NOTE

This manual documents the Model 9000A-8051 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-8051

Interface Pod

Instruction Manual

P/N 744698
OCTOBER 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.

PURPOSE OF INTERFACE POD 1-1.

The 9000A-8051 Interface Pod interfaces any 9000 Series Micro System Troubleshooter to equipment using an 8051- or 8044-family microprocessor.

The Micro System Troubleshooter (referred to hereafter as the Troubleshooter) is used to service printed circuit boards, instruments and systems that use a microprocessor. The 9000A-8051 Interface Pod (referred to as the Pod) adapts the general purpose architecture of the Troubleshooter to a specific architecture of the 8051 and 8044 microprocessor families. The Pod adapts such microprocessor-specific functions as pin layout, status/control functions, interrupt handling, timing, and memory and I/O addressing.

DESCRIPTION OF POD 1-2.

Figure 1-1 shows the communication between the Pod, the Troubleshooter, and the unit-under-test (referred to as the UUT). The Pod connects to the Troubleshooter through a shielded 24-conductor cable. The Pod connects to the UUT through the microprocessor socket. The UUTs microprocessor is removed from the UUT and is replaced by the Pod ribbon cable plug, which gives the Troubleshooter access to all system components which normally communicate with the microprocessor. A microprocessor of the same type that was removed from the UUT is inserted into the socket located under the sliding door on the top of the Pod. Eight configuration switches, also located under the sliding cover, must be set to configure the Pod to the specific UUT being used.

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

The Pod consists of a pair of printed circuit board assemblies mounted within a break-resistant case. The Pod contains a 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 UUTs microprocessor to operate with the UUT.

