with normal Pod operation. The Fast mode is selected by writing a non-zero value to this location. While the Fast mode is selected, the $\overline{\text{WAIT}}$ line is only honored during UUT accesses, and the Pod will not be allowed to timeout due to a stuck $\overline{\text{WAIT}}$ line. If $\overline{\text{WAIT}}$ is asserted for longer than 128 clock cycles during a UUT access while in the Fast mode, the Timeout status bit will be set, and the access aborted.

The default for the Fast mode is ENABLED.

4-38. State of Continuous Interrupt Flag (Address F000 001A)

Normal interrupt processing, where an interrupt is disabled and a status flag set, may make it difficult to diagnose interrupt difficulties using an oscilloscope. To enable a Continuous Interrupt mode, where the continuous occurrence of interrupts may be used to trigger an oscilloscope or may be generated with a pulser, write a non-zero value to this location.

The default for the Continuous Interrupt mode is DISABLED.

4-39. Last Error Group (Addresses F000 001B - F000 0021)

These special addresses contain various error words that may originate during the immediately previous Pod operation.

Note that reading these words does not update the status (because the normal transparent or dummy Read operation does not occur). The entire set may be read without the contents varying.

Last Error Address Segment (No Dummy Read) (Address F000 001B)

This special address contains the segment component of the last address where an address driveability error was detected.

Last Error Address Offset Error (No Dummy Read) (Address F000 001C)

This special address contains the offset component of the last address where an address driveability error was detected.

Last Data Drivability Error (No Dummy Read) (Address F000 001D)

This special address contains a bit map of any data bits which could not be driven properly during the previous UUT access. For example:

READ @ F000 001D = 0300 OK

shows that two data lines, bits 8 and 9, could not be driven during the last UUT access.

Last Control Errors (No Dummy Read) (Address F000 001E)

This special address contains a bit map of any control lines which the Pod might not have been able to drive properly. (Refer to Table 4-2 or the Pod decal for bit assignments). For example:

READ @ F000 001E = 0040 OK

shows that the Pod was not able to drive bit 6, the B/$\overline{\text{W}}$ line.

Last Forcing Line Errors (No Dummy Read) (Address F000 001F)

This special address contains a bit map of any forcing lines which were detected as active during the last UUT access, but have been previously disabled using the Troubleshooter's Setup command *SET - ENABLE xxxx?* commands. Of the four available forcing lines (five on the Z8003 and Z8004), only two, $\overline{\text{WAIT}}$ and $\overline{\text{BUSREQ}}$, are user enableable. These two user-enableable forcing lines will be the only ones effected by the Setup command. For example:

READ @ F000 001F = 0005 OK

shows that both bit 2 ($\overline{\text{WAIT}}$) and bit 0 ($\overline{\text{BUSREQ}}$) have been disabled. (Refer to Figure 4-1 or the Pod decal for bit assignments.)

Last Status (No Dummy Read) (Address F000 0020)

The status word from the immediately previous Pod operation may be read at this address. The data obtained from this operation maybe different from that obtained with a *READ @ STS* operation, since the *READ @ STS* operation performs a UUT bus read at the programmed default address (see the section titled Default Address), while this operation returns data from the previous UUT operation. The data returned is displayed in hexadecimal rather than binary, as is the case with the *READ @ STS* command, but the status bit assignments are the same. Refer to Table 4-2 or the Pod decal (on the bottom of the Pod) for status line bit assignments.

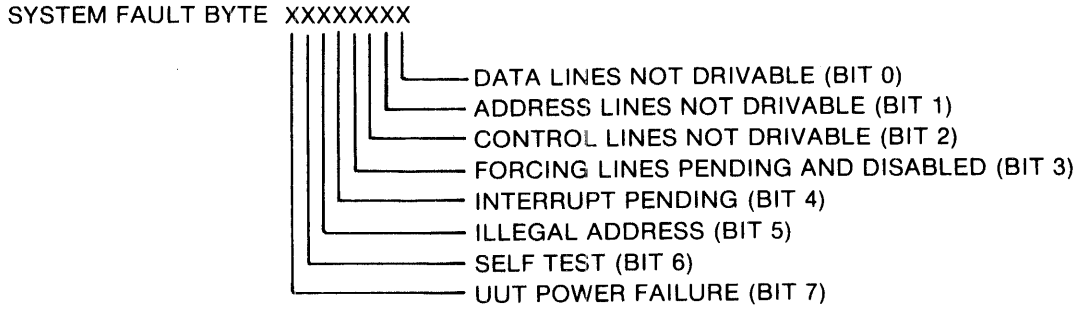
For example,

READ @ F000 0020 = 0314

shows $\overline{\text{WAIT}}$ to be the only active status line. Compare this to the *READ @ STS* example under Status Line Bit Assignments.

Last Error Summary (No Dummy Read) (Address F000 0021)

Contains the System Fault Byte that the Pod returns to the Troubleshooter for error reporting. The user may inhibit the reporting of errors detected by the Pod by using the Setup functions of the Troubleshooter. This address is used to determine the Pod error status even though error reporting by the Troubleshooter has been inhibited. A summary of any errors detected by the Pod during the immediately previous UUT operation may be read from this address. The bit assignments are as follows:



DEFAULT VALUE = FF

For example: If the Pod UUT connector is plugged into the self test socket, the self test is disabled, and all error reporting is inhibited, performing a *READ @ F000 0020 = 0018* indicates that:

1. An interrupt is pending.

2. Forcing line(s) are pending but disabled.

4-40. Error Reporting Masks (Addresses F000 0022 - F000 0023)

These masks control the reporting of Control Line drivability and Forcing Line detection errors. Set any bit in these masks to zero to disable the reporting of errors in that position. The default is all bits ENABLED.

Control Driveability Error Reporting Mask (Address F000 0022)

The reporting of any individual drivability error may be suppressed by setting the appropriate bits of the control drivability mask to zero. The bit assignments correspond to those shown in Table 4-2, Status and Control Line Bit Assignments. The complete error summary can be read from the Last Control Errors special address. Errors corresponding to the suppressed bits will not be reported by the Troubleshooter.

Default value = FFFF.

For example, a certain Z8000 UUT may not allow the processor to drive the $\overline{MO}$ line low. If this is considered normal, performing a *WRITE @ F000 0022 = FBFF* will inhibit the reporting of $\overline{MO}$ line drivability errors during BUS TEST, while allowing drivability error reporting for all of the other control lines.

Forcing Line Reporting Mask (Address F000 0023)

The reporting of any individual forcing line error (e.g., forcing lines asserted but not enabled) may be suppressed by setting the appropriate bits of the forcing line error mask to zero. The bit assignments correspond to those shown in Table 4-2, Status and Control Line Bit Assignments. The complete error summary can be read from the Last Control Errors special address. Errors corresponding to the suppressed bits will not be reported by the Troubleshooter.

Default value = FFFF

For example, with the Pod in the fast mode, a certain UUT asserts the $\overline{WAIT}$ status line in response to a bus read or write at an unimplemented address. The Troubleshooter BUS TEST operation sends an unimplemented address to the UUT while checking the drivability of the address lines. Performing a *WRITE @ F000 0023 = F7FF* will inhibit the reporting of the timeout flag while allowing forcing line error reporting for the status inputs.

4-41. Self Test Diagnostic (Address F000 0024)

This special address is used for troubleshooting operating errors in the Pod itself. Detailed use of this special address for diagnosing Pod defects is described in Section 6, Troubleshooting.

4-42. DEFAULT ADDRESSES FOR LEARN, BUS TEST, AND RUN UUT

Most Troubleshooter operations require operator entry of address information. If the information is not specified, the Troubleshooter supplies default address information. The following paragraphs describe default addresses that are unique to the Pod for the Learn operation, Bus Test, and Run UUT mode. Other default addresses not mentioned in this manual are described in the Troubleshooter operator manual and apply to all Pods.

4-43. Learn Operation Default Address

If the Learn operation is selected and the operator does not specify the starting and ending addresses for the operation, the Pod specifies the default address spaces of 0D00 0000 through 0D00 FFFE. The Learn operation is performed over these address spaces and also 0800 0000 through 0800 FFFE. It might be wise to specify a smaller address range(s) if possible, to avoid making the Troubleshooter take a long time to learn such a large memory space.

4-44. Bus Test Default Address for Data Line Testing

When selecting the Bus Test, no address is explicitly specified by the operator. However, as part of Bus Test, the data lines are tested at a particular address supplied by the Troubleshooter. For the Z8000 Pod, the data line testing occurs at address 0800 FFFE unless otherwise specified. The operator may change this address with the Troubleshooter Setup function by entering the desired address for the Setup message *SET-BUS TEST @ 0800 FFFE-CHANGE?*

4-45. Run UUT Mode

The Run UUT mode allows the Pod to emulate the UUT microprocessor by executing a program directly from UUT memory. When the operator selects Run UUT, the operator may either explicitly specify the address where execution begins or use the Run UUT default execution address which is supplied by the Pod. The default execution address is 00 0000, but may be changed by entering the device address for the Setup message *SET-RUN UUT @ xx00 0000 CHANGE?*. Run UUT at the 00 0000 default address will cause the Pod to start execution as it would if the UUT were reset. That is, the contents of the first two or three words (depending upon the version of the microprocessor) starting at location 0002 will be used as the initial Flag and Control Word and Starting Address.

4-46. INTERRUPT HANDLING

Using the Setup function of the Troubleshooter, the operator has the option of enabling or disabling interrupt reporting. To enable interrupt reporting by the Troubleshooter, use the Troubleshooter Setup function *SET - TRAP ACTIVE INTERRUPT? YES.*

A check of the interrupt status flags (status bits 12 - 15) with a *READ @ STS* operation will indicate whether interrupt information is available.

The contents (if any) of the data bus during an interrupt acknowledge cycle may be read at special addresses F000 0000 - F000 0003.

Special address F000 0018 provides the capability to enable continuous interrupts, such as might be needed to trigger an oscilloscope. Refer to Special Features of the Z8000 Pod for details.

4-47. PROBE AND OSCILLOSCOPE SYNCHRONIZATION MODES

The operator may use the Troubleshooter Synchronization function (selected with the SYNC key) to synchronize probe operation and rear panel TRIGGER OUTPUT pulses to the Pod's microprocessor bus events. The Pod generates a sync signal which is used by the mainframe for the probe and trigger output signals. With the Z8000 Pod, there are four synchronization modes available:

A = Address Sync

D = Data Sync

F = Free-Run

1 = Interrupt Sync

In the Address Sync mode, the sync pulse goes low at the beginning of the UUT bus cycle. The sync pulse goes high with the rising edge of the $\overline{AS}$ signal.

In the Data Sync mode, the sync pulse goes low when the $\overline{AS}$ signal goes high and goes high with the rising edge of the $\overline{DS}$ signal.

In the interrupt sync mode, the Pod sync signal will be similar to that of the data sync mode, but will occur only during an interrupt acknowledge bus cycle.

If Free-Run is selected, then a sync pulse of 2 μsec duration occurring at a frequency of approximately 1 kHz is generated by the mainframe.

If the signal image on the oscilloscope is dim because of a low repetition rate, use the Quick-Looping function described in a previous section to increase the repetition rate and make the signal on the oscilloscope easier to see.

Note that the oscilloscope trigger output pulses are always synchronized to either Address/Data sync or Interrupt sync, even if Free-Run is selected. If Free-Run is selected, the oscilloscope trigger output pulses remain synchronized to the previous sync mode selected. At power on the probe is in Free-Run, but the oscilloscope trigger output pulses are synchronized to Data sync.

Note

The Z8000 Interface Pod is only designed to be used with a Troubleshooters that has been updated with improved delay lines and probes. Earlier models used a slow TTL part as a delay line, which may provide unstable probe readings at the high clock frequencies (possibly greater than 6 MHz) used with the Z8000 CPU. If your Pod is demonstrating such symptoms, you may need to upgrade your Troubleshooter to an improved configuration. Contact a Fluke Technical Service Center for advice.

4-48. PROBLEMS DUE TO A MARGINAL UUT

The Pod is designed to approximate, as closely as possible, the actual characteristics of the microprocessor it replaces in the UUT. However, the Pod does differ in some respects. In general, these differences tend to make marginal UUT problems more visible. A UUT may operate marginally with the UUT microprocessor installed, but exhibit errors with the Pod plugged in. Since the Pod differences tend to make marginal UUT problems more obvious, the UUT is easier to troubleshoot. Various UUT and Pod operating conditions that may reveal marginal problems are described in the paragraphs which follow.

4-49. UUT Operating Speed and Memory Access

Some UUT's operate at speeds which approach the time limits for memory access. The Pod contributes a slight time delay which causes memory access problems to become apparent.

4-50. UUT Noise Levels

As long as the UUT noise level is low enough, normal operation is unaffected. Removing the UUT from its chassis or case may disturb the integrity of the shielding to the point where intolerable noise could exist. The Pod and Pod cable may introduce additional noise. In general, marginal noise problems will actually be made worse (and easier to troubleshoot) through use of the Pod and Troubleshooter.

4-51. Bus Loading

The Pod loads the UUT slightly more than the UUT microprocessor. The Pod also presents more capacitance than the microprocessor. These effects tend to make any bus drive problems more obvious.

4-52. Clock Loading

The Pod increases the normal load on the UUT clock. While this loading will rarely have a significant effect on clock operation, it may make marginal clock sources more obvious.

4-53. POD DRIVE CAPABILITY

As a driving source on the UUT bus, the Pod provides equal to or better than normal Z8000 current drive capability. All Pod inputs and outputs are TTL compatible.

4-54. LOW UUT POWER DETECTION

The Pod has a UUT power detection circuit which constantly monitors the UUT power supply. If the UUT power supply drops below 4.5V or rises above 5.5V, this circuit produces a POWER FAIL output to the Troubleshooter which causes the Troubleshooter to display a *BAD POWER SUPPLY* error message.

The POWER FAIL output can be ignored by changing the Setup command *SET - TRAP BAD POWER SUPPLY? YES* to NO.

Also, anytime the UUT power supply drops below about 3.4V, all active Pod outputs are disabled or written to their low logic level. This feature has been incorporated to protect UUT circuits from possibly being damaged by Pod outputs when the UUT power supply drops below safe operating limits. The Troubleshooter will display a UUT power fail error message. When the proper operating power supplies have been restored to the UUT, the outputs of the Pod will return to normal and the Troubleshooter will be ready for additional testing.

Section 5
Theory of Operation

5-1. INTRODUCTION

The theory of operation of the Pod is described on two levels. The first level is an overall functional description which describes the major sections of the Pod and how they relate to each other, to the UUT, and to the Troubleshooter. The second level is a detailed block diagram of each Pod section. The descriptions are supported by block diagrams in this section and by complete instrument schematics in Section 8 of this manual.

5-2. GENERAL POD OPERATION

The Pod is essentially a complete microprocessor system by itself. It is usually in a "housekeeping" mode, waiting for instructions from the Troubleshooter. When the Pod receives an instruction, it performs an operation or series of operations on the UUT microprocessor bus, using a bus switch approach. Under normal operating conditions, when the Pod is in communication with the Troubleshooter, it functions like any normal microprocessor-controlled system. However, when the Pod accesses the UUT, the bus is momentarily (for the duration of a memory access cycle or an I/O cycle) switched to the UUT by disabling the components in the Pod and connecting all lines to the UUT, buffered in the appropriate direction.

When the Pod emulates the UUT microprocessor in the Run UUT mode, the components within the Pod are permanently disabled, and the Pod microprocessor is effectively permanently connected to the UUT.

The Pod may be divided into the following three major sections:

Processor Section

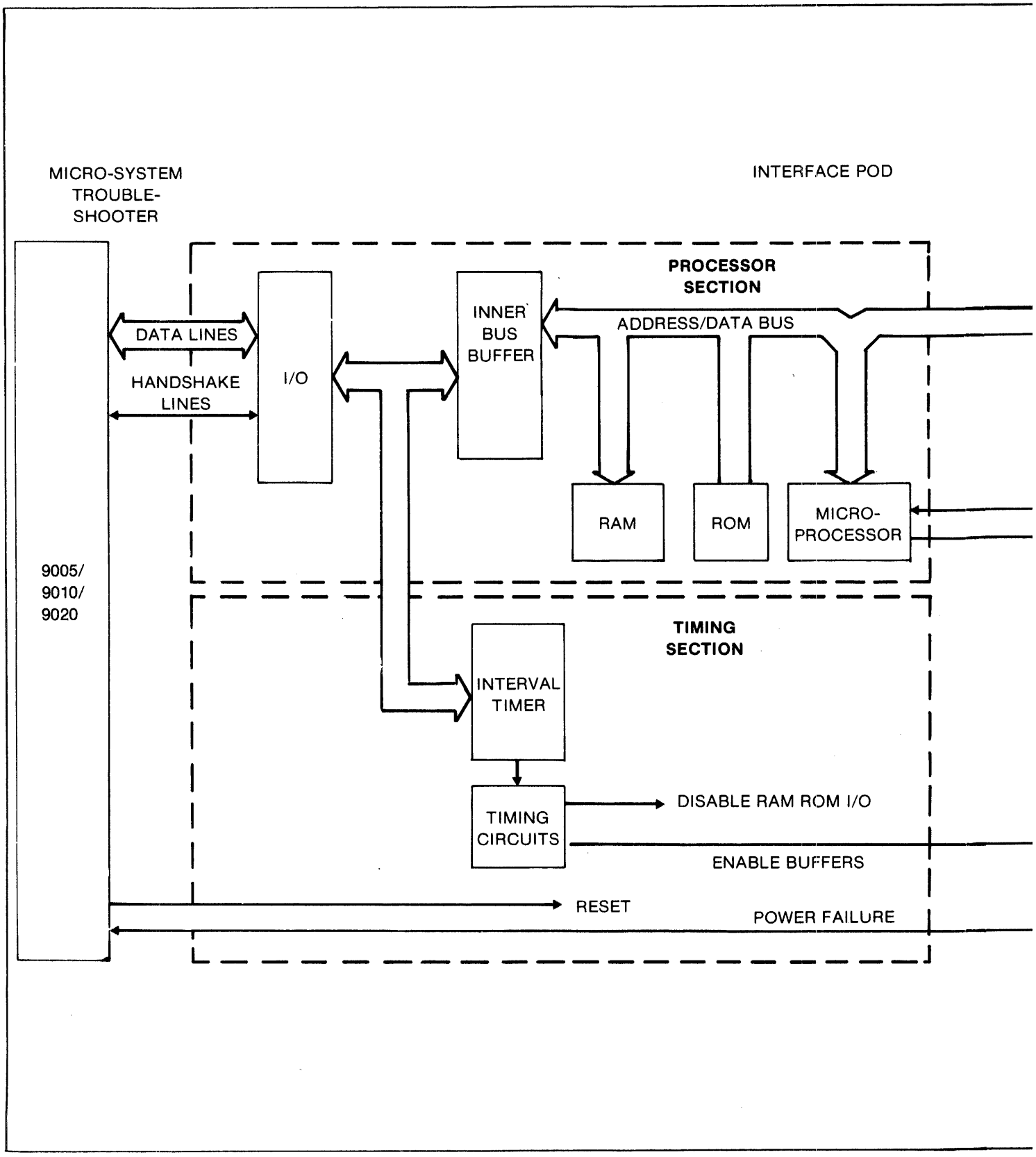
UUT Interface Section

Timing and Control Section

Each of these three sections are described in the following paragraphs.

