CSS
9000A-Z8000

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 (R)

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

Section1
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.

I-I. PURPOSE OF THE INTERFACE POD

The 9000A-Z8000 Interface Pod (hereafter referred to as the Pod) interfaces any Fluke YOOO 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 hoards, 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 28002 and Z8004 versions. A switch on the Pod selects between Segmented Memory28001 and Z8002 versions, and Virtual Memory 28003 and Z8004 versions. Unless otherwise specified, references to the "Z8000" in this manual refer to any of the 28000 family of microprocessors.

I-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 I-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.

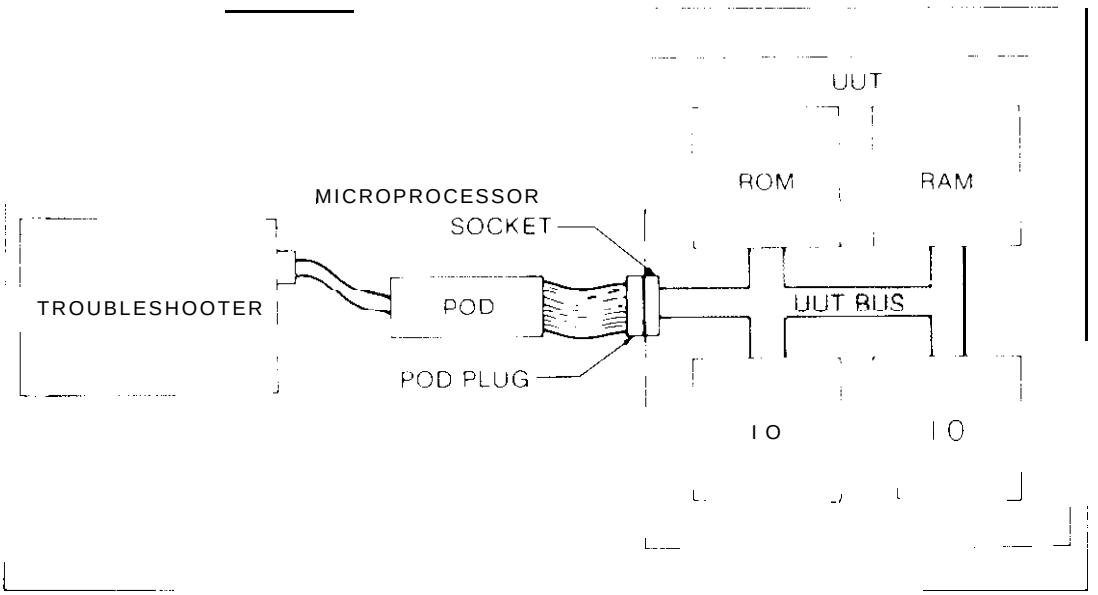


Figure 1-1. Communication Between the Troubleshooter. the Pod. and the UUT

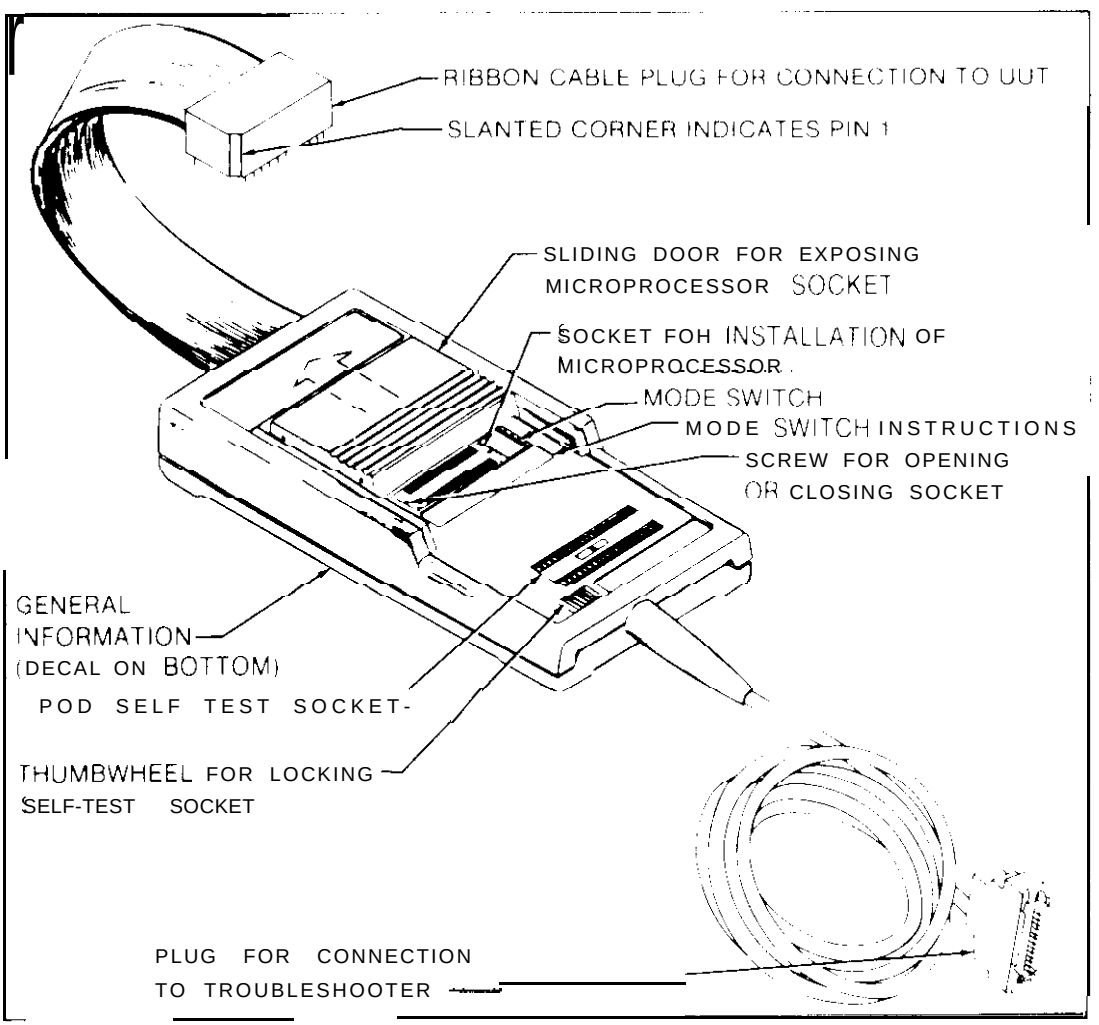


Figure 1-2. External Features of the Z8000 Interface Pod

'The Troubleshooter supplies operating power (+5V) for the Pod. The LJUT 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 U UT. 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 I-I

Table I-I. 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 lo 28000 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	
Sire	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, Rt-<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

Z-I. 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-l) 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 I 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.