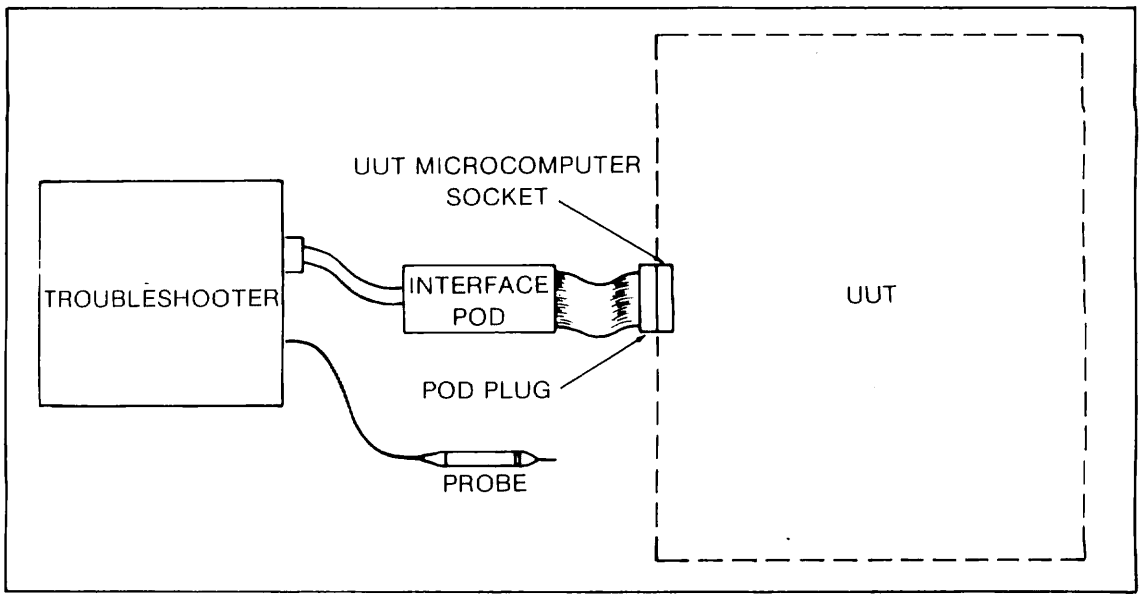


Figure **1-1. Communication Between the Troubleshooter, the Pod, and the UUT.**

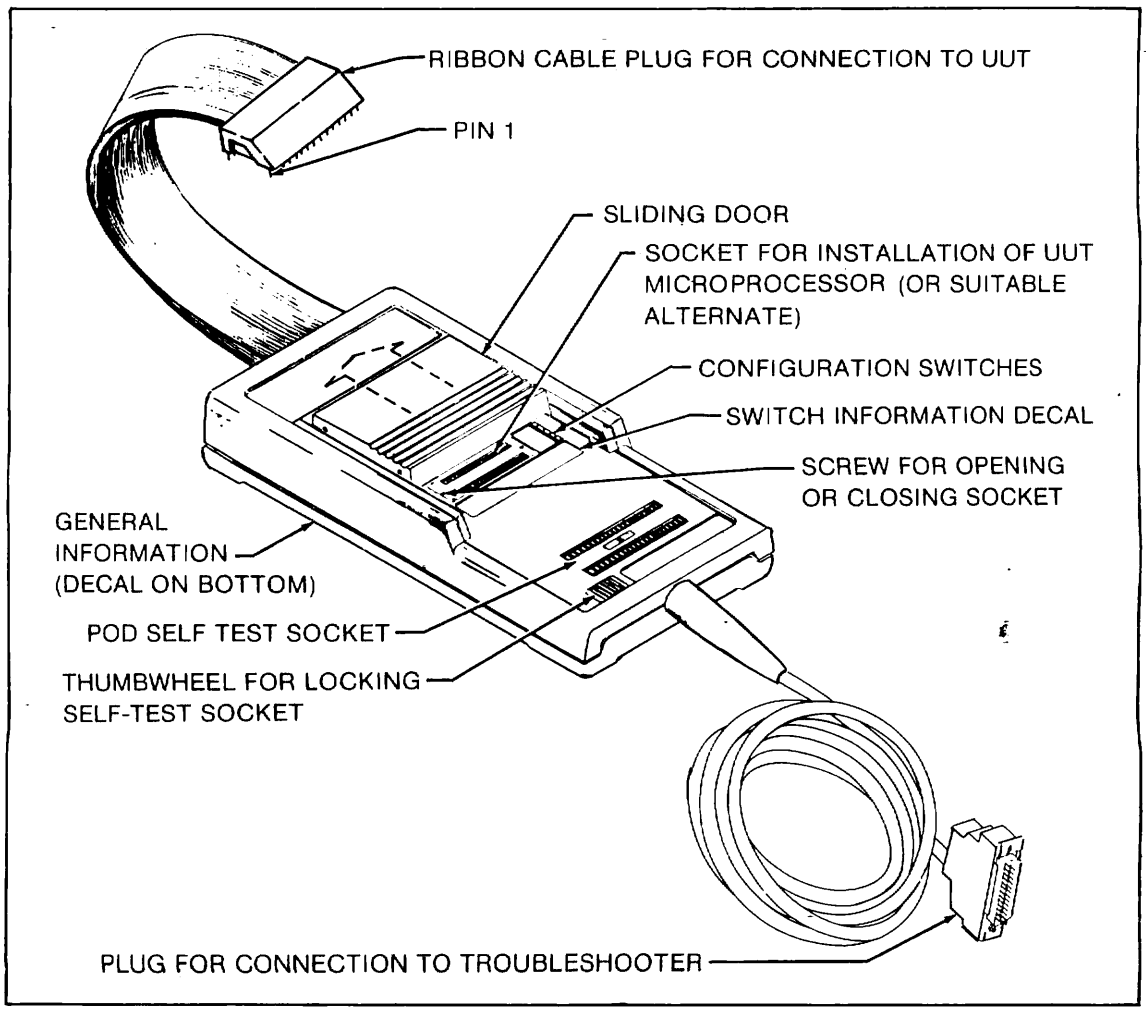


Figure 1-2. External Features of the 8051 Interface Pod.

The Troubleshooter supplies operating power 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.

Logic level detection circuits are provided on each line to the UUT to detect 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.

Over-voltage protection circuits guard against voltages of +12 to -7 volts 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. If UUT power rises above or drops below an acceptable level the Pod notifies the Troubleshooter of the power-fail condition.

A self-test socket on the Pod enables the Troubleshooter to check Pod operation. The self-test socket is a 40-pin zero-insertion-force type socket. The ribbon-cable plug is 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 pins.

SPECIFICATIONS 1-3.

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

Table 1-1. 9000A-8051 Pod Specifications

ELECTRICAL PERFORMANCE	
Power Dissipation	6 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	-0.3V 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 μa
Tristate Output Leakage Current	+-0.02 mA typical, +0.1 to -0.2 mA max.
Input Current	
I_{IH} (RST)	500 μA max.
I_{IH}	40 μA max.
I_{IL}	-800 μA max.