5-3. Processor Section

The Processor Section, shown in Figure 5-1, consists of the microprocessor, RAM, ROM, I/O, and various latches and buffers. These elements, along with some timing components, constitute a small microsystem that performs specific operations in response to commands from the Troubleshooter.


MICRO-SYSTEM TROUBLE-SHOOTER
INTERFACE POD
PROCESSOR SECTION
DATA LINES
HANDSHAKE LINES
I/O
INNER BUS BUFFER
ADDRESS/DATA BUS
RAM
ROM
MICRO-PROCESSOR
9005/ 9010/ 9020
TIMING SECTION
INTERVAL TIMER
TIMING CIRCUITS
DISABLE RAM ROM I/O
ENABLE BUFFERS
RESET
POWER FAILURE

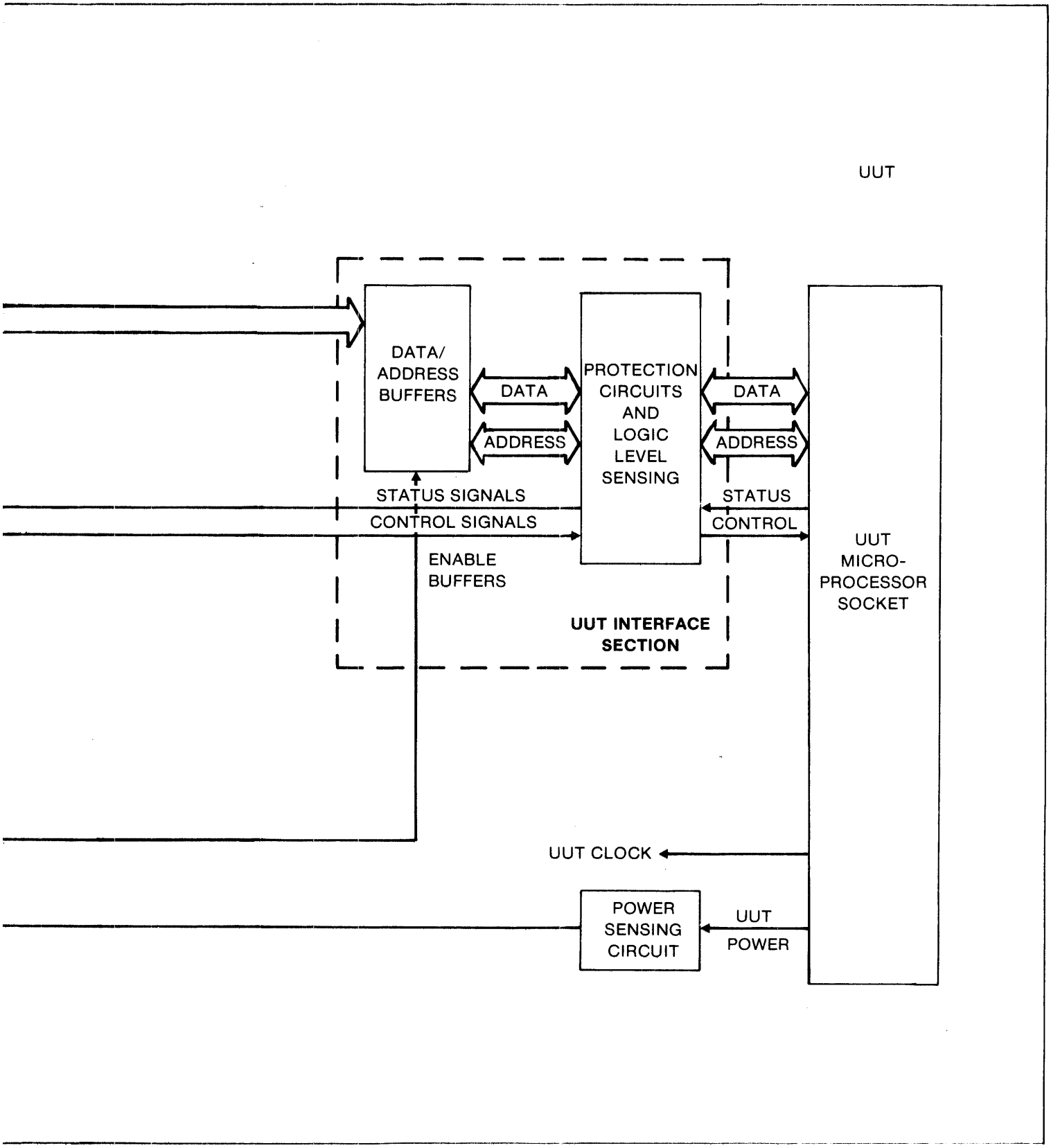


Figure 5-1. General Block Diagram

Any of the four Z8000 microprocessor types (Z8001, Z8002, Z8003, or Z8004) may be used. A switch on the Pod is provided to enable or disable the $\overline{\text{ABORT}}$ line used on the Z8003 and Z8004. 48-to-40 pin adapters are provided to covert the 40-pin Z8002 and Z8004 versions to fit the normal 48-pin socket. The microprocessor must be rated for at least 6 MHz clock speed in order to use the built-in Pod self test.

The microprocessor inputs received from the UUT are referred to as status lines. The microprocessor outputs generated by the Pod are referred to as control lines. Although this nomenclature is not always in agreement with the manufacturer's literature, the convention allows consistency between Pods when implementing the Troubleshooter functions.

Since malfunctioning status lines, such as $\overline{\text{WAIT}}$, could prevent the Pod from performing tests, all incoming status lines which could adversely affect the Pod operation are either automatically disabled by the Pod, or may be disabled by the operator using the Troubleshooter Setup function. The one microprocessor input which may not be disabled, of course, is the UUT clock. The clock signal must always be present for Pod operation. All the status lines are enabled in the Run UUT mode.

The Processor Section also contains circuitry for Pod self test. When the Pod ribbon cable plug is inserted into the self test socket, part of the Pod circuitry becomes a simplified pseudo UUT. During Pod self test, certain tests are performed on this pseudo UUT, and any failures are reported to the Troubleshooter.

5-4. UUT Interface Section

The Interface Section, shown in Figure 5-1, consists of buffers and drivers, protection circuits, logic level detection circuits, and a UUT power sensing circuit. The buffers and drivers switch the UUT to the microprocessor or to the standby control and address signals, as dictated by the Timing and Control Section.

Each UUT interface line is protected from overvoltage or short circuit conditions that might damage Pod components. Resistors in series with the inputs of the detection circuit latches limit the input current, and resistors in series with the output drive lines limit output current. A pair of clipping diodes connected to ground and +5V protect against incorrect voltages.

The detection circuits consist of latches that are clocked during a UUTON cycle, when the signals are expected to be at a known level.

If a signal cannot be driven through the current-limiting resistor, it will be detected when the latches are individually read by the Processor Section and the values are compared with the expected values.

The UUT power sensing circuit shown in Figure 5-1 constantly monitors the UUT power supply. This circuit produces an output to the Troubleshooter in the event UUT power drops below 4.5V or rises above 5.5V.

Anytime the UUT power supply drops below about 3.4V, all active Pod outputs are disabled or written to their low logic level. This feature protects UUT circuits from being damaged by Pod outputs when the UUT power supply drops below safe operating limits. The Troubleshooter will display a UUT power-fail error message. When the proper operating power has been restored to the UUT, the outputs of the Pod will return to normal, and the Troubleshooter will be ready for additional testing.

5-5. Timing and Control Section

The Timing and Control Section, shown in Figure 5-1, consists of a timer and timing and control logic. The Timing and Control Section uses signals from the timer, the output latches as set by the microprocessor, the status lines from the UUT, and the control lines from the microprocessor to switch the bus between the UUT and the Pod internals.

The length of a normal bus switch equals one microprocessor bus cycle. The bus is switched to the UUT between the end of a Pod operation and the start of a UUT operation. The bus switch is initiated by a signal from the timer. The bus is switched back to communication with the Pod at the end of the cycle.

If the microprocessor has sent the Run UUT command through the output latch, the bus switch is started in the normal fashion, but is then held on indefinitely until a reset pulse is received from the Troubleshooter.

During the time the Pod is not communicating with the UUT, the UUT needs the proper signals so that it can perform dynamic memory refresh operations and other similar tasks. In order to provide these signals to the UUT, the Pod performs a read operation at the standby address. This procedure, called a transparent read, generates the transparent or fake control signals required to simulate a normal microprocessor read operation.

5-6. DETAILED BLOCK DIAGRAM DESCRIPTION

Each major section is described in the following paragraphs along with a description of the Self Test circuit. Figure 5-2 is a detailed illustration of the Pod operation.

5-7. Detailed Description of the Processor Section

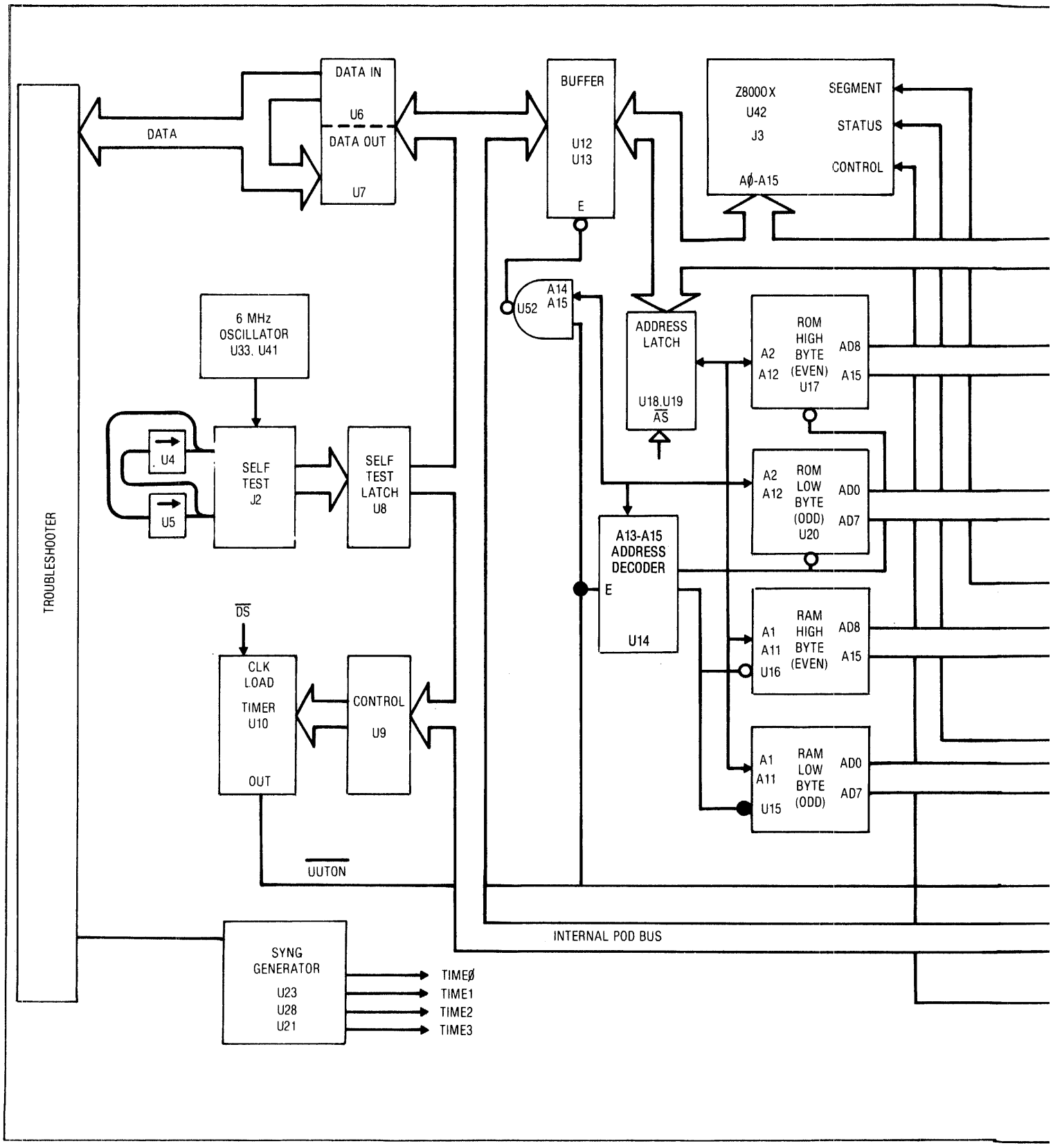
The microprocessor (U42 inserted into J3), RAM (U15, U16), ROM (U17, U20), address latch (U14), and address decoder (U14) form a small microprocessor system which is the heart of the Pod.

Addresses for Pod RAM and ROM components are isolated and demultiplexed from the Z8000 Address/Data bus by the address latches (U18, U19) and the address decoder (U14).

Information from the Z8000 Address/Data bus is passed to an internal Pod data bus by a Bidirectional Buffer (U12, U13). The internal bus carries I/O data to be exchanged with the Troubleshooter, data from self test operations, data for transparent read operations, control data for the Timing and Control section, and data from the UUT signal latches.

Information is exchanged with the Troubleshooter via I/O driver U6 and receiver U7. A control signal from control latch U11 enables U6 to drive the bus that connects the Pod to the Troubleshooter. All communication between the Pod and the Troubleshooter uses the handshake protocol shown in Figure 5-3. The two handshake lines are $\overline{\text{MAINSTAT}}$ and $\overline{\text{PODSTAT}}$. $\overline{\text{MAINSTAT}}$ is driven by the Troubleshooter and monitored by the Pod. $\overline{\text{MAINSTAT}}$ initiates all data transactions and $\overline{\text{PODSTAT}}$ indicates the Pod response.

An external switch (S1) controls the $\overline{\text{ABORT}}$ signal to the microprocessor socket. The $\overline{\text{ABORT}}$ line should be connected when using the virtual memory Z8000 devices (Z8003 and Z8004) and disconnected when using the segmented memory devices (Z8001 and Z8002).


DATA IN
U6
DATA OUT
U7
DATA
BUFFER
U12
U13
E
Z8000X
U42
J3
A∅-A15
SEGMENT
STATUS
CONTROL
TROUBLESHOOTER
6 MHz
OSCILLATOR
U33, U41
U4
U5
SELF
TEST
J2
SELF
TEST
LATCH
U8
U52
A14
A15
ADDRESS
LATCH
U18,U19
AS
ROM
HIGH
BYTE
(EVEN)
U17
A2
A12
AD8
A15
ROM
LOW
BYTE
(ODD)
U20
AD0
AD7
A13-A15
ADDRESS
DECODER
E
U14
RAM
HIGH
BYTE
(EVEN)
U16
A1
A11
RAM
LOW
BYTE
(ODD)
U15
DS
CLK
LOAD
TIMER
U10
OUT
CONTROL
U9
UUTON
INTERNAL POD BUS
SYNG
GENERATOR
U23
U28
U21
TIMEØ
TIME1
TIME2
TIME3