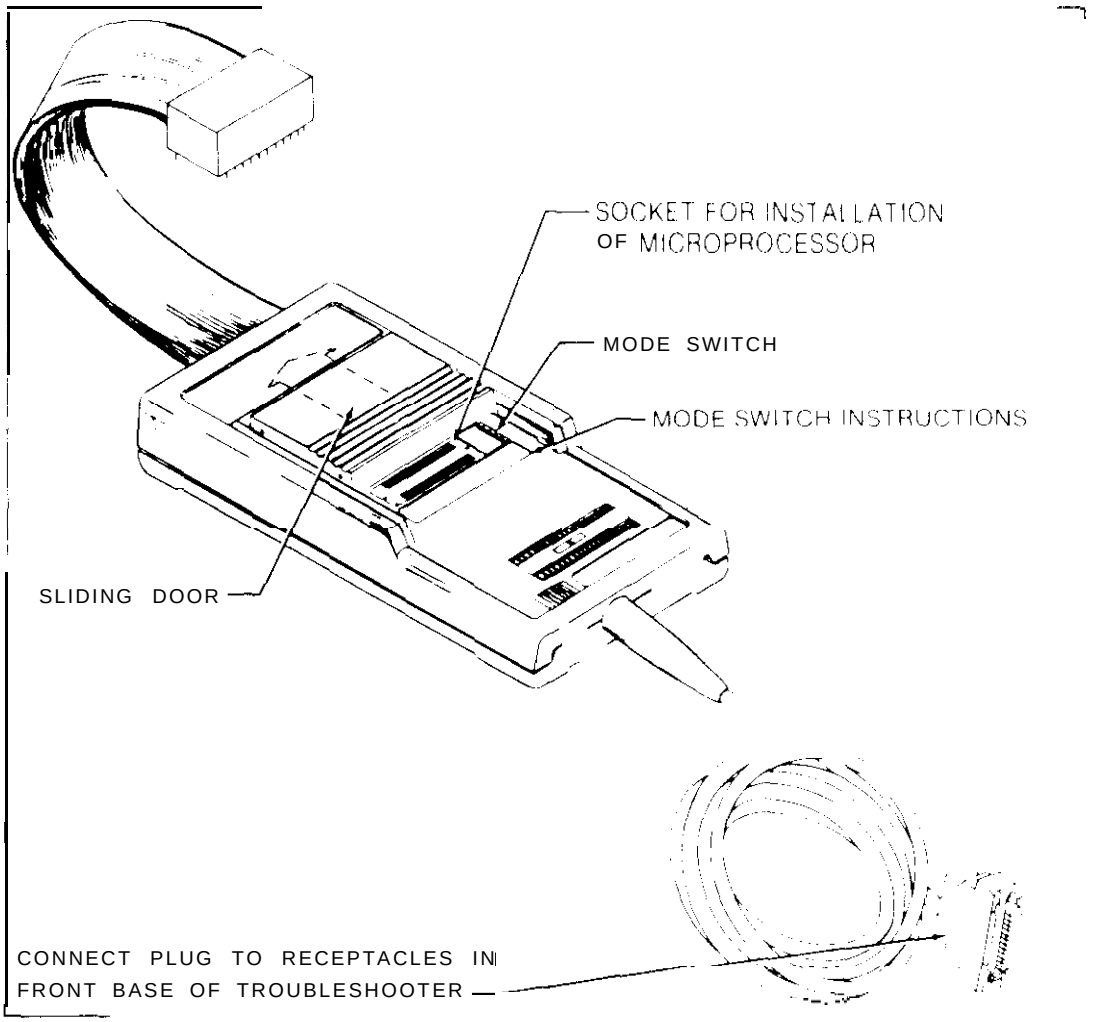


Figure 2-1. Location of Microprocessor Socket and Mode Switches

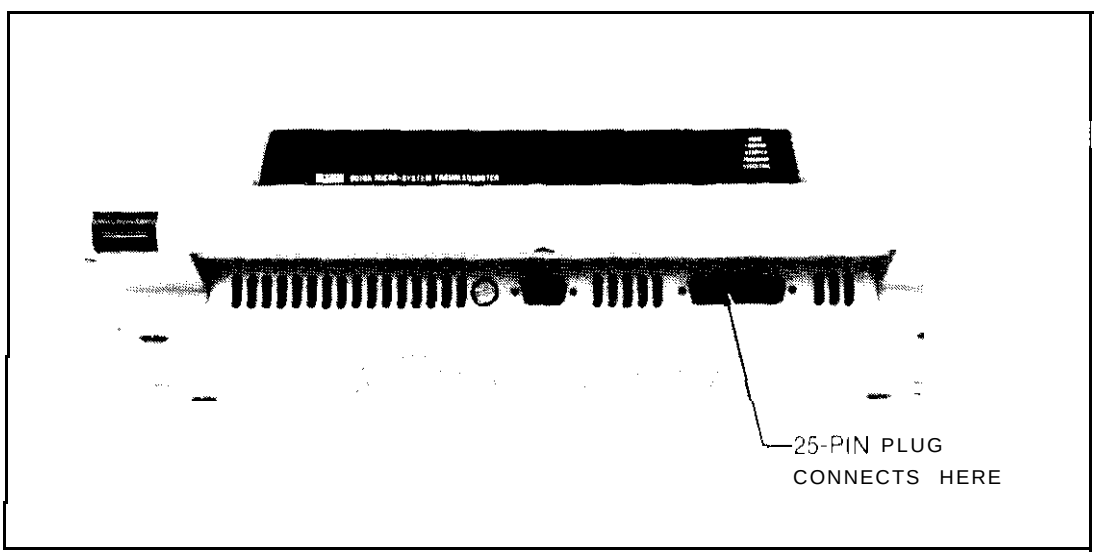


Figure 2-2. Connection at Interface Pod to Troubleshooter

2-4. PERFORMING THE POD SELF TEST

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

1. Make sure that a6 MHz or greater clock speed microprocessorisinstalledinthe 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 I 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-I. INTRODUCTION