Table 1-1. 9000A-8051 Pod Specifications (cont)

TIMING CHARACTERISTICS *

Maximum External Clock Frequency	12.0 MHz
Insertion Delays to 8051 Signals	
Input signals	12 ns typical
Output signals	
Low Addr, ALE, $\overline{PSEN}$, data	15 ns typical
High Addr, $\overline{RD}$, $\overline{WR}$	20 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< +3.5V

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	-40°C to +70°C, RH < 95% non-condensing
STORAGE	0°C to +40°C, RH < 95% non-condensing
OPERATING	+40°C to +50°C, RH < 75% non-condensing

* Signals are specified as they appear at the ribbon cable plug pins

** Pod outputs set to high-impedance or high logic level.

COMPATIBLE MICROPROCESSORS 1-4.

The 8044- and 8051-family microprocessors that can be used with the 9000A-8051 Pod are listed in Table 1-2. The Pod will work with all family members, with the following operating restrictions:

- The programming modes of the EPROM versions, the 8751 and the 8744, are not supported. The Pod only tests UUTs in the normal operating modes.

- Microprocessors with internal ROM or EPROM that have the security bit set will not execute instructions from external memory, which is needed to perform all Troubleshooter functions. A special version of the programmed microprocessor, without the security bit set, must be used if the RUN UUT function is needed. Otherwise, all testing should be done using the 8031 microprocessor that is shipped with the Pod.

Table 1-2. Products Supported by the 9000A-8051 Interface Pod

8051 FAMILY	8044 FAMILY
8031* 8032 8051* 8052 8751	8344 8044 8744
*Including 80C31 and 80C51	

Section 2
Installation and Self Test

INTRODUCTION 2-1.

The procedures for connecting the Pod to the Troubleshooter and the UUT, for setting the Mode Switches on the top of the Pod, and for performing the Pod Self Test are given in the following paragraphs.

INSTALLING A MICROPROCESSOR INTO THE POD 2-2.

A microprocessor must be installed in the Pod before using it to test a UUT.

NOTE

An 8031 microprocessor rated for 12 MHz operation is supplied with the Pod. You will need to replace it if your application requires a different microprocessor, or if your UUT uses internal ROM.

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

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 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 microprocessor socket. Open the Pod socket by using a screwdriver to turn the screw at the end of the socket. Turn the screw counterclockwise to open the socket. Insert the microprocessor into the socket, aligning pin 1 to the marked position. Turn the screw clockwise to close the socket.

4. Set the configuration switches to the required mode as described below under Configuration. Close the sliding door.

CONNECTING THE POD TO THE UUT 2-3.

The 8044- and 8051-families of microprocessors offer a wide variety of operating modes. The Pod must be configured by the operator to match some of these modes prior to testing.

There are eight configuration switches, located on the top of the Pod under the sliding cover (see Figure 2-1). Table 2-1 and a label on the Pod show the switch positions. A brief description of each switch function follows:

- Switch 1, the Port 0 Mode switch, specifies whether the microprocessor's port 0 is used as an I/O port or as a multiplexed address/data bus.
- Switch 2, the Port 2 Mode switch, specifies whether the microprocessor's port 2 is used as an I/O port or as the upper byte of the address bus.
- Switches 3 and 4 are the Port 3 Mode switches. Switch 3 specifies that bit 2 of port 3 is used either as an I/O port or as an interrupt input ($\overline{INT0}$). Switch 4 specifies that bit 3 will be either an I/O port or an interrupt input ($\overline{INT1}$).
- Switch 5 specifies that bits 6 and 7 of Port 3 will be either I/O port lines or the $\overline{WR}$ and $\overline{RD}$ lines respectively.
- Switch 6, the UUT Connection switch, selects whether the UUT will be tested with the Pod plug inserted into the microprocessor socket or with the Pod plug attached to the microprocessor via a clip-on connector (as would be necessary with a soldered-in micro-processor). When set to the Clip-On position, the Pod will attempt to pull the RST line high to hold the microprocessor in a Reset condition during testing.