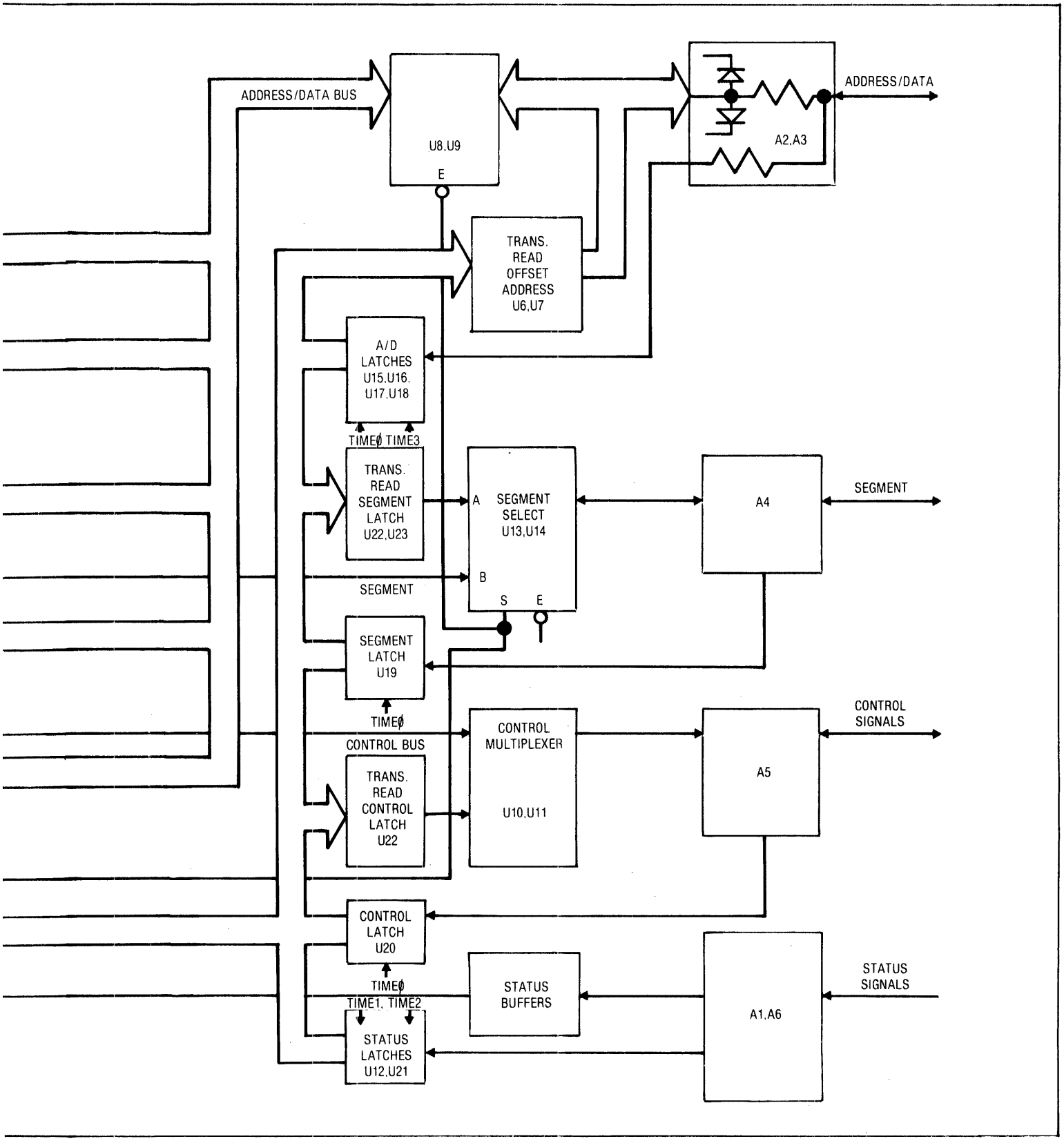


Figure 5-2. Detailed Block Diagram

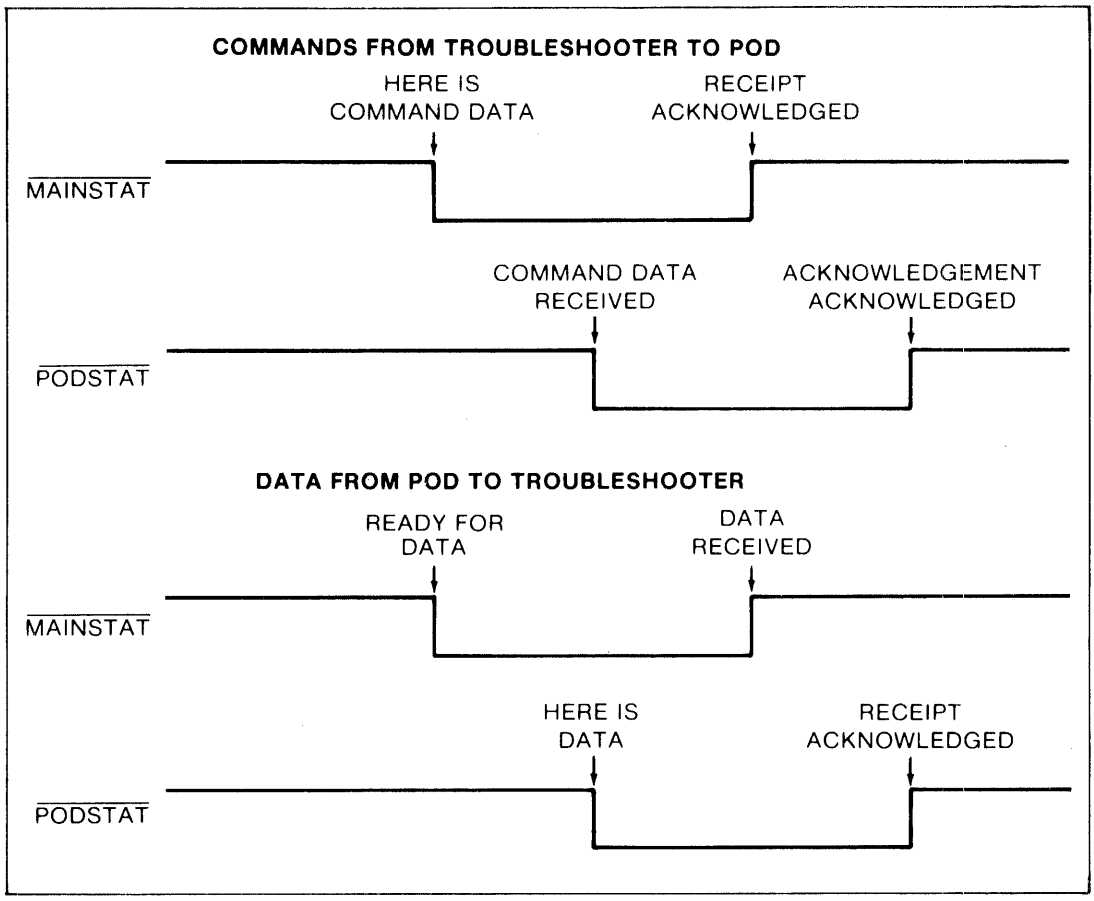


Figure 5-3. Handshake Signals

5-8. Detailed Description of the UUT Interface Section

Each UUT interface line is protected by a 700-ohm resistor in series with the inputs of the detection circuit latches to limit the input current, and a 100-ohm resistor in series with the output drive lines to limit output current. A pair of clipping diodes connected from the interface line to ground and +5 volts protect against incorrect voltages. Devices A1 through A8 are Fluke-designed hybrid circuits containing the current-limiting resistors and voltage-limiting clipping diodes.

The detection circuits consist of latches that are clocked during a UUTON cycle, when the signals are expected to be at a known level. Operation of these latches is described in detail in Detailed Description of the Timing and Control section.

Latches U12 and U15-U20 monitor the logic states of all of the microprocessor lines via the input protection circuits. The latches are clocked at appropriate times during the UUT access cycle to latch their respective signals and hold these logic states for later examination.

Tri-state latches U6, U7 and U22, U23 are used to supply an address for the transparent read operation that is done whenever the Pod is not performing a UUT access. The address is contained in Special Address locations in Pod RAM. The eight segment bits of this address (U22, U23) are contained in special address F000 0013. The 16 offset bits (U6, U7) are contained in special address F000 0014. Refer to Special Features for the Z8000 Pod in Section 4 of this manual for information about the Address components for transparent read operations.

If a signal cannot be driven through the 100-ohm resistor, it will be detected when the latches are individually read by the Processor Section and the values are compared with the expected values.

Comparators U24 and its associated circuitry monitor the UUT power supply voltage. U24 sends an error signal ($\overline{\text{PFAIL}}$) to the Pod and the Troubleshooter if the UUT power supply voltage is not between 4.5V and 5.5V. This signal indicates that the UUT power supply is operating improperly. To prevent possible damage to the UUT, U24 also inhibits all Pod outputs from going high if the UUT power supply voltage falls below approximately 3.4V ($\overline{\text{LOPWR}}$).

Two flip-flops, both parts of U5, check activity on the $\overline{\text{AS}}$ and $\overline{\text{DS}}$ lines. The lines are sampled to confirm that they are not stuck. Simple delay circuits (based around transistors Q1 and Q2) are used to ensure an accurate sample of the $\overline{\text{AS}}$ and $\overline{\text{DS}}$ lines. The edges of the AS and DS lines are delayed sufficiently to allow accurate check of their signal states.

5-9. Detailed Description of the Timing and Control Section

Timing and control functions occur in two alternating cycles (called the "context"): the internal Pod cycle and the UUT access (UUTON) cycle. When in the Pod context, the CPU performs Pod functions; exchanging information with the Troubleshooter and performing tasks dictated by Troubleshooter commands. During the UUT context, the CPU performs UUT functions as it would if it were plugged into the UUT directly. Signal timing is shown in Figure 5-4.

While the Pod is in the Pod context, a "transparent" read operation is performed on the UUT. This dummy operation allows the UUT to maintain its memory refresh processes and provides signs of activity for some UUT's which may have internal monitoring for CPU failure.

Switching between the two contexts is done by a context-control counter (U10). After being enabled by the $\overline{\text{LOAD}}$ line, U10 counts the falling edges of $\overline{\text{DS}}$ cycles. After the correct number of $\overline{\text{DS}}$ cycles, the RIPPLE OUT output line goes low. The next rising edge of the $\overline{\text{DS}}$ will provide a context change to the UUT access cycle for one $\overline{\text{DS}}$ cycle.

A Run UUT command causes the context to change to continuous UUT accesses until the Pod is reset by the Troubleshooter. Any Troubleshooter command which follows (and therefore, cancels) a Run UUT command will send a hardware reset to the Pod.

The four status line ST0 through ST3 are monitored to detect UUT memory refresh cycles or interrupt acknowledge cycles. The context is shifted to the UUT for the duration of either type of activity. The context-control ripple counter is disabled during a shift to UUT access caused by memory refresh.

5-10. Pod Context

The Pod context is primarily devoted to exchanging information with the Troubleshooter and performing tasks dictated by Troubleshooter commands. In this state, the CPU is connected to the Pod's internal circuitry (and is executing the program contained in the Pod's PROM's).

During this time, information from the previous UUT access cycle is available by addressing the various signal latches, which have preserved UUT signal levels when the context was shifted back to the Pod.

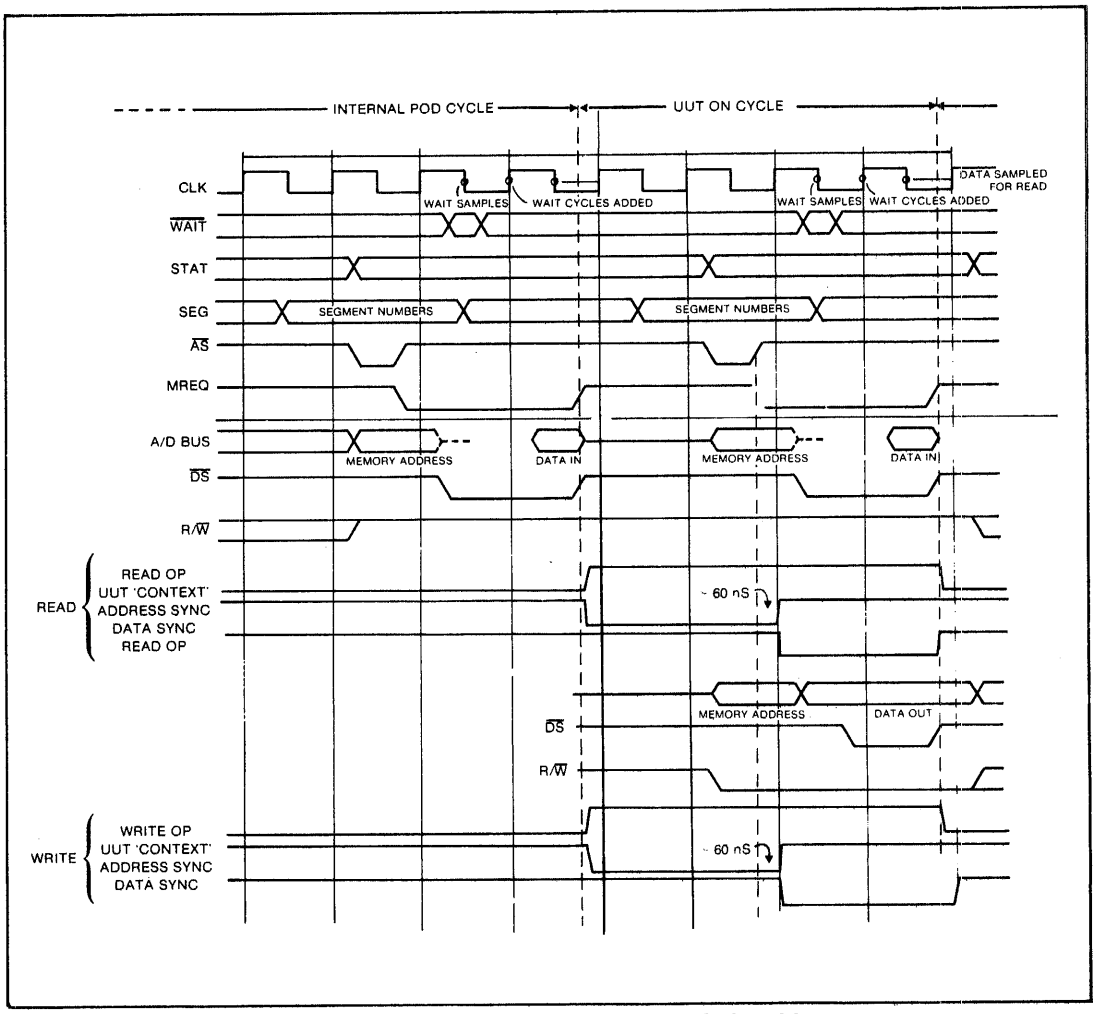


Figure 5-4. Z8000 Signal Timing Relationships

5-11. UUT Context

While in the UUT access context, the Pod microprocessor is functioning as it would as the CPU of the UUT. In addition, Pod latch circuits are monitoring UUT signal activity, in order to provide that information to the Troubleshooter.

The address decoder, U14, is disabled during UUT accesses to keep the Pod inactive.

The three primary latch timing signals are created by the two flip-flops in U21. The three different timings are used to latch information into the various latches at appropriate times.

The timing signal that is keyed to the rising edge of $\overline{AS}$ is used primarily to latch address information. The one keyed to the rising edge of $\overline{DS}$ is used primarily with data information, and the one keyed to the falling edge of $\overline{DS}$ is used for status.

Note that the latch timing generator U21 is only enabled during UUT accesses. This prevents erroneous information being preserved in the various latches during Pod operations.

The four status lines ST0 through ST3 are monitored to detect memory refresh cycles and interrupt acknowledge cycles from the UUT. When these two events occur, the context is transferred to the UUT for the duration of the processes. U22, pin 8, provides the Pod control signal indicating an active refresh cycle or interrupt acknowledge cycle.

The Pod provides sync pulses which will be used for synchronizing the Troubleshooter probe and auxiliary oscilloscopes. The flip-flops in U23 form the sync generator circuit which provides address sync pulses to Troubleshooter. Address sync pulses are only created when in the UUT context. They are delayed by an RC network in conjunction with a CMOS NAND gate, to compensate for a large delay that built into the Troubleshooter's probe data circuit. This is to allow very low duty cycle signals, such as the address data, to be correctly probed at high clock frequencies.

Data and address sync pulses are generated by U23 at the start of the UUT access cycle. Pin 6 of U28 is active low at the beginning of a UUT access cycle if address sync is selected, which makes pin 5 of U23 (the sync output) active. The D input of U23 (pin 2) is held low by U11, making the sync output go inactive at the rising edge of $\overline{AS}$ (U23, pin 3). For data sync operation, the D input (pin 2) is held high, which results in an active sync starting at the rising edge of $\overline{AS}$ and ending with the falling edge of the CONTEXT signal at pin 1. Interrupt sync operation is generated by U35 (pin 12) going low during an interrupt cycle, gated by U38, pin 5.

Sync timing is illustrated in Figure 5-5.

5-12. Detailed Description of the Self Test Circuit

During self test, the Pod appears to the Troubleshooter to be a small UUT with well defined characteristics. The microprocessor's output lines are connected to its input lines to create a configuration where most of lines can be tested. The Troubleshooter performs a series of operations and compares the expected behavior with the actual behavior to determine whether or not the Pod is operating properly.

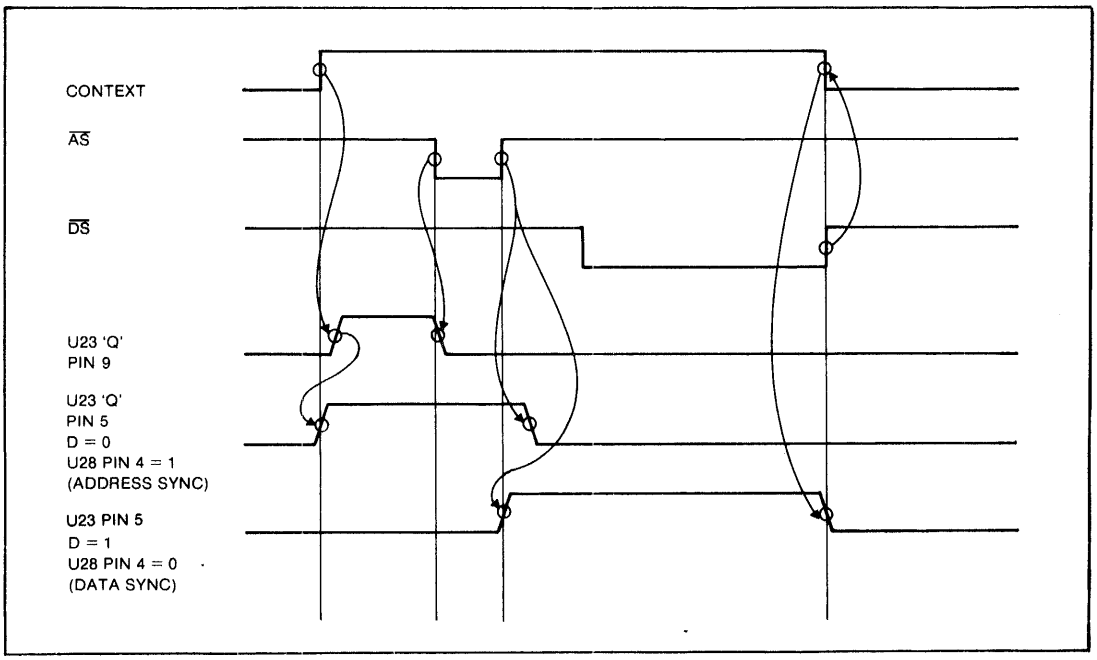


Figure 5-5. Sync Pulse Generation

The self test is initiated when the self test connector's ground pin is grounded. The state of that pin is checked as part of every UUT access. When the UUT cable is plugged into the self test socket, it automatically pulls the connector ground pin (pin 36) to a low logic level, indicating that the Pod should execute the self test routines. This information is also sent to the Troubleshooter, which requests several Pod operations as part of the self test procedure.

Portions of U33 and U41, along with crystal Y1, comprise an oscillator which provides the microprocessor with a 6 MHz clock signal during Pod self test.

The address/data lines AD0-AD7 are cross-connected with lines AD8-AD15 through the two latches U4 and U5. This arrangement allows the self test routines to verify the function of all address and data lines by reading and writing between them. The rising edge of the $\overline{AS}$ line latches the low byte of the address, lines A0 through A7, into U4, while the high byte, lines A8 through A15, is latched into U5. The output of the latches is switched between high and low bytes, so that when the the contents are read (enabled by $R/\overline{W} + \overline{DS}$), lines A0 through A7 read the information that was written as A8 through A15, and lines A8 through A15 receive the original A0 through A7.

To detect drivability problems in the address bus, specifically, shorted lines, the self test software walks a bit pattern along the address bus while monitoring the results.

The segment number lines SN0-SN6 and the $\overline{BUSACK}$ line are cross-connected with the forcing and control lines $\overline{STOP}$, $\overline{WAIT}$, $\overline{BUSREQ}$, $\overline{ABORT}$, $\overline{NVI}$, $\overline{VI}$, $\overline{NMI}$, and $\overline{SEGT(SAT)}$ to allow the self test software to verify segment, status, and control line functions.

$\overline{MO}$, $\overline{MI}$, and $\overline{RESET}$ are tied together for similar cross-connected testing.

The state of status lines ST0-ST3 and the control lines $B/\overline{W}$, $N/\overline{S}$, $R/\overline{W}$, and $\overline{MREQ}$ is latched by the Self Test Status Latch U8.

5-13. Initialization

Whenever the Pod receives a Reset signal, such as when the Pod is first initialized by the Troubleshooter or when Bus Test key is pressed, it sends a string of information to the Troubleshooter. Along with identification information, it transmits a string of default addresses and other information that the Troubleshooter will use for various operations unless otherwise direct by the user:

Address block definitions for LEARN:

0800 0000 - 0800 FFFE program memory
0800 0000 - 0800 FFFE data memory

Default address for Bus Test:

0800 FFFE

Default address for Run UUT:

0C00 0002

lines:

0F

of enableable status lines:

03

Section 6
Troubleshooting

6-1. INTRODUCTION

This section provides troubleshooting information for the Pod, including repair precautions and disassembly procedures.

The built-in Pod Self Test (described in Section 2 of this manual) will detect most Pod malfunctions. Whenever the Troubleshooter displays a message indicating a Self Test error, or whenever the Pod appears to be defective or inoperative, you should make a note of the message or symptoms. If the Pod is still covered under the Warranty, or if you want to have the Pod repaired by Fluke, send the Pod to a Fluke Technical Service Center for repair as described below. If you are going to troubleshoot and repair the Pod yourself, continue to paragraph 6-3, Getting Started.

NOTE

The Z8000 Interface Pod is only designed to be used with a Troubleshooters that has been updated with improved delay lines and probes. Earlier models used a slow TTL part as a delay line, which may provide unstable probe readings at the high clock frequencies (possibly greater than 6 MHz) used with the Z8000 CPU. If your Pod is demonstrating such symptoms, contact a Fluke Technical Service Center for advice.