This section contains microprocessor data which may he useful during operation of the Troubleshooter. 'This information includes descriptions of Z8000 signals and pin assignment\.

3-2. MICROPROCESSOR SIGNALS

Table 3-1 lists all of the Z8000 microprocessor signals and provides a briefdescription 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-I through 3-4 show the **ZXOOO** family pin assignments.

Table 3-1. Signal Descriptions

SIGNAL NAME	DESCRIPTION
ABORT	The Abort line is used in conjunction with the $\overline{SAT}$ line to interrupt instructions before they are completed. (Available on the Z8003 and Z8004 only.)
AD0-AD15	These 16 tri-statemultiplexed Address/Data lines are used to address memory and for Input/Output. The lines contain address information when the Address Strobe ($\overline{AS}$) linerises and data when the Data Strobe ($\overline{DS}$) line rises,
AS	The rising edge of this Address Strobe line indicates valid addresses.
$\overline{BUSACK}$	When this Bus Acknowledge line is Low. the CPU has relinquished control of the bus.
$\overline{BUSREQ}$	The Bus Request line is driven Low to request the bus from the CPU.
$\overline{DS}$	The rising edge of the Data Strobe line indicates valid data available on the multiplexed Address/Data ($\overline{AD}$) lines.
$\overline{MREQ}$	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,
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.)
SEGT	The Segment Trap isasserted 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 28003 only,)
ST0-ST3	These Status lines indicate the CPU status (refer to Table 3-3 Z8000 CPU Status Codes).
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 isnotready for a transfer of data.
B/$\overline{W}$	Byte/Word specifies the nature of the 16-bit information on the address/data bus.
N/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	ABORT	input	low	tri
Address/Data	AD0-AD15	output	high	tri
Address Strobe	$\overline{AS}$	output	low	tri
Bus Acknowledge	$\overline{BUSACK}$	output	low	
BUS Request	$\overline{BUSREQ}$	input	low	tri
Memory Request	MREQ	output	low	
Multi-Micro In	$\overline{MI}$	input	low	
Multi-Micro Out	MO	output	low	
Non-Maskabie Interrupt	NMI	input	low	
Non-Vectored Interrupt	NVI	input	low	
System Clock	CLK	input	—	
Reset	RESET	input	low	tri
Read/Write	R/$\overline{W}$	output	high/low	
Segment Page Address Translation Trap	SAT	input	low	
Segment Trap	SEGT	input	low	tri
Segment Number	SN0-SN6	output	high	tri
status	ST0-ST3	output	high	
stop	STOP	input	low	
Vectored Interrupt	$\overline{VI}$	input	low	
Wait	WAIT	input	low	tri
Byte/Word	B/$\overline{W}$	output	high/low	tri
Normal/System Mode	N/$\overline{S}$	output	high/low	