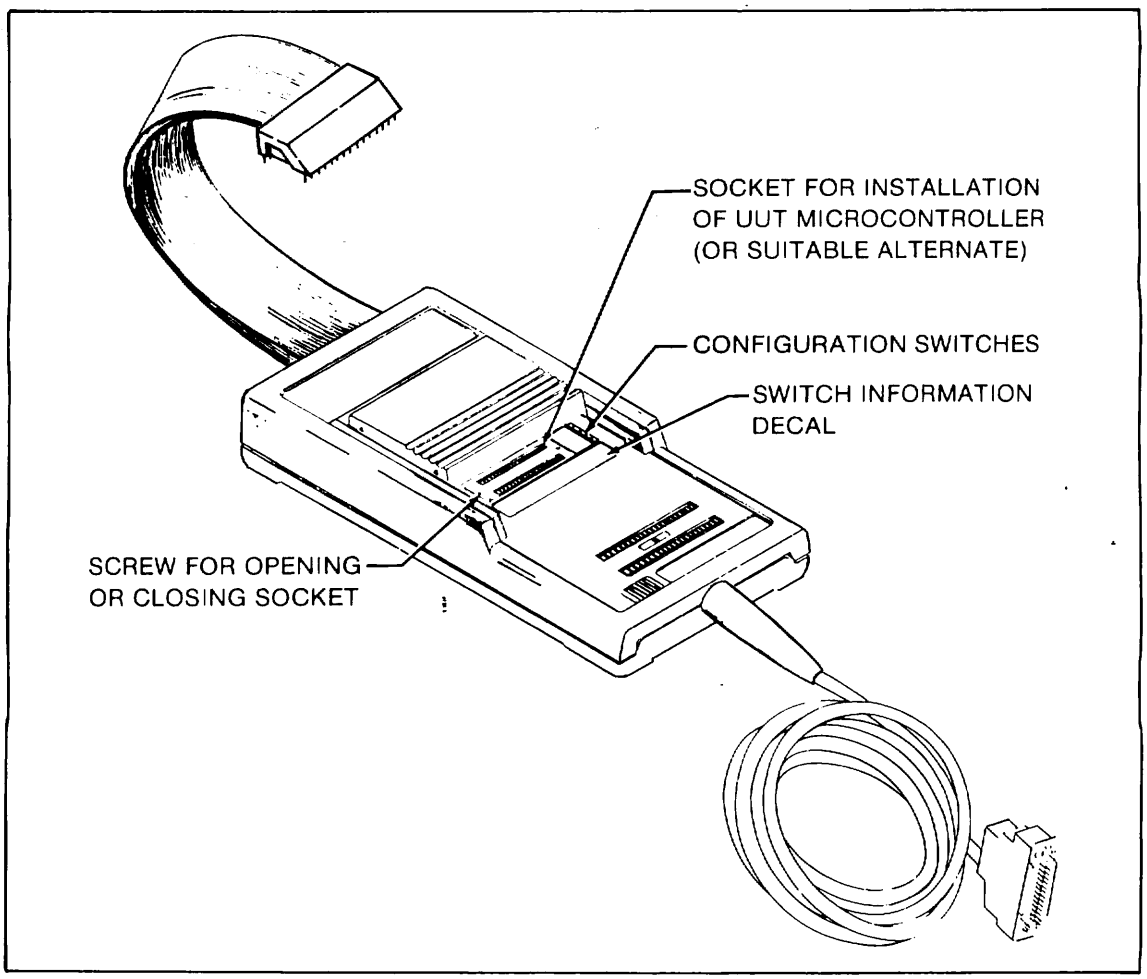


Figure 2-1. Location of Microcontroller Socket and Configuration Switches

Table 2-1. Mode Switch Settings

SWITCH FUNCTION		
1 Port 0	I/O	Addr/Data
2 Port 2	I/O	Address
3 P3.2	I/O	$\overline{INT0}$
4 P3.3	I/O	$\overline{INT1}$
5 P3.6 and P3.7	I/O	$\overline{WR}$ and $\overline{RD}$
6 UUT Connection	Socket	Clip-on
7 Clk Source	Xtal	TTL
8 External Clock	Xtal1	Xtal2

- Switch 7, the Clock Source switch, selects between a TTL clock source and a crystal clock source.

- Switch 8, the External Clock switch. When a TTL clock source is specified by switch 7, this switch is set to specify to which input the clock source will be applied.