6-2. WARRANTY AND FACTORY SERVICE

Troubleshooting and repair during the one-year Warranty period should be done by a Fluke Technical Service Center. (See the Warranty statement at the front of this manual for details of the Warranty.) If the Pod is still covered under the Warranty, send the Pod, along with the description of the symptoms, to a Fluke Technical Service Center. The Troubleshooter Operator Manual or Service Manual contains a list of Fluke Technical Service Centers.

After the Warranty period, if you do not want to service the Pod yourself, or if attempted troubleshooting fails to reveal the Pod fault, you may still ship the Pod to a Fluke Technical Service Center for repair at a reasonable cost. If requested, a free cost estimate will be provided before any repair work is performed.

The Pod should be shipped in its original shipping container. If the original shipping container is not available, you may order a new container from John Fluke Mfg. Co., Inc.; P.O. Box C9090, Everett, WA 98206; telephone (206) 342-6300.

6-3. GETTING STARTED

Troubleshooting the Pod is similar to troubleshooting any other microprocessor-based UUT, and requires the equipment listed in Table 6-1. The troubleshooting procedures provided in the following sections are supported by the Theory of Operation in Section 5 and the schematic diagrams in Section 8.

NOTE

All references to data and addresses in the following sections are in hexadecimal notation.

CAUTION

Static discharge can damage MOS components contained in the Pod. To prevent this possibility, take the following precautions when troubleshooting and/or repairing the unit.

- Never remove, install, or otherwise connect or disconnect PCB (printed circuit board) assemblies without disconnecting the Pod from the Troubleshooter.
- Perform all repairs at a static-free work station.
- Do not handle IC's or PCB assemblies by their connectors.
- Wear a static ground strap when performing repair work.
- Use conductive foam to store replacement or removed IC's.
- Remove all plastic from the work area (including vinyl and expanded foam, such as Styrofoam®).
- Use a grounded soldering iron.
- Always place the Pod in a static-free plastic bag for shipping.

6-4. DETERMINING WHETHER THE POD IS DEFECTIVE OR INOPERATIVE

The first task of troubleshooting the Pod is to determine whether it is defective or inoperative. This determination is based on the results of the Pod self test described in Section 2. If you have not performed the self test, refer to Section 2 and perform the self test before proceeding with the troubleshooting.

Table 6-1. Required Test Equipment for Pod Troubleshooting

EQUIPMENT TYPE	REQUIRED TYPE
Micro System Troubleshooter	Fluke 9000 Series
Interface Pod	Fluke 9000A-Z8000
Digital Multimeter	Fluke 8020
Oscilloscope	Tektronix 485 or equivalent

Styrofoam® is a registered trademark of Dow Chemical Co., Inc.

The results of the Pod self test and the Pod behavior when connected to a known good UUT will categorize the problem into one of the three following groups:

- Defective Pod: The Pod fails the Pod self test and the Troubleshooter displays a self test failure code. Refer to Troubleshooting a Defective Pod in this section.

- Inoperative Pod: The Pod is unable to complete the Pod self test and the Troubleshooter displays an *ATTEMPTING RESET* message. Refer to Troubleshooting an Inoperative Pod in this section.

- Suspected Defective Pod: The Pod passes the Pod self test but exhibits abnormal behavior when connected to a known good UUT. Refer to Extended Troubleshooting Procedures in this section.

6-5. TROUBLESHOOTING A DEFECTIVE POD

6-6. Introduction

This section describes what to do if the Troubleshooter displays the *POD SELF TEST Z8000 FAIL xx* (where *xx* represents a self test failure code) message when the Pod self test is performed. If instead, the Troubleshooter displays an *ATTEMPTING RESET* message, refer to Troubleshooting an Inoperative Pod.

The procedures for troubleshooting a defective Pod are based on the information reported by the self test failure codes. These self test failure codes provide information that can enable the operator to locate the cause of the Pod failure.

6-7. Interpreting the Self Test Failure Codes

Introduction

The fact that the self test was completed is a good indication that the problem is probably located in the UUT Interface Section of the Pod. Since the self test was completed, the Processor Section and the Timing Section are probably functioning normally. They are essential for accepting the self test commands and communicating the results to the Troubleshooter. Self Test Failure Codes are described in Table 6-2.

Table 6-2. Standard Self Test Failure Codes

FAILURE CODE	DESCRIPTION
00	UUT read access failed or the enhanced self test failed
01	UUT write access failed
02	Control line(s) cannot be driven
03	Enableable status line(s) failed

Self Test Diagnostic (Address F000 0024)

This special address contains diagnostic information derived from the most recent self test operation. (For a general description of special addresses, refer to Special Functions of the Z8000 Pod in Section 4 of this manual.)

A *READ* @ operation at the special address F000 0024 will return number code or message indicating the type of error (if any) encountered during a Pod self test. The various messages and their meaning in the self test context are as follows:

FFFF	Hex FFFF indicates that the Pod passed the internal self test without any errors being detected.
ACTIVE FORCE LINE	After receiving an Active Force Line error message, pressing the MORE key on the Troubleshooter will provide a bit map showing the status lines that the Pod has determined are probably faulty. Refer to Figure 4-2. or the Pod decal, for forcing line bit assignments. A "1" indicates a defective status line.
CTL ERR	Press the MORE key to display a bot map of the control lines that the Pod self test has determined are probably faulty. A "1" signifies a bad control line.
ADDR ERR	The MORE key displays a map of the address/data lines that failed a simple read/write test (see discussion below).
DATA ERR	The MORE key will display a map of the address/data lines that failed a simple drivability test (see discussion below).
BAD PWR SUPPLY	The Pod has measured an out-of-tolerance power supply voltage.
1	The Pod has computed an internal ROM signature that differs from what was expected.
2	The Pod has found a Read/Write error in its internal RAM.

During the self test, some of the CPU's lines are cross-connected. An apparent error on one line may also be the result of an error on the line connected to it. Therefore, error reported for the following lines are ambiguous and the actual failing line can be either or both the lines in the left and right columns:

Status Line Reported	Additional Possible Failing Lines
$\overline{\text{STOP}}$ (bit 1)	SN0 (line 26)
$\overline{\text{WAIT}}$ (bit 2)	SN1 (line 25)

$\overline{\text{BUSREQ}}$ (bit 0)	SN2 (line 37)
$\overline{\text{ABORT}}$ (bit 3)	SN3 (line 24)
$\overline{\text{NVI}}$ (bit 7)	SN4 (line 42)
$\overline{\text{VI}}$ (bit 8)	SN5 (line 46)
$\overline{\text{NMI}}$ (bit 6)	SN6 (line 47)
$\overline{\text{MI}}$ (bit 10)	$\overline{\text{MO}}$ (line 17) RESET (line 16)
$\overline{\text{SAT}}$ (bit 5)	$\overline{\text{BUSACK}}$ (line 29)

The Pod self test also checks for address/data lines that are shorted to power or ground, and for broken lines. The first two tests write a walking bit down the 16 lines checking for drivability errors. The Troubleshooter reports any such drivability errors as *DATA ERR*. The test for open lines is done by latching the address in a latch at the self test socket and reading back the results for two different addresses that exercise each line both high and low. Any lines not drivable both high and low by the address latches are considered defective and are reported as a *ADDR ERR*. Because high and low bytes on the output of the address latches are switched at the self test socket, it is somewhat ambiguous whether the driving line to the latch or the line being read is the open one and generally both will be reported as bad. For instance, if address/data line AD0 is open, the message *ADDR ERR ADDR BTS 0101* is given. If AD0 is shorted to ground, the message is instead *ADDR ERR ADDR BTS 0101 DATA ERR DATA BTS 0001*. Thus, if both an address and data error are reported, the data message should be given higher confidence.

A *WRITE @* operation with a non-zero value to special address F000 0024 will disable the Pod self test and allow the Pod self test socket to function as a simple UUT. Note that the Pod is an unusual UUT, with crossed data and address lines, and forcing lines connected to segment inputs. Refer to the schematic diagrams in Section 8.

6-8. Preparation for Troubleshooting a Defective Pod

CAUTION

Any adjustment, maintentance, or repair of the opened Pod under voltage shall be avoided as far as possible and, if inevitable, shall be carried out only by a skilled person who is aware of the hazard involved.

Prepare to troubleshoot your defective Pod as follows:

1. Disassemble the Pod, referring to the later section titled Disassembly. It is not necessary to separate the two PCB assemblies at this point. The two PCB assemblies should remain securely fastened together with screws to avoid possible problems with electrical connections between the two PCB assemblies.

2. Look for any obvious problems such as burned components or IC's that are loose in their sockets. Replace components if necessary.

3. Connect the Pod to the Troubleshooter, and insert the ribbon cable plug into the self test socket as shown in Figure 6-1. Rotate the locking knob (next to pin 1 of the Self Test Socket) to close the Self Test socket contacts.

4. Press the Bus Test key on the Troubleshooter to initiate the Self Test, then press the Stop Key. Perform a *WRITE @ F000 0024 = XXXX,* where XXXX is any none-zero value, to disable the Pod Self Test. (The Pod Self Test may be re-enabled by cycling Pod power off and then on, or by a *WRITE @ F000 0024 = 0000.*)

5. Press the Setup key on the Troubleshooter and set the following conditions:

   SET-TRAP BAD POWER SUPPLY? YES

   SET-TRAP ILLEGAL ADDRESS? YES

   SET-TRAP ACTIVE INTERRUPT? NO

   SET-TRAP ACTIVE FORCE LINE? NO

   SET-TRAP CTL ERR? YES

   SET-TRAP ADDR ERR? YES

   SET-TRAP DATA ERR? YES

   SET-ENABLE BUSREQ? NO

   SET-ENABLE WAIT? NO

When the Pod and the Troubleshooter are connected in this configuration (and with Special Address F000 0024 = non-zero to disable the self test), the tests and troubleshooting functions of the Troubleshooter can be applied to the Pod, much like any other UUT. For example, you can perform read or write operations on the UUT (which is actually the self test socket). The Troubleshooter does not know that it is plugged into the Pod.

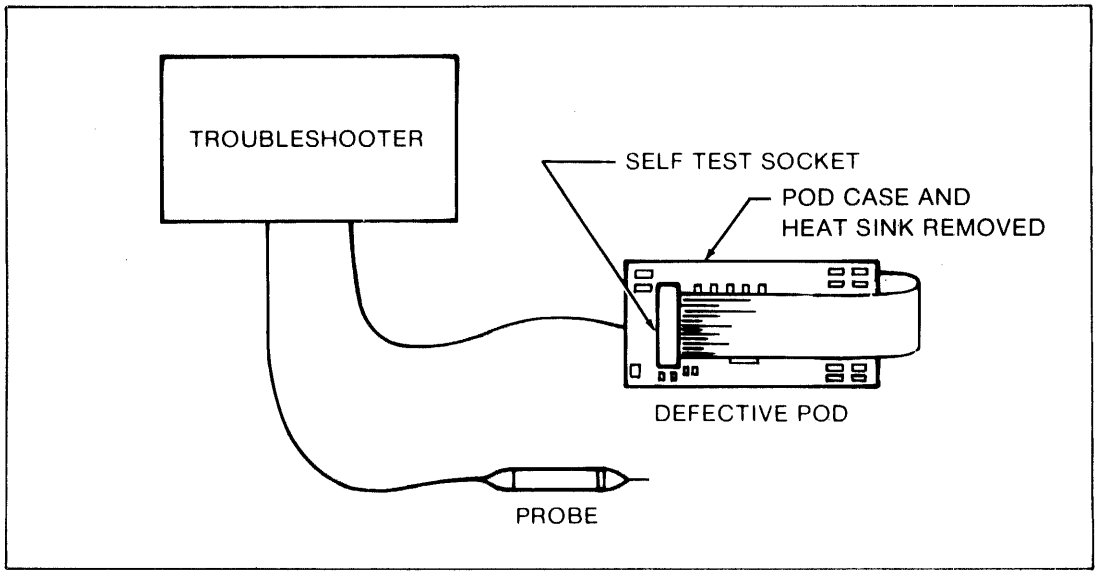


Figure 6-1. Troubleshooting a Defective Pod

6-9. TROUBLESHOOTING AN INOPERATIVE POD

6-10. Introduction

This section describes what to do if the Troubleshooter displays any of the three *ATTEMPTING RESET* messages when the Pod self test is performed. The *ATTEMPTING RESET* messages indicate that the Pod is not operating and is not responding to the Troubleshooter.

If you correct a problem while using the procedures provided in this section, try the Pod self test again. If the Troubleshooter again displays an *ATTEMPTING RESET* message, continue with the procedures in this section. However, if the Troubleshooter displays the message *POD SELF TEST Z8000 FAIL xx*, refer to the previous section titled Troubleshooting a Defective Pod. The reason for referring to the other section is that when the Pod is again communicating with the Troubleshooter, you may use the Pod to help troubleshoot itself.

The procedures in this section apply primarily to the Processor and Timing Sections. (Details of the Processor Section and the Timing Section are described in Theory of Operation in Section 5.)

6-11. Preparation for Troubleshooting an Inoperative Pod

An inoperative Pod is like any other microprocessor-based UUT that is not operating properly; the easiest way to fix an inoperative Pod is by using a Troubleshooter and a good Pod. Preparation instructions also apply to troubleshooting without a good Pod, but note that the detailed troubleshooting steps that follow only apply to using the second Troubleshooter and Pod. Prepare to troubleshoot the inoperative Pod by performing the following steps:

1. Disassemble the Pod, referring to the later section titled Disassembly, but do not separate the two PCB assemblies.

2. Look for any obvious problems, such as burned components or IC's that are loose in their sockets. Replace components if necessary. If such obvious defects are found, it might be prudent to try the self test again at this point.

3. Remove the Pod microprocessor from its socket.

4. If a second Troubleshooter is available, connect the Pod cable plug from the inoperative Pod to the second Troubleshooter to supply the inoperative Pod with power. If a second Troubleshooter is not available, connect a +5V dc (2 amp) power supply to the Pod as shown in Figure 6-2. An easy place to make the power connections is at the connector that usually connects the cable to the Troubleshooter.

CAUTION

Do not operate the Pod with the voltage supply exceeding 5.25 volts, or damage to the Pod could result.

5. If a second Troubleshooter is not used, provide a clock signal for the inoperative Pod by inserting the ribbon cable of the inoperative Pod into its own self test socket. (Make sure the clock is working properly first.) An alternative source for a clock signal is a known good UUT or frequency generator.

6. If a second Troubleshooter is not used, connect pins 2 and 15 to +5V and pin 25 to ground.

7. Connect the Troubleshooter to the good Pod as shown in Figure 6-2. Apply power to the Troubleshooter, then install the ribbon cable plug of the good Pod into the microprocessor socket of the inoperative Pod.

CAUTION

Do not apply or remove power to the good Pod with the ribbon cable connected between the good Pod and the inoperative Pod.

CAUTION

Do not separate the PCB assemblies of the inoperative Pod with power applied to the inoperative Pod. Failure to comply with this can damage CMOS components in the Pod. The PCB assemblies should be securely fastened together with the proper screws before applying power.

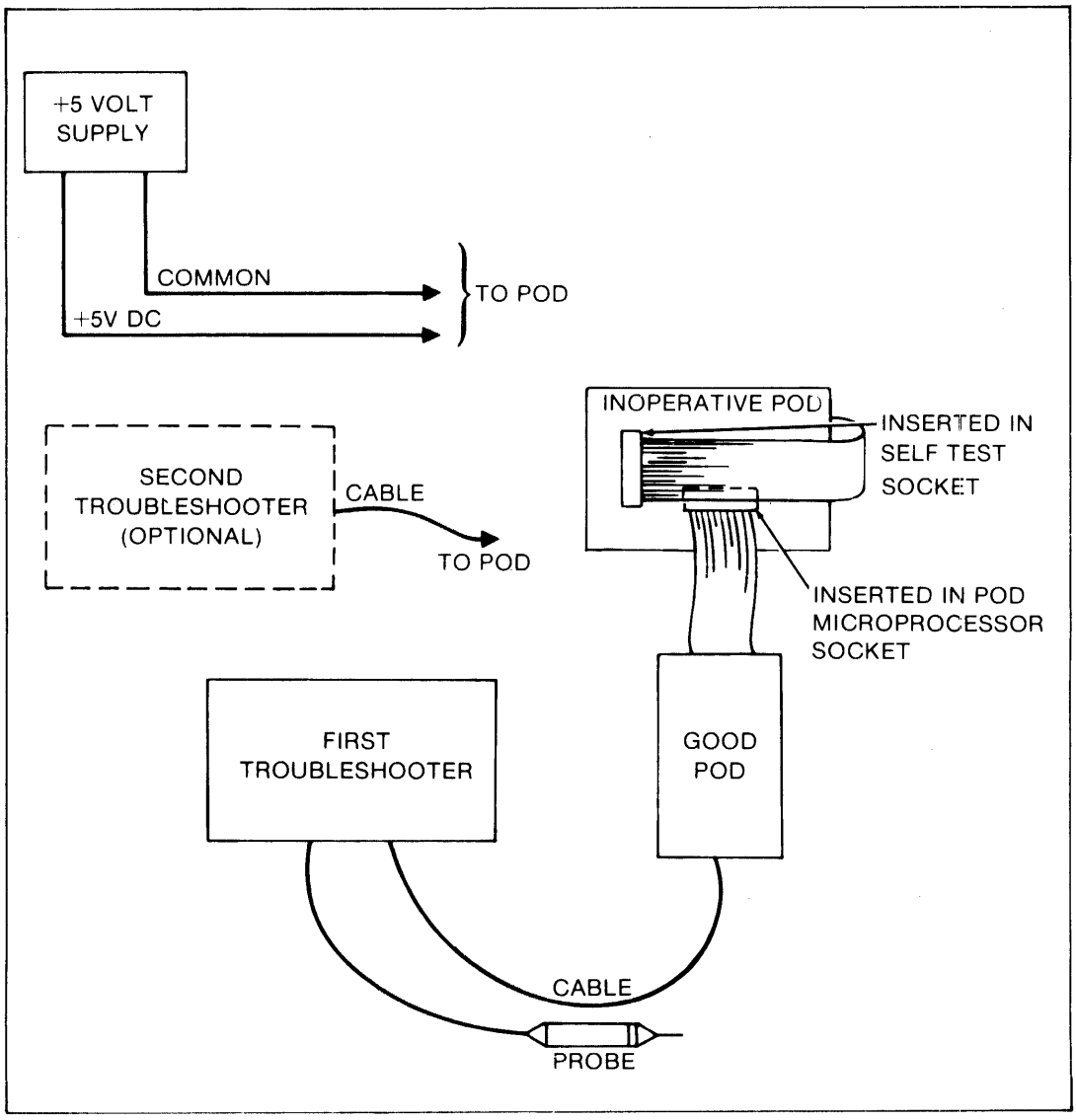


Figure 6-2. Troubleshooting an Inoperative Pod

6-12. Procedure for Troubleshooting an Inoperative Pod

Use the following steps as a guide for troubleshooting an inoperative Pod using a good Pod. The circuits and components mentioned in these steps appear in the schematic diagrams in Section 8, and the circuits are described in the Theory of Operation in Section 5.