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

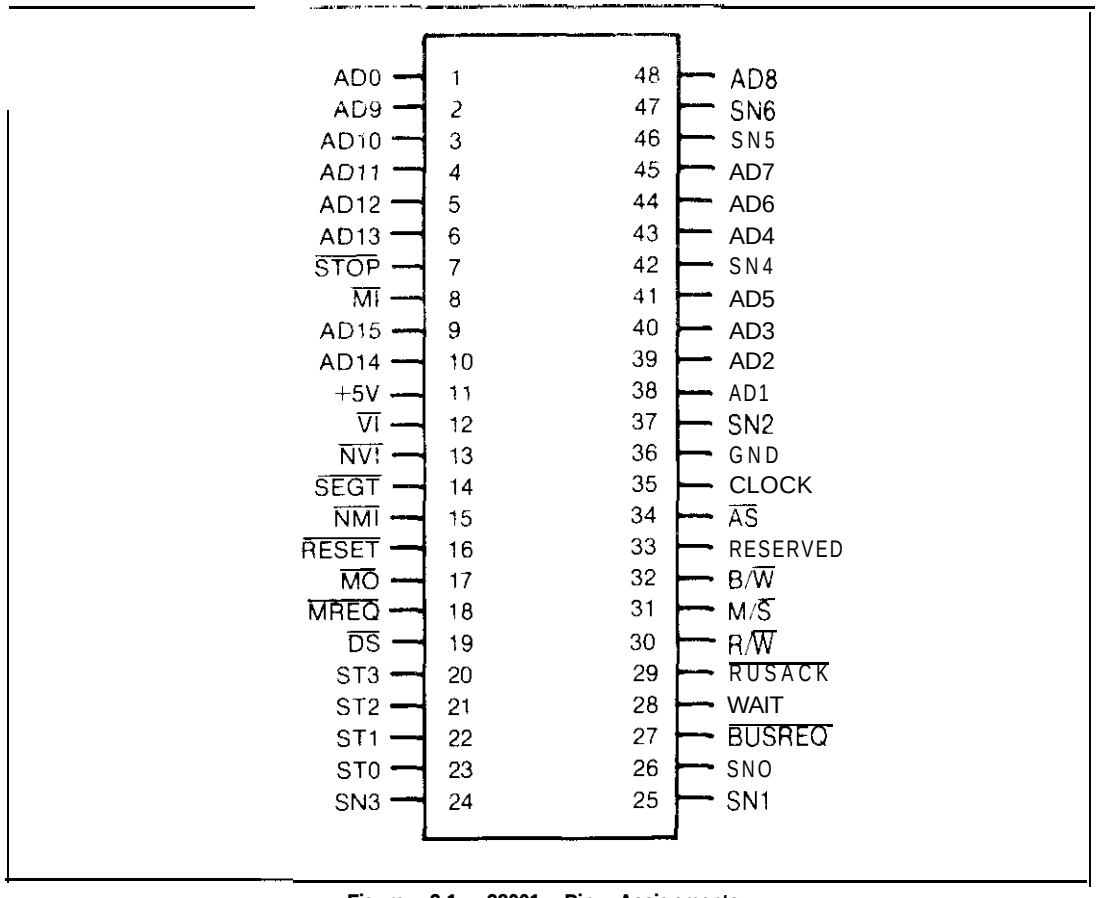


Figure 3-1. 28001 Pin Assignments

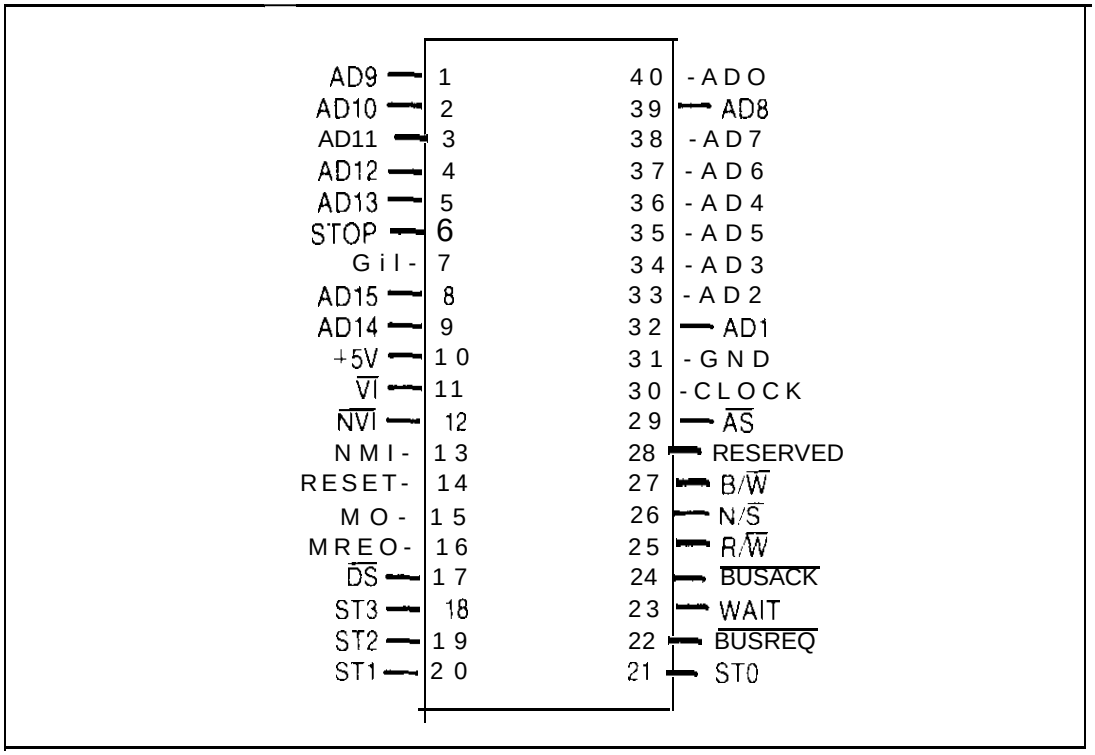


Figure 3-2. 28002 Pin Assignments

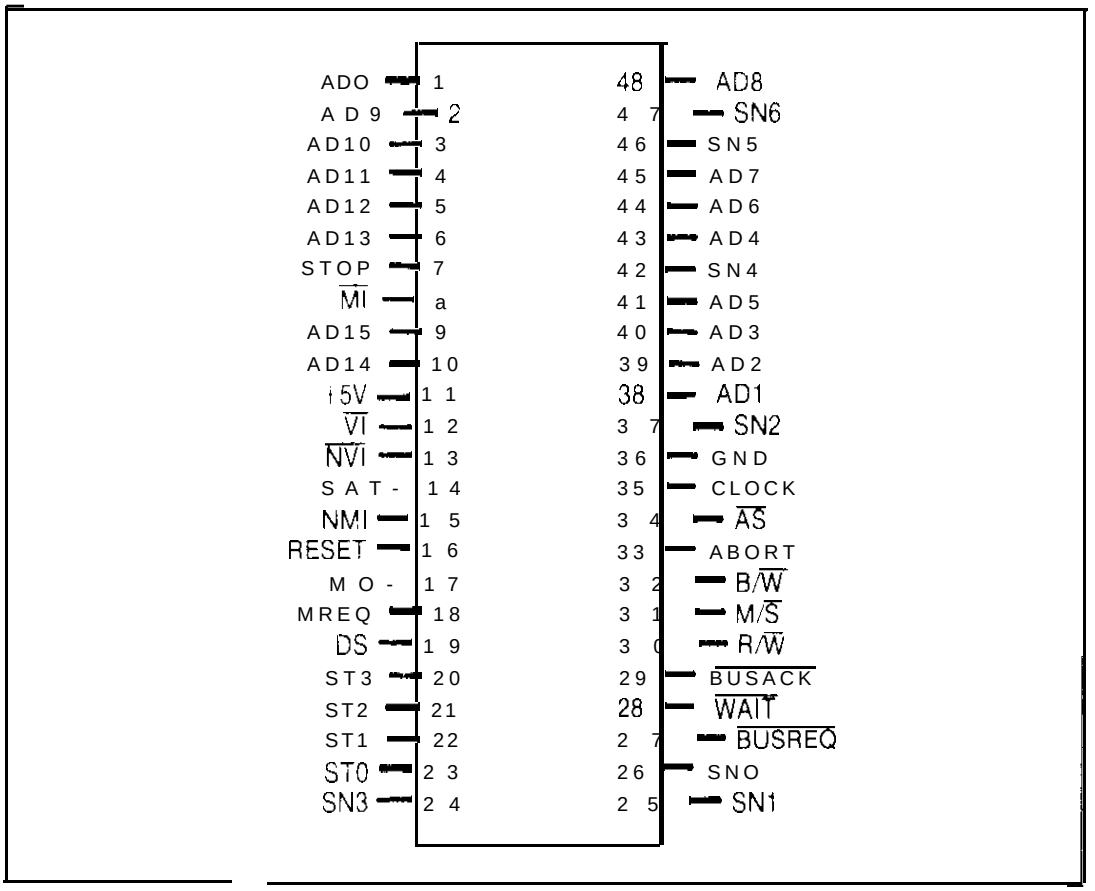


Figure 3-3. 28003 Pin Assignments

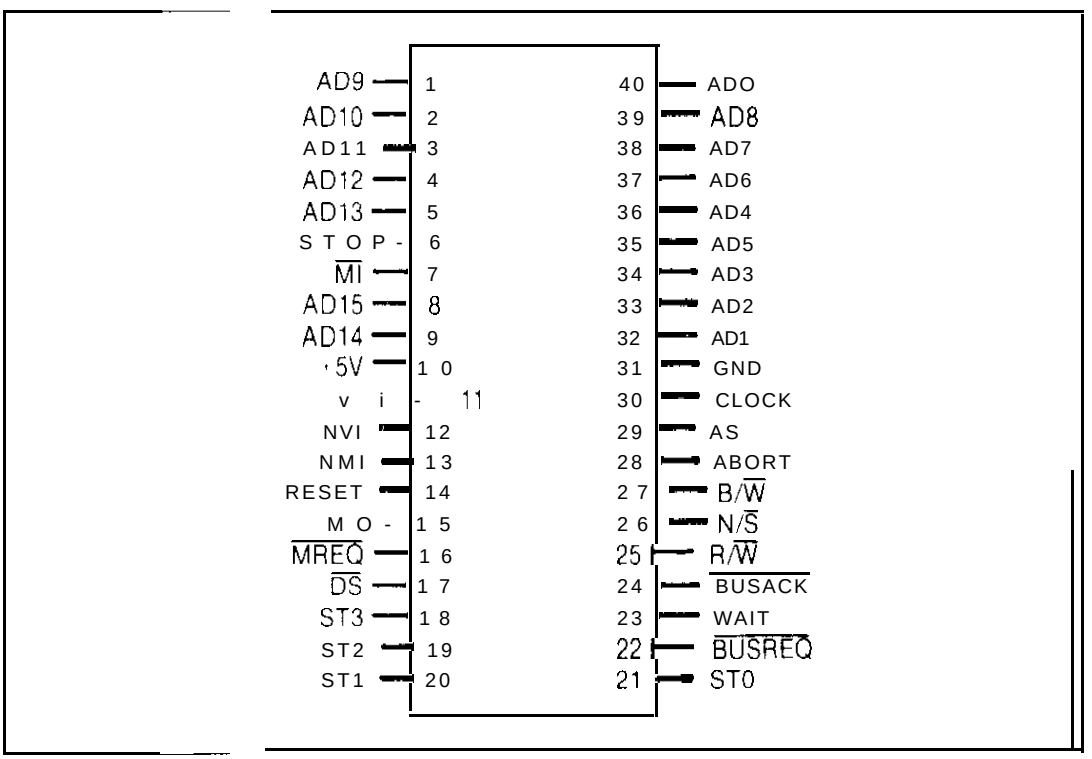


Figure 3.4. 28004 Pin Assignments

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 hit 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 U UT is asserting a forcing status line: either the $\overline{\text{BUSREQ}}$ (Bus Request), or 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 keyallowsyoutoseewhichlineis 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 a" *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 28000 Pod in this section.

4-3. ADDRESS SPACE ASSIGNMENT

4-4. Introduction