NOTE

The following procedures are intended only to direct the technician towards the source of a problem. In each case, once an anomaly is detected, it is up to the technician to pursue the problem further. It is suggested that standard troubleshooting procedures be used, once a problem is identified, to locate the source of the trouble. This involves such things as checking and verifying activity of enabling and timing signals, input and output signals, and logic.

NOTE

When performing looping read or write operations with a synchronized oscilloscope connected to the Troubleshooter, use the Quick-Looping Read or Write feature described in Section 4 to obtain a brighter signal trace on the scope.

1. Prepare the Pod as outlined in the previous section (Preparation for Troubleshooting an Inoperative Pod). Position the inoperative Pod so that the interface board (the one without the microprocessor socket) is upwards. Apply power to the defective Pod. If a second Troubleshooter is used to supply power to the Pod to be tested, press the STOP key on the second Troubleshooter to prevent repetitive Pod resets.

2. Check that the microprocessor clock signal is present at pin 12 of protection circuit A1. (Use the terminal at the lower right corner of the board as a ground reference.)

3. Perform a *READ @ 0* operation with the Troubleshooter (this causes a Pod reset, which then brings initialization information from the Pod to the Troubleshooter), then select the Troubleshooter Setup function and disable all the enableable lines of the good Pod ($\overline{\text{BUSREQ}}$ and $\overline{\text{WAIT}}$).

4. Check the signal levels at U24 pin 1 ($\overline{\text{PFAIL}}$) and U24 pin 2 ($\overline{\text{LOPWR}}$). With proper self test socket (or UUT) voltages, $\overline{\text{PFAIL}}$ will be high and $\overline{\text{LOPWR}}$ will be low.

5. Position the inoperative Pod so that the processor board is facing up. Reset the Pod by momentarily shorting the end of R6 that is closest to the microprocessor to ground. This is the Pod's internal hardware reset line. (A ground test point is located near the left bottom corner of the PCB.)

   Verify that the $\overline{\text{RESET}}$ line goes low at the processor socket by performing a Looping Read Status operation with the Troubleshooter mainframe. While shorting pin 13, the mainframe should display the following status bits:

   XXXX 0101 1110 1111

   After removing the short the status word should be:

   XXXX 0111 1111 1111

   Check the other status bits to ensure that they are at their proper logic levels.

NOTE

Before initially resetting the Pod, the status word may be different from that shown.

6. Check the address decoder (U14) by performing looping read operations using the following address offsets and noting that the corresponding outputs of U14 go low:

Read Address Offset	U14 Output Pin
0000	15
2000	14
4000	13
6000	12
8000	11
A000	10
E000	7

7. Reset the Pod (by grounding the end of R6 that is closest to the microprocessor momentarily), and perform the RAM short and RAM long tests at addresses 2000 through 2FFE.

8. Perform a ROM test on addresses 0000 through 1FFC. Compare the resultant signature with the expected signature, which can be read at location 1FFE (the last word of ROM).

CAUTION

Steps 12 and 13 are for checking the output operation of I/O port A. These steps are intended to be used only if the inoperative Pod is supplied with power from separate power supplies. If the Pod is supplied with power by a second Troubleshooter, I/O port A may be overdriven by the outputs from the Troubleshooter.

Remove the UUT cable from the Pod's self test socket for the following tests.

9. Momentarily connect the lead of R6 that is closest to the microprocessor (U42, J3) to ground. This will reset the Pod and prevent errant timeout counter actions.

CAUTION

The side of R6 that is FARTHEST from the microprocessor is connected to the +5 volt power supply. Inadvertently connecting this side of the resistor to ground might damage the Troubleshooter's power supply.

10. Check the output operation of the Troubleshooter I/O port (U6) as follows:

    a. Do a *WRITE @ 4000 = 8FF* to enable the output port and write all lines high.

    b. Check the output port lines (pins 2,5,6,9,12,15,16,19 of U6) with the probe or an oscilloscope to confirm that all the levels are high.

    c. Repeat Step a with 800 as the data, to set all of the levels low.

    d. Repeat Step b, checking for all logic low levels.

11. Check the input operation of the Troubleshooter I/O port (U7) as follows:

    a. Select the data sync mode and turn-on the low pulse key of the Troubleshooter.

    b. Do the operation *WRITE @ 4000 = 0* to disable the output port.

    c. Do a *READ @ F000 0004* to start a Looping-Read operation at address 4000, the address of the I/O input port register.

    d. Apply the probe to each of the input port lines (pins 2 - 9 of U7). Observe the Troubleshooter display and check that each of the lower eight input bits toggle.

12. Check the operation of internal Pod control port U11 as follows:

    a. Do a *WRITE @ 4000 = FF00.*

    b. Check the output lines of U11 (pins 2,5,7,10,12,15) with the probe or oscilloscope to confirm that all the levels are high.

    c. Do a *WRITE @ 4000 = 0.*

    d. Repeat Step b, checking for all logic low levels.

13. Check the operation of internal Pod control port U37 as follows:

    a. Do a *WRITE @ 6000 = 003F.*

    b. Check the output lines of U37(pins 2,5,7,10,12,15) with the probe or oscilloscope to confirm that all the levels are high.

    c. Do a *WRITE @ 6000 = 0.*

    d. Repeat Step b, checking for all logic low levels.

14. Check the operation of internal Pod status port U8 as follows:

    a. Select the Free-Run Troubleshooter sync. Set the pulse polarity to low. Prepare a jumper wire from board ground (TP1 or TP2).

    b. Do a *READ @ F000 0004* to start a Looping-Read operation at address 6000, the address of the internal self test status register.

c. Hold the probe to pin 11 of U8. Touch the grounded jumper to each of the input lines (pins 3,4,7,8,13,14,17,18 of U8) as you observe the Troubleshooter display and check that each of the upper eight input bits toggles as it is probed.

15. Check the operation of internal Pod status port U8 as follows:

a. Select the Free-Run Troubleshooter sync. Set the pulse polarity to low. Prepare a jumper wire from board ground (TP1 or TP2).

b. Do a *READ @ F000 0004* to start a Looping-Read operation at address 6000, the address of the internal self test status register. Hold the probe to pin 11 of U8. Touch the grounded jumper to each of the input lines (pins 3,4,7,8,13,14,17,18 of U8) as you observe the Troubleshooter display and check that each of the eight input bits toggle.

c. Check the output lines (pins 2,5,7,10,12,15) with the probe or oscilloscope to confirm that all the levels are high.

d. Do a *WRITE @ 4000 = 0.*

e. Repeat Step b, checking for all logic low levels.

16. Check the operation of internal Pod control port U9 as follows:

a. Do a *WRITE @ 6000 = 3F00.*

b. Check the output lines of U9 (pins 2,5,7,10,12,15) with the probe or oscilloscope to confirm that all the levels are high.

c. Reset the Pod by momentarily shorting line 1 of U11 to ground.

d. Do a *WRITE @ 6000 = 0.*

e. Repeat Step b, checking for all logic low levels.

17. Check the operation of the Pod UUTON timer U10 as follows:

a. Set the Troubleshooter probe sync to Free-Run, then check the ripple out line (pin 13) with the probe or an oscilloscope to verify that the line is stable in a high state.

b. Do a *WRITE @ 6000 = 1000.*

c. Verify that the ripple out line (pin 13) is now toggling (and at a period of approximately 12 μ sec if using an oscilloscope).

18. Short the lead of R6 that is closest to the microprocessor to ground to reset the Pod.

6-13. EXTENDED TROUBLESHOOTING PROCEDURES

The troubleshooting procedures provided in this section supplement the circuit checks performed on the Pod during the Pod self test; these procedures are appropriate for use with a Pod that passes the Pod self test but does not appear to function normally when used with a Troubleshooter and a good UUT. If a Pod fails the self test, it would be better to begin troubleshooting with the procedure provided in the previous section titled Troubleshooting a Defective Pod.

6-14. Cable Lines

The self test checks every line in the cable to ensure that it may be driven both high and low (except for the power supply lines). If the Pod passes self test, but the Troubleshooter displays the message *POD TIMEOUT* when the Pod is connected to a UUT, check the clock output of the UUT.

6-15. Pod Enable Lines

The circuitry for enabling the various forcing lines is checked by the normal Pod self test (corresponding to the failure code 03). During this test routine, all the forcing lines are simultaneously enabled. If the enableable line circuitry seems to be functioning improperly when the Pod is connected to a UUT, try selectively enabling the lines with the Pod cable inserted into the self test socket, but with the Pod self test disabled. (The Pod self test is disabled by writing the special address location F000 0024 = non zero). If a line is found that does not cause the Pod to timeout, then the logic controlling the enabling of that line should be examined.

6-16. Timing Problems

These problems are usually caused by components that are still functioning, but are not functioning within the allowable specifications. The best way to check this problem is to look at suspected signals using an oscilloscope synchronized to valid addresses. Look for slow rise or fall times or signals driven to marginal logic levels. If the part is too slow, it might fail in the UUT, but pass the Pod self test because the Pod clock rate is somewhat slower. The clock rate at the self test socket is approximately 6 MHz.

6-17. Noise Problems

If a part has marginal drive capabilities, the added noise of a UUT environment might cause it to fail. Be sure to note that inputs as well as outputs can malfunction (they may exhibit excessive leakage) and put too much load on an output causing either low levels, slow transition times, or both.

6-18. DISASSEMBLY

To gain access to the two PCB assemblies in the Pod, perform the following steps:

1. Remove the Pod ribbon cable plug from the self test socket.

2. Turn the Pod over on its top (with the large Pod decal facing up). Remove the four Phillips screws that hold the case halves together and remove the top and bottom case halves. Place the PCB assemblies so that the self test socket (on the processor PCB assembly) is facing up.

3. On the corner opposite the self test socket thumbwheel, remove the single Phillips screw that retains the shield surrounding the PCB assemblies. (A washer will come off with the screw.) Remove the shield.

NOTE

When the shield is removed, all the components are exposed. It should not be necessary to separate the two PCB assemblies while troubleshooting except to replace components.

4. To separate the two PCB assemblies, turn the PCB assemblies over so that the self test socket is facing down. Remove the four Phillips screws at the corners of the PCB assemblies and carefully pull the boards apart at the two connectors along the sides.

Section 7
List of Replaceable Parts

7-1. INTRODUCTION

This section contains an illustrated parts list for the instrument. Components are listed alphanumerically by assembly.

Parts lists include the following information:

1. Reference Designation.
2. Description of Each Part.
3. Fluke Stock Number.
4. Federal Supply Code for Manufacturers (see the 9000 Series Troubleshooter Service Manual for Code-to-Name list).
5. Manufacturer's Part Number.
6. Total Quantity of Components Per Assembly.
7. Recommended quantity: This entry indicates the recommended number of spare parts necessary to support one to five instruments for a period of two years. This list presumes an availability of common electronic parts at the maintenance site. For maintenance for one year or more at an isolated site, it is recommended that at least one of each assembly in the instrument be stocked.

7-2. HOW TO OBTAIN PARTS

Components may be ordered directly from the manufacturer by using the manufacturer's part number, or from the John Fluke Mfg. Co., Inc. or an authorized representative by using the Fluke Stock Number.

In the event the part ordered has been replaced by a new or improved part, the replacement will be accompanied by an explanatory note and installation instructions if necessary.

To ensure prompt and efficient handling of your order, include the following information.

1. Quantity.
2. Fluke Stock Number.

3. Description.

4. Reference Designation.

5. Printed Circuit Board Part Number and Revision Letter.

6. Instrument Model and Serial Number.

A Recommended Spare Parts Kit for your basic instrument is available from the factory. This kit contains those items listed in the REC QTY column for the parts lists in the quantities recommended.

Parts price information is available from the John Fluke Mfg. Co., Inc. or its representative. Prices are also available in a Fluke Replacement Parts Catalog, which is available upon request.

CAUTION

Indicated devices are subject to damage by static discharge.

7-3. MANUAL CHANGE AND BACKDATING INFORMATION

Table 7-4 contains information necessary to backdate the manual to conform with earlier PCB configurations. To identify the configuration of the PCB's used in your instrument, refer to the revision letter on the component side of each PCB assembly.

As changes and improvements are made to the instrument, they are identified by incrementing the revision letter marked on the affected PCB assembly. These changes are documented on a supplemental change/errata sheet which, when applicable, is inserted at the front of the manual.

To backdate this manual to conform with an earlier assembly revision level, perform the changes indicated in Table 7-4. There are no backdating changes at this printing. All PCB assemblies are documented at their original revision level.

Table 7-1. Z8000 Final Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
	FINAL ASSEMBLY FIGURE 7-1 (9000A-Z8000-5001)						
A33	Ⓟ PROCESSOR PCB				1		
A34	Ⓟ INTERFACE PCB				1		
H1	SCREW, SEMS, 4-40 X 1/4	185918	89536	185918	6		
H2	SCREW, PHP, 4-40 X 3/4	115063	89536	115063	4		
H3	WASHER, INTLK #4	110403	89536	110403	1		
H4	SCREW, PHP, 4-40 X 5/8	145813	89536	145813	1		
MP1	LABEL, "Static Caution"	605808	89536	605808	1		
MP2	DECAL, "Warning"	659813	89536	659813	1		
MP3	SHIELD, INNER ALUM/MYLAR	659771	89536	659771	1		
MP4	SPACER, HEX ALUM., 4-40 X 0.375	187575	89536	187575	1		
MP5	SHELL, BOTTOM	648881	89536	648881	1		
MP6	ACTUATOR BUTTON	582916	89536	582916	1		
MP7	SHELL, TOP	607648	89536	607648	1		
MP8	COVER, SLIDE	728758	89536	728758	1		
MP9	DECAL, "Switching"	707356	89536	707356	1		
MP10	DECAL, "Pod"	640631	89536	640631	1		
MP11	DECAL, "Spec."	650366	89536	650366	1		
MP12	ACCESSORY KIT (COMPLETE KIT)	609206	89536	609206	1		
	ADAPTER ASSY., 40-48 PIN	728774	89536	728774	1		
MP13	ADAPTER ASSY., 48-40 PIN	728766	89536	728766	1		
TM1	TECHNICAL MANUAL, 9000A-Z8000	716035	89536	716035	1		
U17	IC, PROGRAMMED (V1.0) (9000A-Z8000-99100)	661496	89536	661496	1	1	1
U20	IC, PROGRAMMED (V1.0) (9000A-Z8000-99101)	715953	89536	715953	1	1	1
U42	Ⓟ IC, NMOS Z80001A, 16 BIT MICROPROCESSOR	723098	89536	723098	1		1
	CABLE, POD	607184	89536	607184	1		
	CABLE, UUT, 48-POSITION, SHIELDED	650432	89536	650432	1		

1 USED ON PROCESSOR PCB A33 ASSY.

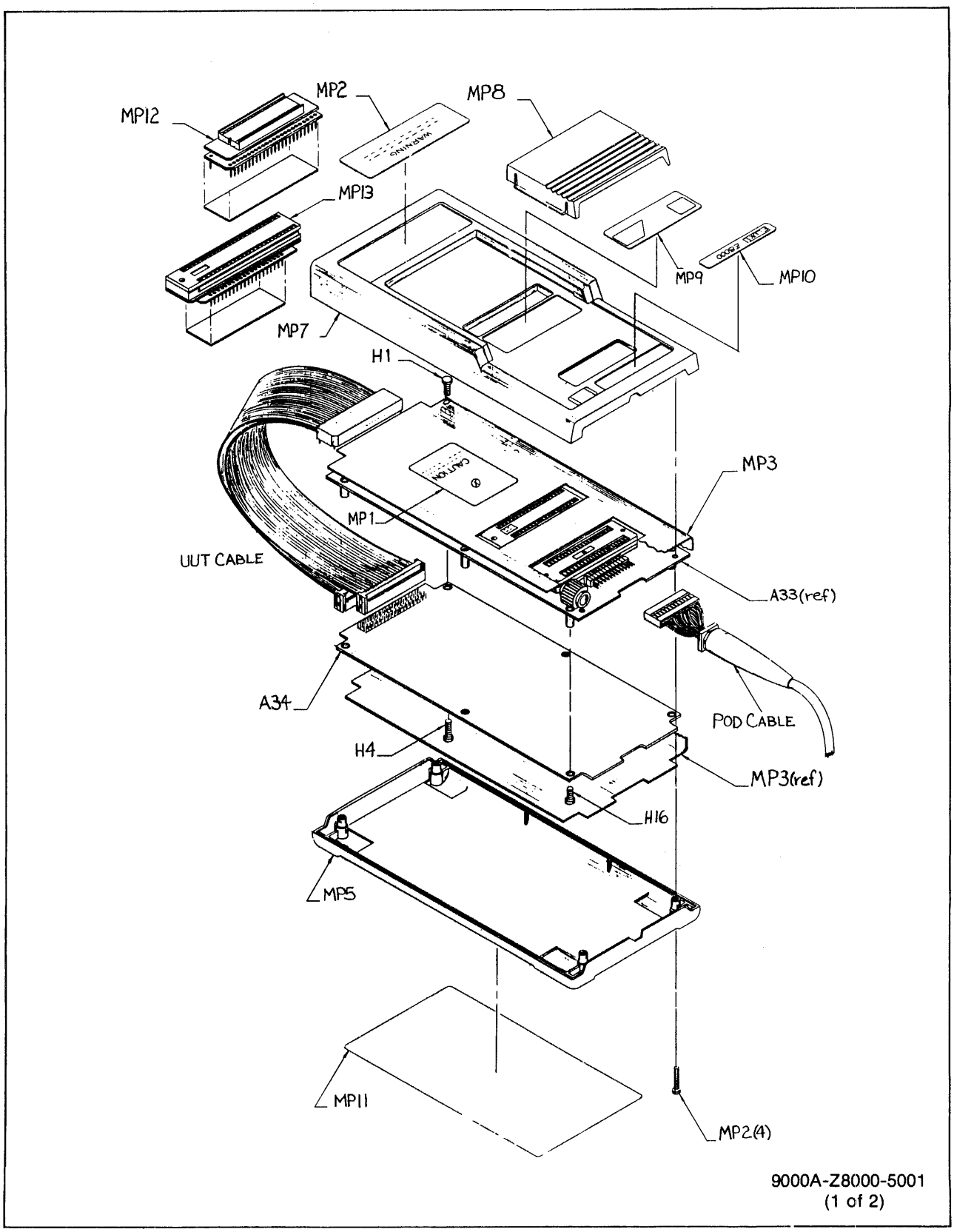
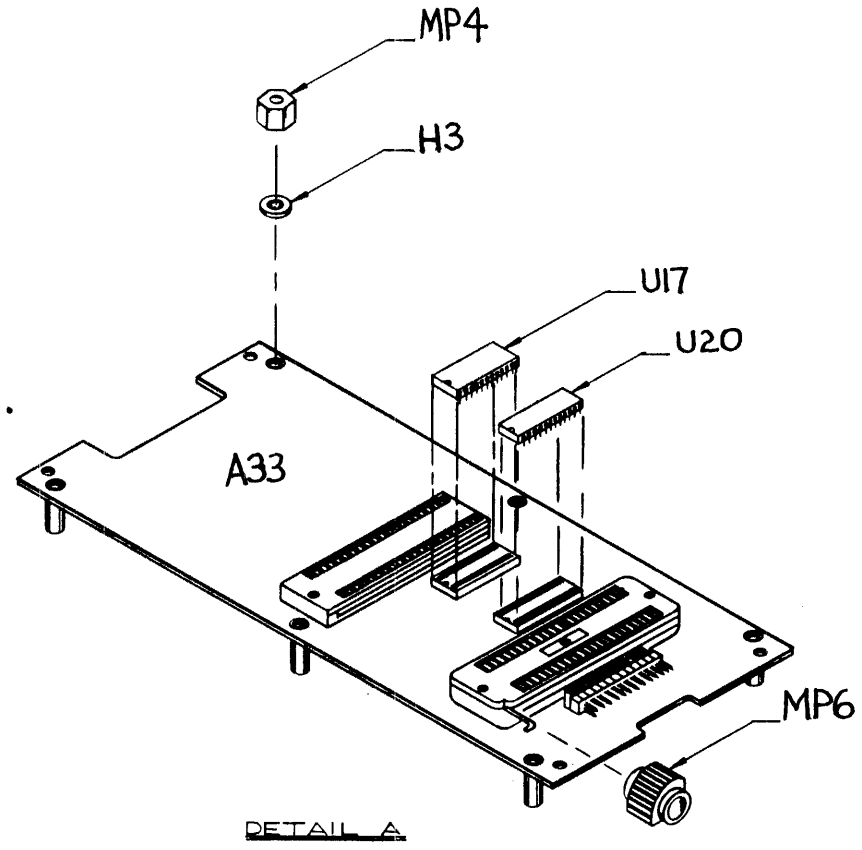


Figure 7-1. 9000A-Z8000 Final Assembly



CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

9000A-Z8000-5001
(2 of 2)

Figure 7-1. 9000A-Z8000 Final Assembly (cont)

Table 7-2. A33 Processor PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A33	Ⓐ PROCESSOR PCB ASSEMBLY FIGURE 7-2 (9000A-Z8000-4071)				REF		
C1	CAP, CER, 18 PF +/-2%, 100V	512335	89536	512335	2		
C2	CAP, CER, 18 PF +/-2%, 100V	512335	89536	512335	REF		
C3-C14	CAP, CER, 0.22 UF +/-20%, 50V	309849	71590	CZ30C224M	12		
C15	CAP, CER, 47 PF +/-2%, 100V	512368	89536	512368	1		
CR1	DIODE, HI SPEED SWITCHING	203323	07910	1N4448	1		
J1	CONNECTOR, RIGHT ANGLE, 26-PIN	512590	89536	512590	1		
J2	SOCKET, ZIP 48-PIN	707109	89536	707109	1		
J3	SOCKET, DIP 48-PIN	714220	89536	714220	1		
MP2	HOLDER, COMPONENT	422865	98159	2829-76-2	1		
P1,P2	CONNECTOR PIN	267500	00779	87022-1	100		
R1	RES, DEP. CAR, 33 =/-5%, 1/4W	414524	80031	CR251-4-5P33E	1		
R2	RES, DEP. CAR, 100 +/-5%, 1/4W	348771	80031	CR251-4-5P100E	1		
R3	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	1		
R4	RES, COMP, 47 +/-5%, 1/2W	159608	01121	EB4705	1		
R5	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	4		
R6	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R7	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R8	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R9	RES, DEP. CAR, 68 =/-5%, 1/4W	414532	80031	CR251-4-5P68E	1		
S1	SWITCH, SLIDE, SPDT	477984	34328	GS115G20-52	1		
TP1,2, TP4,5	TERMINAL, TEST POINT	512889	02660	62395	4		
U1	Ⓐ IC, FTTL, QUAD 2 INPUT- AND GATE	634444	07263	74F04PC	2	1	
U2	IC, LSTTL, QUAD BUS BUF W/3 STATE OUT	472746	01295	SL74LS125N	1	1	
U3	Ⓐ IC, FTTL, QUAD 2 INPUT- AND GATE	634444	07263	74F04PC	REF		
U4	IC, LSTTL, OCTAL D TRANSPARENT LATCHES	504514	01295	SN74LS373N	4	1	
U5	IC, LSTTL, OCTAL D TRANSPARENT LATCHES	504514	01295	SN74LS373N	REF		
U6	IC, TTL, OCTAL D F/F +EDG TRIGGERED	473223	01295	SL74LS374N	2	1	
U7	IC, LSTTL, OCTAL BUFFER/LINE DRIVER	634105	01295	SN74LS541N	1	1	
U8	IC, TTL, OCTAL D F/F +EDG TRIGGERED	473223	01295	SN74LS374N	REF		
U9	IC, LSTTL, HEX D F/F +EDG TRG W/CLEAR	393207	01295	SL74LS174N	3	1	
U10	IC, LSTTL, PRESET 4 BIT UP/DWN COUNTER	707299	01295	SL74LS191N	1	1	
U11	IC, LSTTL, HEX D F/F +EDG TRG W/CLEAR	393207	01295	SL74LS174N	REF		
U12	IC, LSTTL, OCTAL BUS TRNSCVR W/3ST OUT	477406	01295	SL74LS245N	2	1	
U13	IC, LSTTL, OCTAL BUS TRNSCVR W/3ST OUT	477406	01295	SL74LS245N	REF		
U14	Ⓐ IC, FTTL, 1 OF 8 DECODER/DEMULTIPLXR	707281	07263	74F138PC	1	1	
U15	Ⓐ IC, CMOS, 2K X 8 STAT RAM	707265	89536	707265	2	1	
U16	Ⓐ IC, CMOS, 2K X 8 STAT RAM	707265	89536	707265	REF		
U17	IC, PROGRAMMED (99100)		SEE	TABLE 7-2			1
U18	IC, LSTTL, OCTAL D TRANSPARENT LATCHES	504514	01295	SN74LS373N	REF		
U19	IC, LSTTL, OCTAL D TRANSPARENT LATCHES	504514	01295	SN74LS373N	REF		
U20	IC, PROGRAMMED (99101)		SEE	TABLE 7-2			1
U21	Ⓐ IC, FTTL, DUAL D F/F+EDG TRG W/CL&SET	659508	07263	74F74PC	3	1	
U22	Ⓐ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	7	2	
U23	Ⓐ IC, FTTL, DUAL D F/F+EDG TRG W/CL&SET	659508	07263	74F74PC	REF		
U24	Ⓐ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	REF		
U25	Ⓐ IC, FTTL, QUAD 2-INPUT OR GATE	659904	97263	74F32PC	REF		

Table 7-2. A33 Processor PCB Assembly (cont)

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
U26	Ⓟ IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	3	1	
U27	Ⓟ IC, FTTL, DUAL D F/F+EDG TRG W/CL&SET	659508	07263	74F74PC	REF		
U28	Ⓟ IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	REF		
U29	Ⓟ IC, FTTL, QUAD 2-INPUT AND GATE	650523	07263	7408PC	2	1	
U30	Ⓟ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	REF		
U31	Ⓟ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	REF		
U32	Ⓟ IC, LSTTL, TRIPLE 3-INPUT NOR GATE	393090	01295	SN74LS27N	2	1	
U33	Ⓟ IC, CMOS, QUAD 2-INPUT NAND GATE	707323	12040	MM74HC00N	1	1	
U34	IC, LSTTL, DUAL DIV BY 16 BINARY COUNTER	483578	01295	SN74LS393	1	1	
U35	Ⓟ IC, FTTL, QUAD 2-INPUT AND GATE	650523	07263	74F08PC	REF		
U36	Ⓟ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	REF		
U37	IC, LSTTL, HEX D F/F +EDG TRG W/CLEAR	393207	01295	74LS174N	REF		
U38	Ⓟ IC, LSTTL, TRIPLE 3-INPUT NOR GATE	393090	01295	SN74LS27N	REF		
U39	Ⓟ IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	REF		
U40	Ⓟ IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	REF		
U41	Ⓟ IC, CMOS, CLOCK GENERATOR	698175	32293	ICM72091PA	1	1	
U42	SEE FINAL ASSEMBLY, U42						
XU15	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	4		
XU16	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	REF		
XU17	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	REF		
XU20	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	REF		
XU33	SOCKET, IC, 14-PIN DIP	276527	09922	DILB8P-108	1		
XU41	SOCKET, IC, DIP 8-PIN	478016	91506	308-AG39D	1		
Y1	Ⓟ CRYSTAL, 6 MHz+/-0.015%	461665	89536	461665	1	1	

1 ALSO SEE FIGURE 7-1, DETAIL A

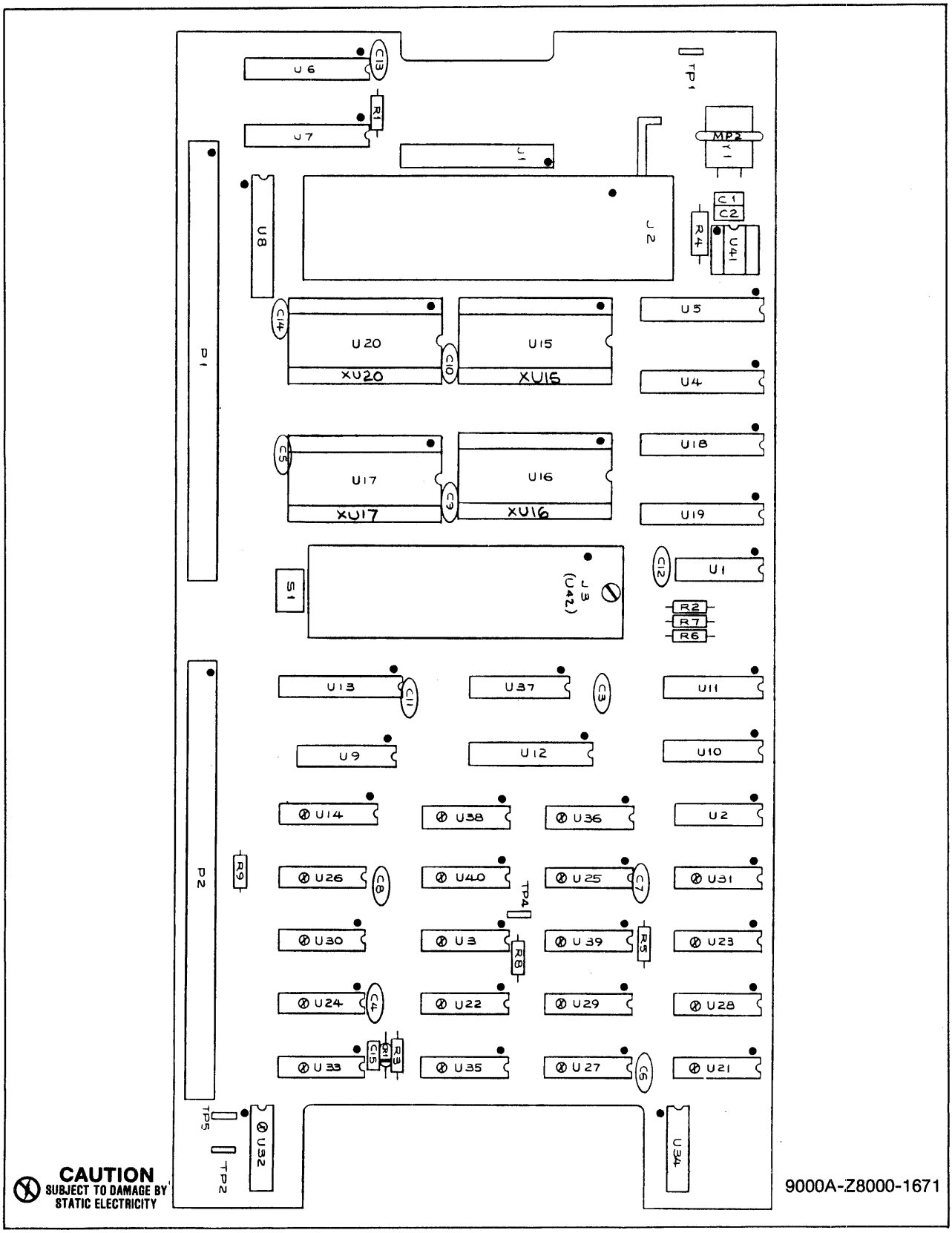


Figure 7-2. A33 Processor PCB Assembly

Table 7-3. A34 Interface PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A34	⊗INTERFACE PCB ASSEMBLY FIGURE 7-3 (9000A-Z8000-4072)				REF		
A1-A6 A1-A6	HYBRID, PROTECTION 700 TESTED	582189	89536	582189	6		
C1,C2	CAP, CER, 82PF, +/-2%, 100V, COG	512350	89536	512350	2		
C3-C14	CAP, CER, 0.22UF +/-20%, 50V, Z5U	309849	71590	CW3C0C224K	12		
J1,J2	CONNECTOR, PCB MOUNT, 50-POSITION	649848	00779	86396-5	2		
MP1	SPACER, NYLON (W/VR1)	175125	89536	175125	1		
Q1,Q2	TRANSISTOR, SI, NPN, SMALL SIGNAL	333898	89536	333898	1	1	
R1	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	1		
R2, R3	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	3		
R4	RES, DEP. CAR, 1M +/-5%, 1/4W	348987	80031	CR251-4-5P1M	1		
R5	RES, COMP, 4.7M +/-5%, 1/4W	220046	01121	CB4755	2		
R6,R7	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	4		
R8	RES, MTL. FILM, 2.49K +/-1%, 1/8W	226209	91637	CMF552491F	1		
R9	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R10-R13	RES, DEP. CAR, 680 +/-5%, 1/4W	368779	80031	CR251-4-5P680E	4		
R14	RES, MTL. FILM, 1K +/-1%, 1/8W	168229	91637	CMF551001F	1		
R15	RES, DEP. CAR, 33 +/-5%, 1/4W	414524	80031	CR251-4-5P33E	1		
R16	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	REF		
R17	RES, COMP, 4.7M +/-5%, 1/4W	220046	01121	CB4755	REF		
R18	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	REF		
TP1-TP4	TERMINAL, TEST POINT	512889	02660	62395	40		
U1,U2	⊗IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	2	1	
U3	IC,LSTTL, QUAD BUS BFR W/3-STATE OUT	472746	01295	SN74S125N	1	1	
U4	⊗IC, FTTL, HEX INVERTER	634444	07263	74F04PC	1	1	
U5	⊗IC, FTTL, DUAL D F/F +EDG TRG W/CL&SET	659508	07263	74F74PC	1	1	
U6,U7	⊗IC, TTL, OCTAL D F/F +EDG TRIG	473223	01295	SN74LS374N	2	1	
U8,U9	IC, ALSTTL, OCTAL BUS XCVR W/3-STATE	647214	01295	SN74ALS245N	2	1	
U10,U11	IC, FTTL, QUAD 2-INPUT MULTIPLEXER	647156	01295	SN74F257N	4	1	
U12	⊗IC, CMOS, OCTAL DUAL F/F +EDG TRIG 3-ST	707695	01295	SN74SC374N	8	2	
U13,U14	IC, FTTL, QUAD 2-INPUT MULTIPLEXER	647156	01295	SN74F257N	REF		
U15-U21	⊗IC, CMOS, OCTAL DUAL F/F +EDG TRIG 3-ST	707695	01295	SN74SC374N	REF		
U22,U23	IC, LSTTL, HEX D F/F +EDG TRG W/CLEAR	393207	01295	SN74LS174N	2	1	
U24	IC, COMPARATOR, QUAD, 14-PIN DIP	387233	12040	LM339N	1	1	
U25	⊗IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	1	1	
VR1	VOLT REFERENCE, 1.22V BAND GAP	452771	89536	452771	1	1	
XU6-XU9	SOCKET, IC, 20-PIN, DIL	454421	09922	DILB20P-108	12		
XU12	SOCKET, IC, 20-PIN, DIL	454421	09922	DILB20P-108	REF		
XU15-21	SOCKET, IC, 20-PIN, DIL	454421	09922	DILB20P-108	REF		
Z1	RESISTOR NETWORK, THICK FILM	583476	89536	583476	1	1	

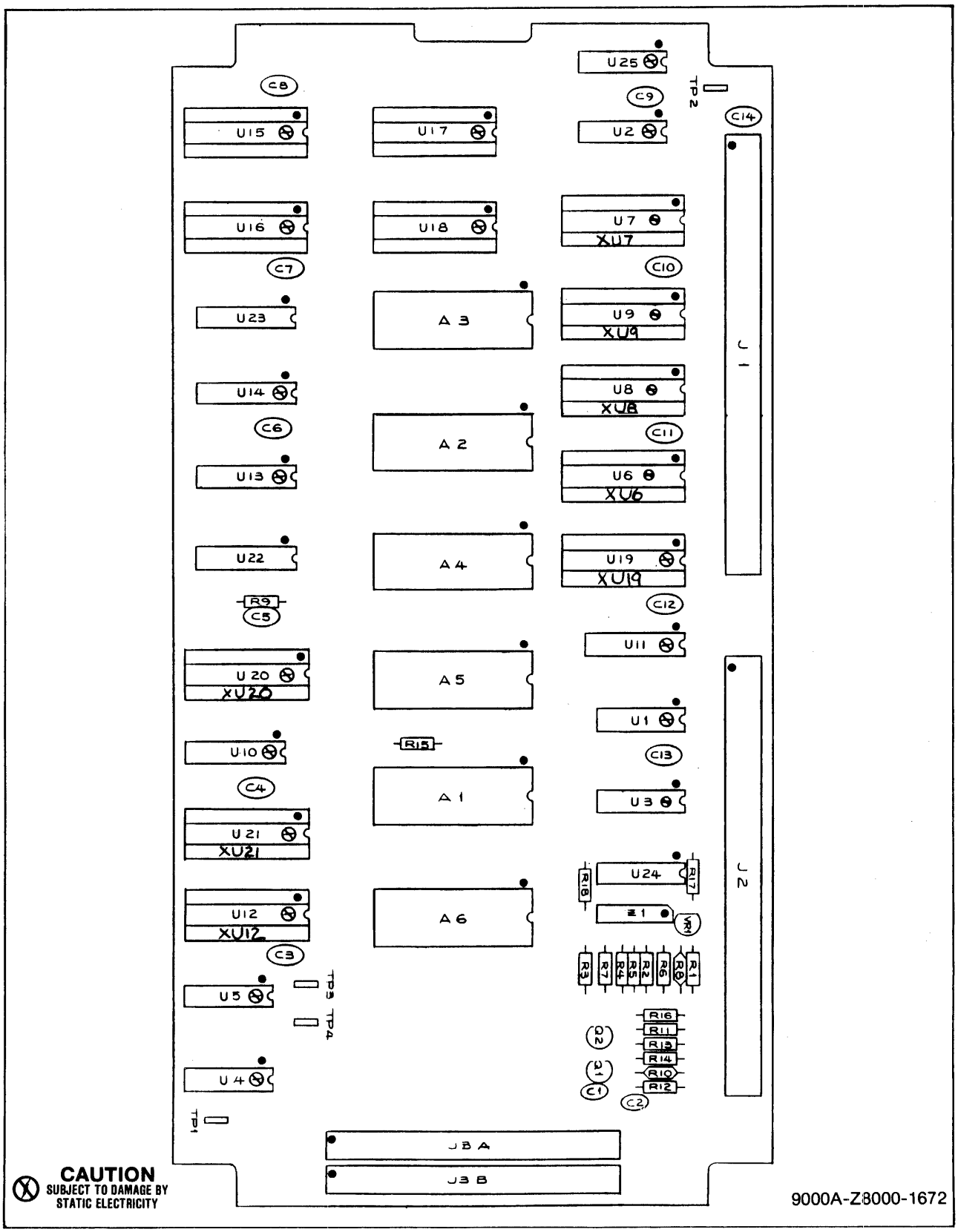


Figure 7-3. A34 Interface PCB Assembly

Table 7-4. Manual Status and Backdating Information

Ref Or Option No.	Assembly Name	Fluke Part No.	* To adapt manual to earlier rev configurations perform changes in desending order (by no.), ending with change under desired rev letter																					
			–	A	B	C	D	E	F	G	H	J	K	L	M	N	P							
A33	Processor PCB	581025	X																					
A34	Interface PCB	581231	X																					

* X = The PCB revision levels documented in this manual.
● = These revision letters were never used in the instrument.
–= No revision letter on the PCB.