Ail of the Z8000 family of microprocessors have 16 multiplexed address lines (ADO - AD15) which allow direct addressing of 64K bytes of memory. Inaddition, the Z8001 and 28003 versions have seven segment lines (SN0-SN6) which select one of 128 64K address segments, allowing a total addressing range of 8M bytes. The 28000 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 28000, a simplified address descriptor is used for specifying addresses via the Troubleshooter.

Addresses for the segmented versions of the 28000 microprocessor (28001 and 28003) are normally defined by a" Offset (a" address within a 64K byte block) and a Segment (one of 128 possible memory blocks).

64K Byte Offset Addresses

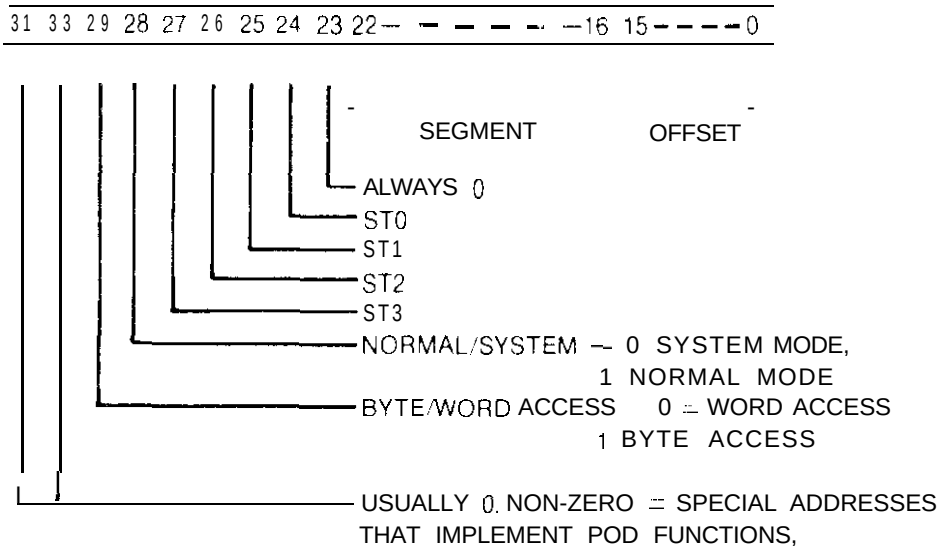
Address offsets (and addresses for non-segmented versions) are specified by bits 0-15 of the address. This provides an address offset range of 0000- FFFF. The address offset is put on lines AD0-AD15 during a bus access by the Pod.