Section 8
Schematic Diagrams

TABLE OF CONTENTS

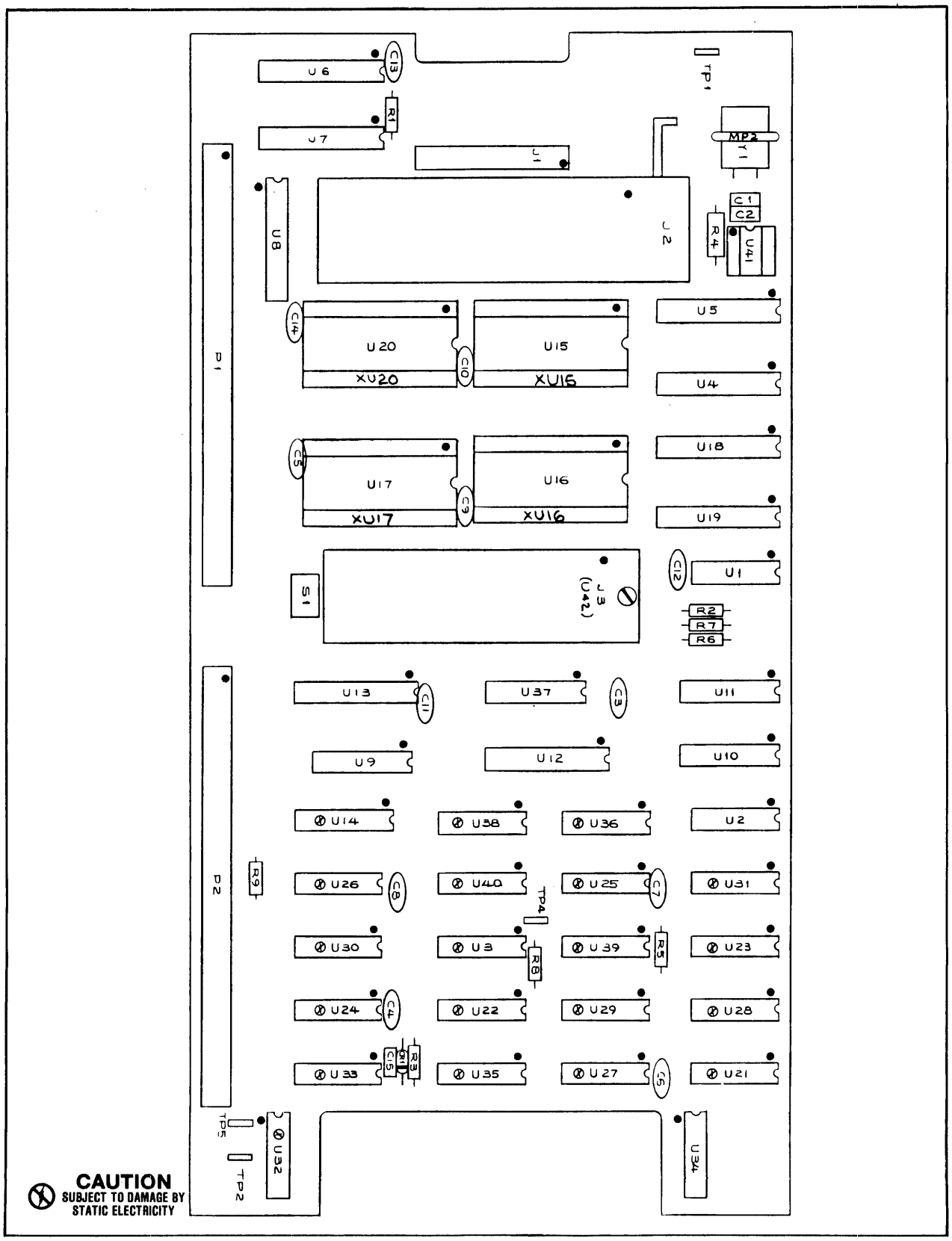
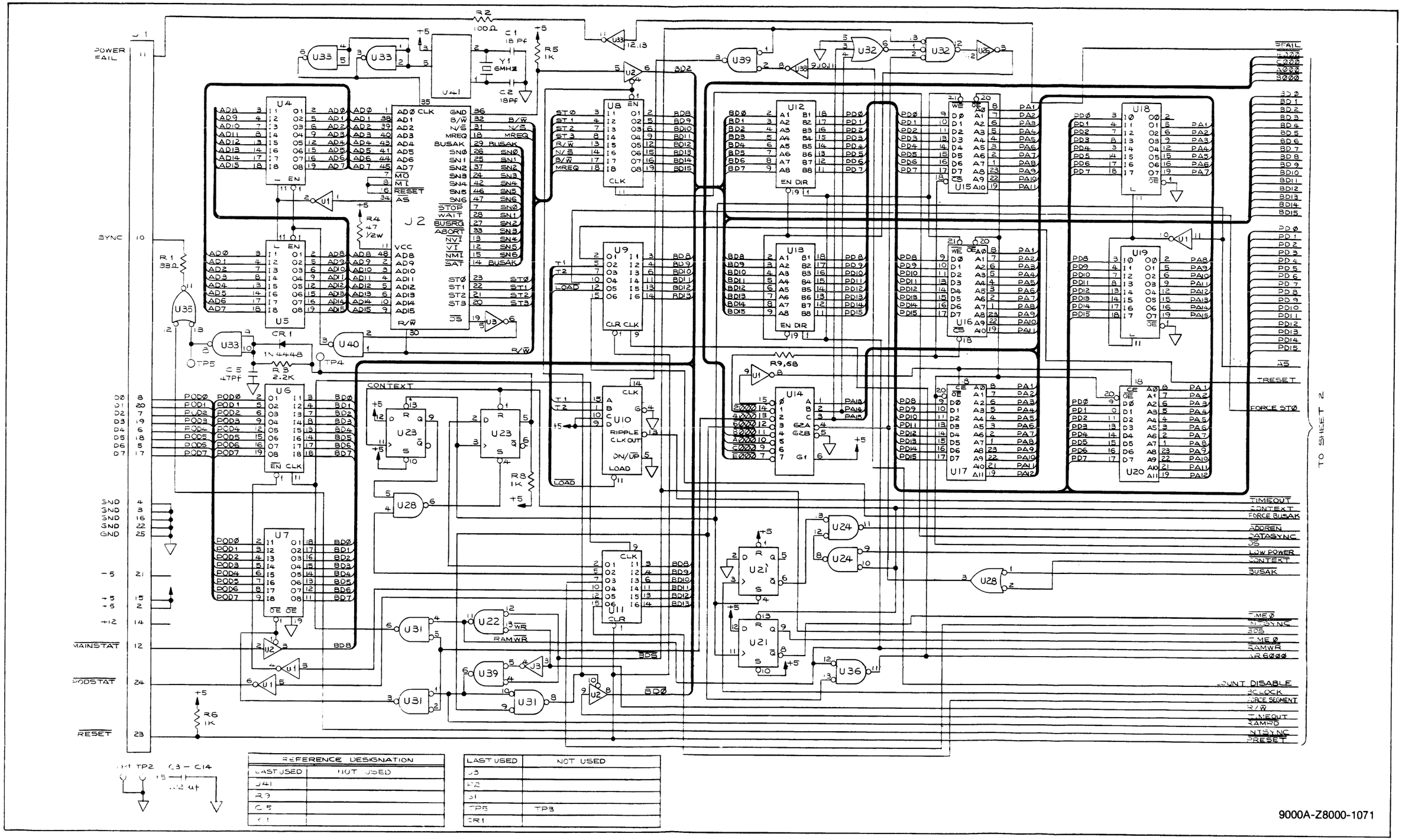


Figure 8-1. A33 Processor PCB Assembly

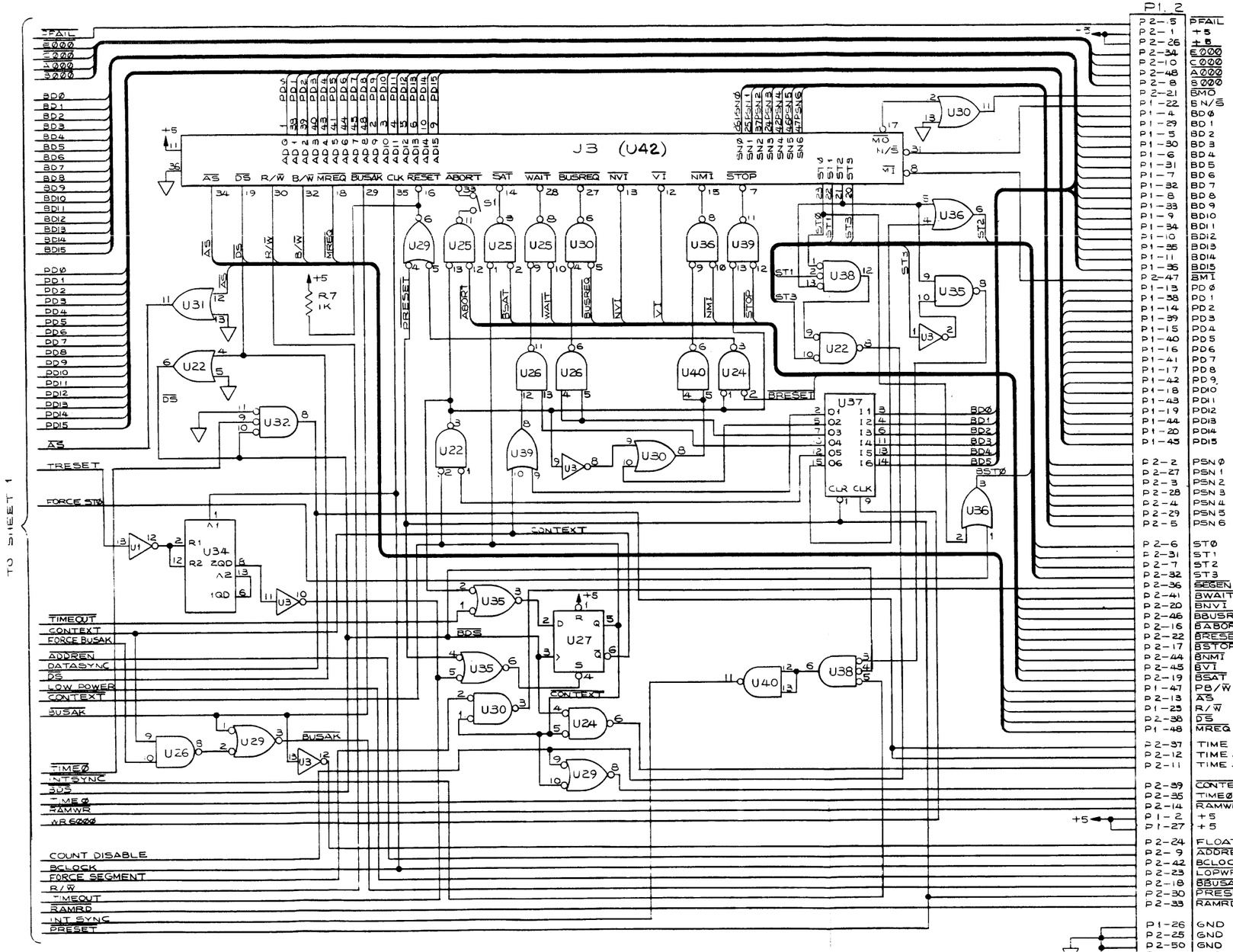


REFERENCE DESIGNATION	
LAST USED	NOT USED
U41	
R9	
C5	
CR1	

LAST USED	NOT USED
U3	
R2	
J1	
TP5	TP3
CR1	

9000A-Z8000-1071

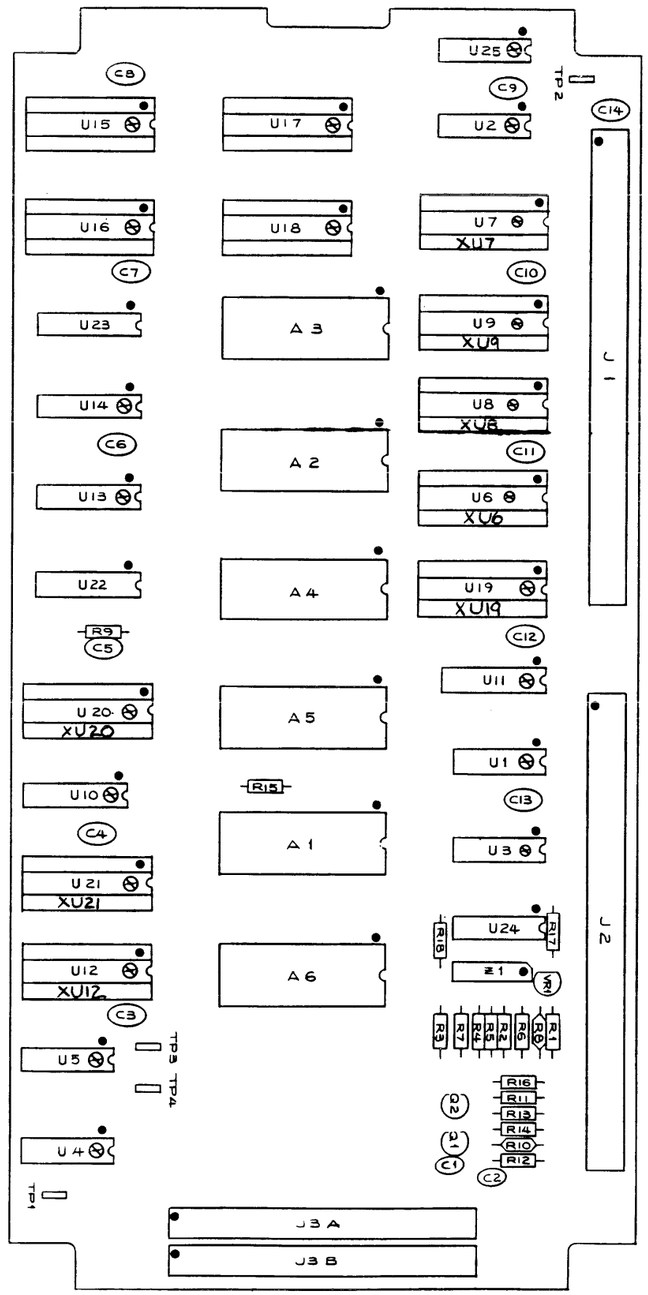
Figure 8-1. A33 Processor PCB Assembly (cont)



DES	DEVICE	+5	GND	QTY
U 1, 3	74F04	14	7	2
U 2	74LS125	14	7	1
U 4, 5, 18, 19	74LS373	20	10	4
U 6, 8	74LS374	20	10	2
U 7	74LS541	20	10	1
U 10	74LS, 91	16	8	1
U 11, 9, 37	74LS174	16	8	3
U 12, 13	74LS245	20	10	2
U 14	74F138	16	8	1
U 15, 16	6116P	24	12	2
U 17, 20	MB7142E	24	12	2
U 21, 23, 27	74F74	14	7	3
U22, 24, 25, 30, 31, 36, 39	74F32	14	7	7
U 26, 40, 28	74F00	14	7	3
U 29, 35	74F08	14	7	2
U 32, 38	74LS27	14	7	2
U 33	74HC00	14	7	1
U 34	74LS393	14	7	1
U 41	ICM 7209	4	8	1
J42	P28000A	11	36	1

9000A-Z8000-1071

Figure 8-1. A33 Processor PCB Assembly (cont)



CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

9000A-Z8000

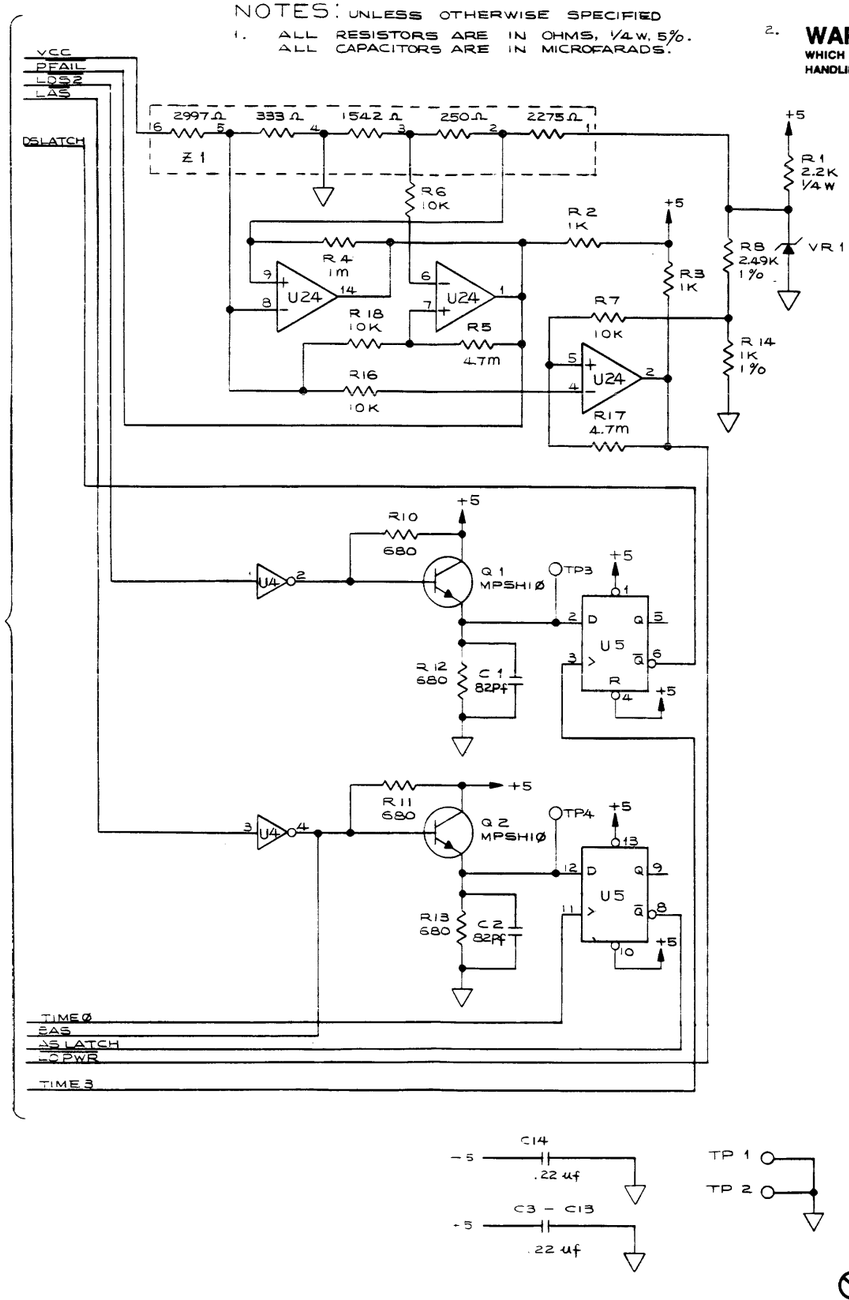
Figure 8-2. A34 Interface PCB Assembly

9000A-Z8000-1072

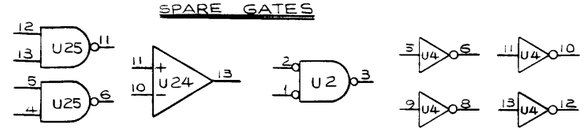
Figure 8-2. A34 Interface PCB Assembly (cont)