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 "sing the Troubleshooter. the seven segment hits appear in bits 16-22 of the address.

As a" added convenience for the operator. the high byte of the address designates Normal/System mode and Byte/Word operation. There arc 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 ZKOOO family of microprocessors provides for both word and hpte 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/WORD) line to the UUT will he 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, hit 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.

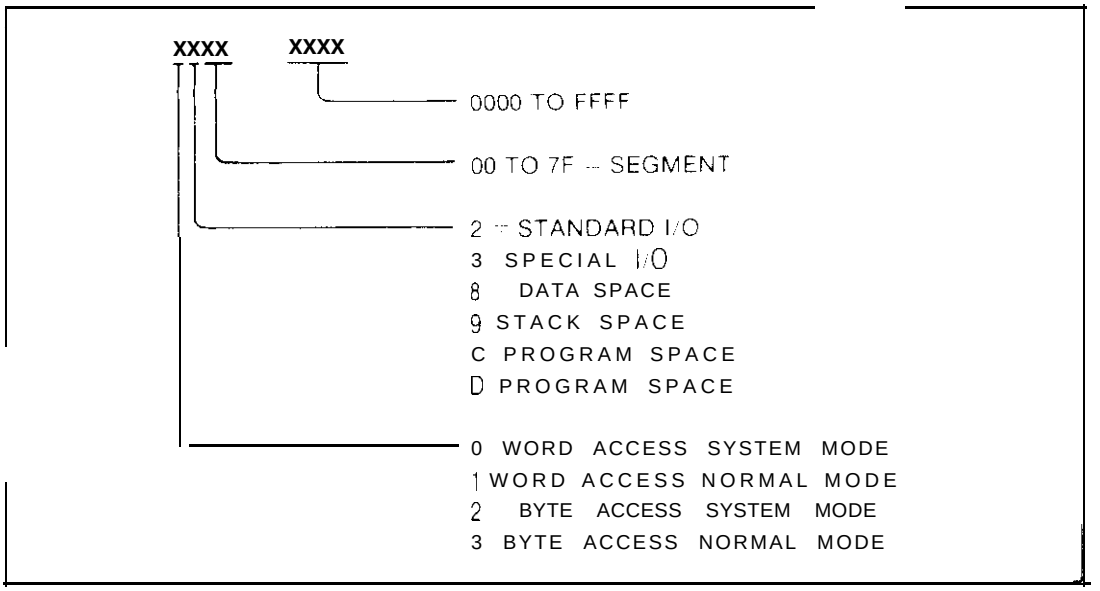
Normal System Mode

The Z8000 microprocessors provide a control signal (N/$\overline{S}$) to the UUT that indicates the CPU's operating mode. This signal can he controlled through setting or clearing bit 28 in the address. If bit 2X 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 combination\ of the four status lines, the Normal/$\overline{System}$ status line. and the Byte; Word status line will occur. All possible status code combinations are shown in Table 4-l. status Codes.

Table 4-I. 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 FOOO 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 28000 Pod.

For example, the address

F000 00 16

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

4-6. STATUS/CONTROL LINES

4-i'. 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 11101111 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	$\overline{SAT}$ ACKNOWLEDGE FLAG	15	
14	$\overline{VI}$ ACKNOWLEDGE FLAG	14	—
13	$\overline{NVI}$ ACKNOWLEDGE FLAG	13	
12	$\overline{NMI}$ ACKNOWLEDGE FLAG	12	—
11	TIMEOUT FLAG	11	$\overline{AS}$
10	$\overline{MI}$	10	$\overline{MO}$
9	RESET	9	N/S
a	** $\overline{VI}$	a	MREQ
7	** NVI	7	R/$\overline{W}$
6	** $\overline{NMI}$	6	B/$\overline{W}$
5	. . $\overline{SAT}$	5	$\overline{DS}$
4	POWER FAIL FLAG	4	ST3
3	. ABORT (Z8003, Z8004 ONLY)	3	ST2
2	. *** WAIT	2	ST1
1	* STOP	1	ST0
0	* *** $\overline{BUSREQ}$	0	. $\overline{BUSACK}$

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

. USER WRITEABLE

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

NO *TE*

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 28000 signals, but are generated within rhe 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 ZXOOO has two inputs ($\overline{BUSREQ}$ and 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 $\overline{WAIT}$ 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 $\overline{WAIT}$ 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 $\overline{WAIT}$ 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 $\overline{\text{NMI}}$, $\overline{\text{NVI}}$, VI, and SEGT(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 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 IX line is not drivable. The Troubleshooter will then display the message *CTL ERR 00000000 00100000-LOOP?*. The zeros and ones correspond to the kit 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 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
FOOO 0000	Read $\overline{NMI}$ acknowledge word
FOOO 0001	Read NVI acknowledge word
FOOO 0002	Read $\overline{VI}$ acknowledge word
FOOO 0003	Read $\overline{SAT}$ acknowledge word
FOOO 0004	Fast-looping read/write at last address
FOOO 0005	Read/write default high address
FOOO 0006	Read/write fast RAM increment
FOOO 0007	Read/write fast RAM start high address
FOOO 0008	Read/write fast RAM start offset
FOOO 0009	Read/write fast RAM end high address
FOOO OOOA	Read/write fast RAM end offset
F000 000B	Read/write fast ROM start high address
FOOO OOOC	Read/write fast ROM start offset
FOOO OOOD	Read/write fast ROM end high address
FOOO OOOE	Read/write fast ROM end offset
FOOO OOOF	Read fast RAM error high address; ROM checksum
FCQO 0010	Read fast RAM error low address! Inactive ROM bits
FOOO 0011	Read/write refresh enabled
FOOO 0012	Readiwrite refresh rate
FOOO 0013	Read/write transparent read high address
FOOO 0014	Read/write transparent read offset
FOOO 0015	Read/write runuut FCW
FOOO 0016	Read/write state of $\overline{MO}$ output
FOOO 0017	Read/write state of fast mode
FOOO 0018	Read/write state of continuous Interrupt flag
FOOO 0019	Read last address high errors (no dummy read)
FOOO 001A	Read last address low errors (no dummy wad)
FOOO 001 B	Read last data drivability (no dummy read)
FOOO 001 C	Read last control errors (no dummy read)
FOOO 001 D	Read last forcing line errors (no dummy read)
FOOO 001 E	Read last status (no dummy read)
FOOO 001 F	Read last error summary (no dummy read)
FOOO 0020	Read/write control drivability mask
FOOO 0021	Read/write forcing line reporting mask
FOOO 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 he preferable for some uses over the normal method of loading individual special addresses that is dcrcrihed 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 U UT 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 OOFE (with status code hits of 1000), writing the data 8C17.

The Quick-Looping function may he used for read or write operations at any of the valid 28000 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
FOOO 0000	$\overline{NMI}$ Acknowledge Word (paragraph 4-27)
FOOO 0001	$\overline{NVI}$ Acknowledge Word (paragraph 4-27)
FOOO 0002	$\overline{VI}$ Acknowledge Word (paragraph 4-27)
FCOO 0003	$\overline{SAT}$ Acknowledge Word (paragraph 4-27)
FOOO 0004	Fast Looping Read/Write at Last Address (paragraph 4-23, 4-28)
FOOO 0005	Default High Address (paragraph 4-29)
FOOO 0006	Fast RAM/ROM Test Increment (paragraph 4-24. 4-25)
F000 0007	Fast RAM Test Start High Address (paragraph 4-24)
FOOO 0008	Fast RAM Test Start Offset (paragraph 4-24)
FOOO 0009	Fast RAM Test End High Address (paragraph 4-24)
FOOO OOOA	Fast RAM Test End Offset (paragraph 4-24)
	Code — Meaning
	XX00 — No test requested
	XXBO — Busy. R/W test in progress
	XXB1 — Busy, performing decode test
	xxco — Complete, no errors
	XXFO — Fail Read/Write test
	XXF1 — Failed Address Decode test
	xx — Indicates the upper byte of the current RAM offset being tested.
FOOO 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
	XXBO — 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 001 A	State of Continuous Interrupt Flag (paragraph 4-38)
F000 001 B	Last Address High Drivability Error (No Dummy Read) (paragraph 4-39)
F000 001 C	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 001 F	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) OUT 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 MCRE 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 28000 Pod

UNCTION	SPECIAL ADDRESSES AND OPERATIONS	DESCRIPTION OF USE
Quick Looping Write	Write @ XYYY YYYY = ZZZZ	Performs a normal write of data ZZZZ at theaddress XYYY YYYY. X may only be O-3 hex.
	Write @ FOOO 0004 enter enter	Causes the Pod toperform 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 O-3 hex,
	Read @ FOOO 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 morequickly 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 hits held high or low. After the read-write check is completed, a unique hit 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.**

I he **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 he written** into **FOOO 0006. To specify the starting address, write the top four digits of the address into special address FOOO 0007 and the address offset into FOOO** 0008. **The top four digits of the ending address segment should he written into FOOO 0009 and the offset into** F000 000A. **Either word or byte addresses may he used. The** ending **address must** be **greater than the starting address, and both addresses must he even for word addresses.** The **status code assigned to the beginning address will also he used for the ending address. I he address increment value must be even for word addresses.**

The Quick R A M **'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 he 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-l. The upper byte of the Pod response shows bits 8 through IS of theaddress undertest. 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 FOOOOOO6 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 FOOO 0007 and F000 0008 respectively:

   WRITE @ FOOOOOO7 = 0800

   WRITE @ FOOOOOO8 = 5500

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

   WRITE@ FOOOOOO9 = 0800

   WRITE @ F000 000A = 7510

   Writing the ending offset into location FOOO OOOA 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 beencompleted before reading at any ofthespecial 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 @ FOOO 0006 = Z	Specifies the address increment to be used by the quick RAM test. If Z-O, the address increment defaults to 2 (word increments).
Write @ FOOO 0007 = UUUU	Specifies the upper word UUUU of the start address for the quick RAM test.
Write @ FOOO 0006 = LLLL	Specifies the lower word of the start address (offset) for the quick RAM test.
Write @ FOOO 0009 = UUUU	Specifies the upper word UUUU of the ending address for the quick RAM test.
Write @ FOOO OOOA = LLLL	Specifies the lower word of the ending address (offset) for the quick RAM test. Execution ofthequick RAM test beginsatthecompletionofthis 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 XXFO — Fail Read/Write test XXF1 — Failed Address Decode test x x — indicates the upper byte of the current RAM offset being tested.
Read @ FOOOOOOF	High word of the error address.
Read @ FOOO 0010	Low word of the error address.
Read @ FOOO 0011	Hex Mask of the bad binary bits from a Read/Write failure.
Read @ FOOO 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 he 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 X) accesses will he per-formed. If the user specifies a ROM test in I 0 space (status = hex 2 or 3) or stack space (status = hex 9), it will he mapped to the data space.

Testing of high and low ROM's separately can he achieved by using a byte type operation (hit 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 he greater than the starting address.