NOTES: UNLESS OTHERWISE SPECIFIED

1. ALL RESISTORS ARE IN OHMS, 1/4 W, 5%.
   ALL CAPACITORS ARE IN MICROFARADS.
2. **WARNING:** Ⓝ INDICATES USAGE OF MOS DEVICE(S) WHICH MAY BE DAMAGED BY STATIC DISCHARGE. USE SPECIAL HANDLING PER S.O.P. 12.1



SPARE GATES



DES	DEVICE	+5	GND	+12V	PVCC	QTY	
U1, 2	74F32	14	7			2	Ⓝ
U3	74LS125	14	7			1	
U4	74F04	14	7			1	Ⓝ
U5	74F74	14	7			1	Ⓝ
U6, 7	74LS374	20	10			2	
U8, 9	74ALS245	20	10			2	
U10, 11, 13, 14	74F257	16	8			4	Ⓝ
U12, 15, 16, 17, 18, 19, 20, 21	74SC374	20	10			8	Ⓝ
U22, 23	74LS174	16	8			2	
U24	LM339	3	12			1	
A1 - A6	9000A-4H02T	4	23			6	
U25	74F00	14	7			1	

REFERENCE DESIGNATIORS	
LAST USED	NOT USED
U 25	
R 18	
TP 4	
Q 2	
C14	
VR 1	
J3	
Z1	

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

9000A-Z8000-1072

Figure 8-2. A34 Interface PCB Assembly (cont)

INDEX

Appendix A
Utility Program for Quick Functions

INTRODUCTION

The program contained in this Appendix will make the use of the Z8000 Pod's three Quick Functions, the Quick-Looping Read and Write, the Quick RAM Test, and the Quick ROM Test, appear to operate more like the tests that are built into the Troubleshooter. When using this program, the Operator can use the Quick Functions by entering parameters in response to display prompts, just like to normal built-in tests, rather than by writing information to several special addresses.

The program is provided here in two forms. The first is a standard program, which may be used on any Troubleshooter. It is entered line-by-line as shown, then saved on magnetic tape. Refer to the 9010A Programmers Manual if you need help. The second form is a source program for the 9010A Language Compiler. The Language Compiler is a program that is available to run on several different microcomputers and instrument controllers. The Compiler program will convert the source file into a form that is executable on the Troubleshooter. See 9010A Language Compiler below for more information.

Once the program is available on tape, regardless of which form it originated from, it needs to be loaded into the Troubleshooter and read into memory. Consult the 9010A Operator's Manual for information about using stored programs. Once the program is in memory, it will begin looping through a display sequence which prompts the operator to select from the three available Quick Functions, then prompts for address information to use with the selected test.

9010A LANGUAGE COMPILER

The 9010A Language Compiler is a program which converts source files which are created and edited on a microcomputer into programs for the Troubleshooter. It is available for several common microcomputers, including the Fluke 1720A and 1722A Instrument Controllers, computers with the CP/M operating system, and the IBM Personal Computer. Contact Fluke Customer Service about the 9010A Language Compiler.

Standard Troubleshooter Program

```
1  PROGRAM 0
2   DPY FAST Z8K OPERATIONS
3   EXECUTE 5
4  LABEL 0                          ! SCROLL THROUGH OPTIONS
5   DPY FAST RAM TEST <Y-N>?A
6   IF REGA = 0 GOTO 1
7   EXECUTE 1                        ! DO THE FAST RAM TEST
8  LABEL 1
```

Standard Troubleshooter Program (cont)

```
 9   DPY FAST ROM TEST <Y-N>?A
10   IF REGA = 0 GOTO 2
11   EXECUTE 3                          ! DO THE FAST ROM TEST
12  LABEL 2
13   DPY FAST LOOP <Y-N>?A
14   IF REGA = 0 GOTO 0
15   EXECUTE 4                          ! DO THE FAST LOOP OPERATION
16   GOTO 0
17
18
19  PROGRAM 1
20   DPY WORD INCREMENT <Y-N>?A
21   IF REGA = 1 GOTO 0
22  LABEL 1
23   WRITE @ F0000006 = 1               ! SET BYTE/WORD BIT
24   REG8 = 2000                        ! BYTE TYPE IN HIGH ADDRESS
25   GOTO 2
26  LABEL 0
27   WRITE @ F0000006 = 2
28   REG8 = 0                           ! RESET BYTE/WORD BIT
29  LABEL 2
30   DPY SYSTEM MODE <Y-N>?A
31   IF REGA = 1 GOTO 3
32   REG8 = REG8 OR 1000      ! SET SYSTEM/NORMAL BIT IN HIGH ADDRESS
33  LABEL 3
34   REG8 = REG8 OR 0800      ! HIGH ADDRESS FOR STANDARD DATA SPACE
35   DPY DATA SPACE <Y-N>?A
36   IF REGA = 1 GOTO 4
37   DPY ENTER DATA SPACE TYPE/A      ! ELSE SELECT NEW SPACE
38   REGA = REGA SHL 8
39   REG8 = REG8 AND FOFF
40   REG8 = REG8 OR REGA              ! OR IN NEW SPACE
41  LABEL 4
42   EXECUTE 2
43
44
45  PROGRAM 2                 ! FAST RAM TEST USING Z8000 POD FAST RAM
46  LABEL 0                          ! CARRY HIGH_ADDRESS INTO PROGRAM
47   DPY STARTING SEGMENT?/A
48   IF REGA > 7F GOTO 1              ! SEGMENT VALUE OUT OF RANGE
49   REGB = REGA              ! SAVE SEGMENT VALUE FOR LATER COMPARISON
50   REGA = REGA OR REG8
51   WRITE @ F0000007 = REGA
52   DPY STARTING OFFSET?/A   ! BOTTOM 4 DIGITS OF STARTING ADDRESS
53   WRITE @ REGF INC = REGA
54   DPY ENDING SEGMENT?/A
55   IF REGA > 7F GOTO 1              ! SEGMENT VALUE OUT OF RANGE
56   IF REGB > REGA GOTO 2            ! ENDING SEGMENT HAS TO BE = OR GREATER
57                                           ! THAN STARTING SEGMENT
58   REGA = REGA OR REG8
59   WRITE @ REGF INC = REGA
60   DPY ENDING OFFSET?/A             ! BOTTOM 4 DIGITS OF ENDING ADDRESS
61   WRITE @ REGF INC = REGA          ! START RAM TEST
62  LABEL 3
63   READ @ REGF                              ! READ @ ENTER
64   IF REGE AND 000000F0 = F0 GOTO 4         ! ERROR DETECTED
65   IF REGE AND 000000FF = C0 GOTO 5         ! TEST DONE IF SECOND BYTE = 0
66   DPY $E                                   ! ELSE DISPLAY STATUS
67   GOTO 3
68
69  LABEL 4
70   READ @ REGF
```

Standard Troubleshooter Program (cont)

```
 71   REG1 = REGE AND 0000000F          ! MASK IN ERROR REPORTING BYTE
 72   READ @ F000000F                   ! GET HIGH ADDRESS OF ERROR
 73   REG2 = REGE AND 0000007F          ! MASK IN SEGMENT VALUE
 74   READ @ F0000010                   ! GET LOW ADDRESS OF ERROR
 75   REG3 = REGE
 76   READ @ F0000011                   ! GET BIT ERROR MASK
 77   REG4 = REGE
 78   IF REG1 = 0 GOTO 6
 79   DPY DCD ERR @ $2$3 BTS $4#
 80   GOTO 7
 81  LABEL 6
 82   DPY R/W ERR @ $2$3 BTS $4#
 83   GOTO 7
 84  LABEL 1
 85   DPY SEGMENT VALUE OUT OF RANGE#
 86   EXECUTE 5
 87   GOTO 0
 88  LABEL 2
 89   DPY INVERTED SEGMENT ENTRIES#
 90   EXECUTE 5
 91   GOTO 0
 92  LABEL 5
 93   DPY RAM OK
 94  LABEL 7
 95   STOP
 96
 97
 98  PROGRAM 3                         ! Z8000 POD FAST ROM PROGRAM
 99                                    ! CHECKSUM OF PROGRAM SPACE (C)
100   DPY ADDRESS INCREMENT?/A
101   WRITE @ F0000006 = REGA
102   DPY PROGRAM SPACE <Y-N>?A
103   IF REGA = 1 GOTO 0
104   REG8 = 0800
105   GOTO 1
106  LABEL 0
107   REG8 = 0D00
108  LABEL 1
109   DPY SYSTEM MODE <Y-N>?A
110   IF REGA = 1 GOTO 2
111   REG8 = REG8 OR 1000
112  LABEL 2
113   DPY WORD ACCESS <Y-N>?A
114   IF REGA = 1 GOTO 3
115   REG8 = REG8 OR 2000
116  LABEL 3
117   DPY STARTING SEGMENT?/A
118   IF REGA > 7F GOTO 4               ! SEGMENT VALUE OUT OF RANGE
119   REGB = REGA
120   REGA = REGA OR REG8
121   WRITE @ F000000B = REGA
122   DPY STARTING OFFSET?/A
123   WRITE @ REGF INC = REGA
124   DPY ENDING SEGMENT?/A
125   IF REGA > 7F GOTO 4               ! OUT OF RANGE SEGMENT VALUE
126   IF REGB > REGA GOTO 5             ! START SEGEMNT > FINISH SEGMENT
127   WRITE @ REGF INC = REGA
128   DPY ENDING OFFSET?/A
129   WRITE @ REGF INC = REGA
130  LABEL 6
131   READ @ REGF                       ! READ STATUS OF TEST
132   IF REGE AND 000000F0 = C0 GOTO 7
```

Standard Troubleshooter Program (cont)

```
 DPY $E
 GOTO 6
LABEL 4
 DPY OUT OF RANGE SEGMENT VALUE#
 EXECUTE 5
 GOTO 3
LABEL 5
 DPY INCORRECT SEGMENT SEQUENCE#
 EXECUTE 5
 GOTO 3
LABEL 7
 READ @ F0000011
 REG 1 = REGE
 READ @ F000000F
 DPY CHKSUM $E INAC BTS $1
 STOP

PROGRAM 4                          ! Z8000 POD FAST LOOPING FEATURE
 DECLARATIONS
 ASSIGN REG1 TO BIG_ADDRESS
LABEL 0
 DPY LOOP ADDRESS?/1
 DPY READ OPERATION <Y-N>?A
 IF REGA = 1 GOTO 1
 DPY DATA TO WRITE?/A
 WRITE @ REG1 = REGA               ! WRITE TO SELECTED LOCATION
 WRITE @ F0000004 = REGA           ! NOW LOOP ON IT
 DPY LOOPING WRITE
 EXECUTE 5
 DPY LOOP ON ANOTHER ADDRESS <Y-N>?A
 IF REGA = 1 GOTO 0
 GOTO 2
LABEL 1
 READ @ REG1                       ! READ AT SELECTED LOCATION
 READ @ F0000004                   ! NOW LOOP ON IT
 DPY LOOPING READ
 EXECUTE 5
 DPY LOOP ON ANOTHER ADDRESS <Y-N>?A
 IF REGA = 1 GOTO 0
LABEL 2

PROGRAM 5                          ! DELAY
 DECLARATIONS
 ASSIGN REG1 TO COUNTER
 REG1 = 30
LABEL 0
 REG1 = REG1 DEC
 IF REG1 > 0 GOTO 0
```

9010A Language Compiler Program

```
        FORCELN WAIT = 5
        FORCELN BUSREQ = 4
        BUSADR = 0800FFFE
        UUTADR = 0
        DECLARATIONS
        assign reg8 to high_address
        assign reg9 to low_address
        assign rega to keyboard
        assign regb to scratch

SETUP INFORMATION
        pod - z8000
        enable wait-yes
        enable busreq-yes

program MAIN
        dpy FAST Z8K OPERATIONS
        execute DELAY
keyloop:                                ! scroll through options
        dpy FAST RAM TEST <Y-N>?keyboard
        if keyboard = 0 goto romtest
        execute FASTRAM                 ! do the fast RAM test
romtest:
        dpy FAST ROM TEST <Y-N>?keyboard
        if keyboard = 0 goto floop
        execute FASTROM                 ! do the fast ROM test
floop:
        dpy FAST LOOP <Y-N>?keyboard
        if keyboard = 0 goto keyloop
        execute FASTLOOP                ! do the fast loop operation
        goto keyloop


program FASTRAM
        dpy WORD INCREMENT <Y-N>?keyboard
        if keyboard = 1 goto ramword
rambyte:
        write @ F0000006 = 1            ! set byte/word bit
        high_address = 2000             ! byte type in high address
        goto system
ramword:
        write @ F0000006 = 2
        high_address = 0                ! reset byte/word bit
system:
        dpy SYSTEM MODE <Y-N>?keyboard
        if keyboard = 1 goto d_space
        high_address = high_address or 1000 ! set S/N bit in high address
d_space:
        high_address = high_address or 0800 ! high addr for std data space
        dpy DATA SPACE <Y-N>?keyboard
        if keyboard = 1 goto doram
        dpy ENTER DATA SPACE TYPE/keyboard  ! else select new space
        keyboard = keyboard shl 8
        high_address = high_address and F0FF
        high_address = high_address or keyboard  ! or in new space
doram:
        execute DO_F_RAM
```

9010A Language Compiler Program (cont)

```
program DO_F_RAM                    ! fast RAM test using z8000 pod fast RAM
retry:                              ! carry high_address into program
        dpy STARTING SEGMENT?/keyboard
        if keyboard > 7F goto err1      ! segment value out of range
        scratch = keyboard              ! save segment value for comparison
        keyboard = keyboard or high_address
        write @ F0000007 = keyboard
        dpy STARTING OFFSET?/keyboard   ! bottom 4 digits of starting address
        write @ ADR inc = keyboard
        dpy ENDING SEGMENT?/keyboard
        if keyboard > 7F goto err1      ! segment value out of range
        if scratch > keyboard goto err2 ! ending segment has to be = or greater
                                        ! than starting segment
        keyboard = keyboard or high_address
        write @ ADR inc = keyboard
        dpy ENDING OFFSET?/keyboard     ! bottom 4 digits of ending address
        write @ ADR inc = keyboard      ! start RAM test
stat_lp:
        read @ ADR                      ! READ @ ENTER
        if DAT and 000000F0 = F0 goto ram_err   ! error detected
        if DAT and 000000FF = C0 goto ram_ok    ! test done if second byte = 0
        dpy $e                                  ! else display status
        goto stat_lp

ram_err:
        read @ ADR
        reg1 = DAT and 0000000F         ! mask in error reporting byte
        read @ F000000F                 ! get high address of error
        reg2 = DAT and 0000007F         ! mask in segment value
        read @ F0000010                 ! get low address of error
        reg3 = DAT
        read @ F0000011                 ! get bit error mask
        reg4 = DAT
        if reg1 = 0 goto rw_err
        dpy DCD ERR @ $2$3 BTS $4#
        goto ramexit
rw_err:
        dpy R/W ERR @ $2$3 BTS $4#
        goto ramexit
err1:
        dpy SEGMENT VALUE OUT OF RANGE#
        execute DELAY
        goto retry
err2:
        dpy INVERTED SEGMENT ENTRIES#
        execute DELAY
        goto retry
ram_ok:
        dpy RAM OK
ramexit:
        stop

program FASTROM                         ! z8000 pod fast ROM program
                                        ! checksum of program space (c)
        dpy ADDRESS INCREMENT?/keyboard
        write @ F0000006 = keyboard
        dpy PROGRAM SPACE <Y-N>?keyboard
        if keyboard = 1 goto doromp
        high_address = 0800
        goto dosys
doromp:
        high_address = 0D00
```

9010A Language Compiler Program (cont)

```
dosys:
        dpy SYSTEM MODE <Y-N>?keyboard
        if keyboard = 1 goto dobyte
        high_address = high_address or 1000
dobyte:
        dpy WORD ACCESS <Y-N>?keyboard
        if keyboard = 1 goto dorom
        high_address = high_address or 2000
dorom:
        dpy STARTING SEGMENT?/keyboard
        if keyboard > 7F goto romerr1   ! segment value out of range
        scratch = keyboard
        keyboard = keyboard or high_address
        write @ F0000008 = keyboard
        dpy STARTING OFFSET?/keyboard
        write @ ADR inc = keyboard
        dpy ENDING SEGMENT?/keyboard
        if keyboard > 7F goto romerr1   ! out of range segment value
        if scratch > keyboard goto romerr2  ! start segment > finish segment
        write @ ADR inc = keyboard
        dpy ENDING OFFSET?/keyboard
        write @ ADR inc = keyboard
stat_lp:
        read @ ADR                      ! read status of test
        if DAT and 000000F0 = C0 goto done
        dpy $e
        goto stat_lp
romerr1:
        dpy OUT OF RANGE SEGMENT VALUE#
        execute DELAY
        goto dorom
romerr2:
        dpy INCORRECT SEGMENT SEQUENCE#
        execute DELAY
        goto dorom
done:
        read @ F0000011
        reg 1 = DAT
        read @ F000000F
        dpy CHKSUM $e INAC BTS $1
        stop
program FASTLOOP                        ! z8000 pod fast-looping feature
        DECLARATIONS
        assign reg1 to big_address
flp:
        dpy LOOP ADDRESS?/big_address
        dpy READ OPERATION <Y-N>?keyboard
        if keyboard = 1 goto fastread
        dpy DATA TO WRITE?/keyboard
        write @ big_address = keyboard  ! write to selected location
        write @ F0000004 = keyboard     ! now loop on it
        dpy LOOPING WRITE
        execute DELAY
        dpy LOOP ON ANOTHER ADDRESS <Y-N>?keyboard
        if keyboard = 1 goto flp
        goto flpexit
```

9010A Language Compiler Program (cont)

```
fastread:
        read @ big_address              ! read at selected location
        read @ F0000004                 ! now loop on it
        dpy LOOPING READ
        execute DELAY
        dpy LOOP ON ANOTHER ADDRESS <Y-N>?keyboard
        if keyboard = 1 goto flp
flpexit:

program DELAY
        DECLARATIONS
        assign reg1 to counter
        counter = 30
dloop:
        counter = counter dec
        if counter > 0 goto dloop
```