Like the Quick KAM Test, the Quick ROM I est may be aborted by selecting another operation. lo 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 'I able 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 arc described in Table 4.7

4-26. SPECIAL FEATURES OF THE 26000 POD

The following paragraphs describe special features of the ZROOO 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 @FOOO 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 @ FOOO OOOB = UUUU	Specifies the upper word UUUU of the start address for the quick ROM test.
Write @ FOOO OOOC = LLLL	Specifies the lower word of the start address (offset) for the quick ROM test.
Write @ FOOO OOOD = UUUU	Specifies the upper word UUUU of the ending address for the quick ROM test.
Write @ FOOO OOOE = LLLL	Specifies the lower word of the ending address (offset) for the quick ROM test, Execution of the quick ROM test beginsatthecompletionofthis 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 XXBO — Busy, test in progress xxco — Complete, no errors XXC1 — Complete, inactive bits xx — indicates the upper byte of the current ROM offset being tested.
Read @ FOOO OOOF	Checksum — not related to the ROM signature that is obtained from the standard ROM test.
Read @ FOOO 0010	Hex mask indicating inactive bits detected during test.
Read @ FOOO 0012	Last Test Status.

4-27. Interrupt Acknowledge Words (Addresses FOOO 0000 - FOOO 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).

NMI Acknowledge Word (Address F000 0000)

$\overline{\text{NVI}}$ Acknowledge Word (Address F-000 0001)

VI Acknowledge Word (Address FOOO 0002)

SAT Acknowledge Word (Address F000 0003)

4-28. Fast-Looping Read/Write at Last Address (Address FOOO 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 FOOO 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 FOOO 0006 - FOOO 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 FOOO 0006)

Fast RAM Start High Address (Address FOOO 0007)

Fast KAM Start Offset (Address F000 0008)

Fast RAM End High Address (Address FOOO 0009)

Fast KAM End Offset (Address F000 000A)

Fast ROM Start High Address (Address FOOO 000B)

Fast ROM Start Offset (Address F000 OOOC)

Fast ROM End High Address (Address F000 000D)

Fast KOM End Offset (Address F000 000E)

Fast RAM Error High Address/ROM Checksum (Address F000 OOOF)

Fast RAM Error Low Address! Fast KOM inactive Bits (Address F000 0010)

Fast RAM Error Bits (Address FOOO 001 I)

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-(l). 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 FOOO 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 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 MO line by writing to this address or observe the current state of MO by reading this address. The default is high (inactive).

4-35. Refresh Enabled (Address FOOO 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 I (Refresh enabled). The default is ENABLED.

4-36. Refresh Rate (Address F000 0016)

This address contains a value used to set the rate in the Z8000's RAM Refresh Counter. The value (within the range 0-64) isshifted 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 $\overline{WAIT}$ 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 $\overline{WAIT}$ line is only honored during UUT accesses, and the Pod will not be allowed to timeout due to a stuck $\overline{WAIT}$ line. If $\overline{WAIT}$ 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/$\overline{W}$ line

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

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 he the only ones effected by the Setup command. For example:

READ @F000 001F = 0005 OK

shows that both bit 2 ($\overline{\text{WAIT}}$) and hit 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 FOOO 0020)

The status word from the immediately previous Pod operation may he read at this address. The data obtained from this operation maybe different from that obtained with a *READ @ S JS* 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 ULJT operation. The data returned is displayed in hexadecimal rather than binary, as is the case with the *READ @* STS command, hut 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 hit assignments.

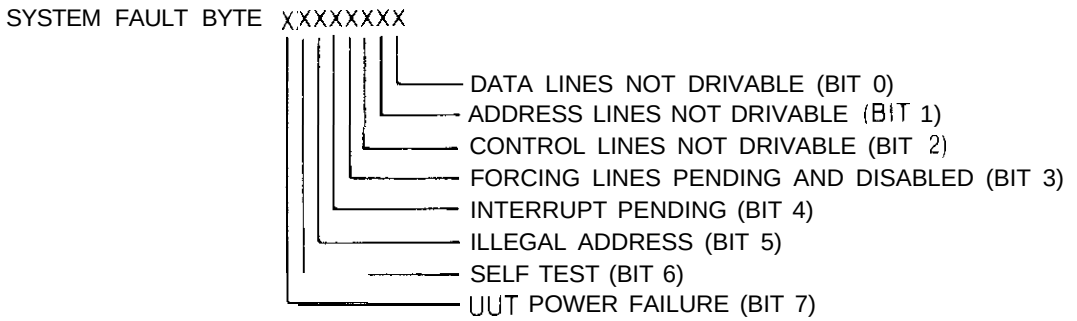
For example,

READ @F000 0020 = 0314

shows 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 FOOO 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 hy the Pod during the immediately previous UUT operation may he 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 inhihited, performing a *READ @ F000 0020 = 0018* indicates that:

1. An interrupt is pending.
2. Forcing line(s) are pending hut disabled

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

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

Control Driveability Error Reporting Mask (Address FOOO 0022)

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

Default value = FFFF.

For example. a certain 28000 U UT may not allow the processor to drive the $\overline{\text{MO}}$ line low. If this is considered normal. performing a *WRITE @ F000* 0022 = *FBFF* will inhibit the reporting of MO line drivability errors during BUS TEST, while allowing drivability error reporting for all of the other control lines.

Forcing Line Keporting Mask (Address FOOO 0023)

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

Default value = FFFF

For example. with the Pod in the fast mode. a certain UUT asserts the 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 inhihit the reporting of the timeout flag while allowing forcing line error reporting for the status inputs.

4-41. Sell Test Diagnostic (Address FOOO 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. Hus 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 other-wise 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 he 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 hits 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 0 0 0 0 F000 0003.

Special address f-000 0018 provides the capability to enable continuous interrupts. such as might be needed to trigger an oscilloscope. Refer to Special Features of the 28000 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. hut will occur only during an interrupt acknowledge bus cycle.

If Free-Run is selected, then a sync pulse of2 μsec duration occurring at a frequency of approximately I 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 rhe 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, eve" if Free-Run is selected. If Free-Ku" is selected, the oscilloscope trigger output pulses remain synchronized to the previous sync mode selected. At power on the prob" is in Free-Kun, hut 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. hut 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 U UT 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 28000 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 U UT 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.