Switches 1 through 5 may be overridden by commands from the Troubleshooter. Refer to Special Functions of the 8051 Pod in Section 4 for a description of the Special Addresses which may be used to override the switch positions.

PERFORMING THE POD SELF TEST 2-4.

In order to perform the Pod self test, do the following steps:

1. Remove power from the UUT and the Troubleshooter.
2. Use the thumbwheel to open the pins of the Pod self test socket. Insert the ribbon-cable plug into the socket and use the thumbwheel to close the socket.
3. If it is not already connected, use the round shielded cable to connect the Pod to the Troubleshooter at the location shown in Figure 2-2. Secure the connector using the sliding collar.

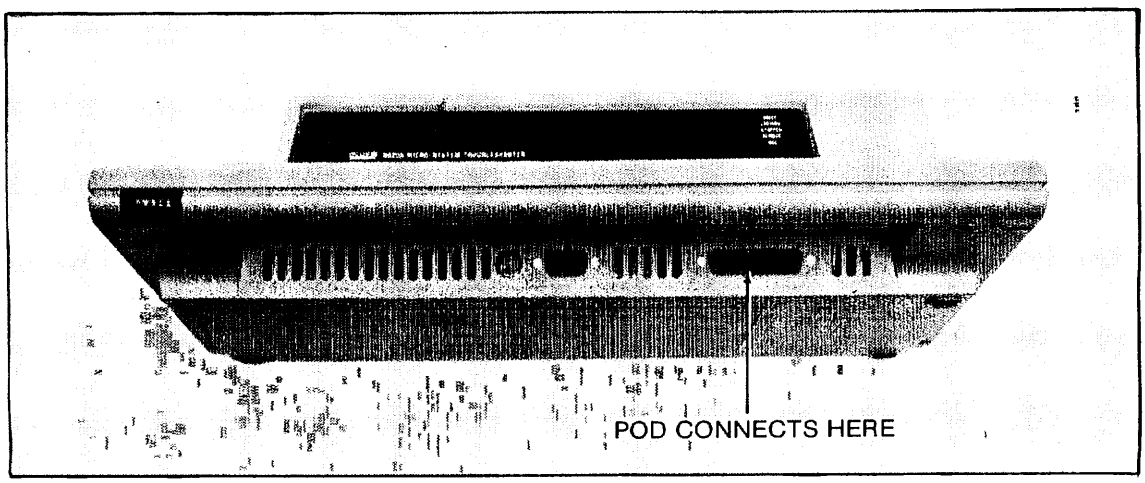


Figure 2-2. Connection of Interface Pod to Troubleshooter

4. Apply power to the Troubleshooter.

5. Press the BUS TEST key to initiate the Pod self test.

If the Troubleshooter displays the message *POD SELF TEST 80xx OK*, then the Pod is operating properly.

NOTE

The Pod name that is displayed in the self test message corresponds to the mode as determined by the switches on the top of the Pod. It is not effected by the type of processor that in contained in the Pod at the time.

If the Troubleshooter displays the message *POD SELF TEST 80xx FAIL xx*, the Pod may not be operating properly. (The letters xx correspond to a failure code describing the error; the failure codes are listed in Section 6.) If the UUT microprocessor has been installed in the Pod, the UUT microprocessor may be causing the failure. Replace the UUT microprocessor with a suitable alternate and try the self test again. Make sure the microprocessor is properly positioned in the microprocessor socket before trying the test.

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

INITIALIZATION 2-5.

When power is applied to the Pod, it sends certain default information to the Troubleshooter, some of which is based on settings of the Configuration switches described above. Most of the information is used to set up the Troubleshooter for this particular Pod, such as the width of the data bus and names of enablable lines. The Configuration switches for Port 0, Port 2, and Port 3 determine what external memory will be used by default with the Troubleshooter's Learn function. Table 2-2 shows the switch settings and the resultant default external Address Blocks and BUS TEST Address.

Table 2-2. Learnable External Address Blocks

CONFIGURATION SWITCHES			LEARNABLE EXTERNAL ADDRESS BLOCKS ENABLED	BUSTEST ADDRESS	NAME DISPLAY
P0	P2	P3.6&3.7			
I/O	x	x	None	30000	8051
A/D	A	x	0000 0000 through 0000 FFFF	--	--
A/D	A	$\overline{RD}/\overline{WR}$	0001 0000 through 0001 FFFF	10000	8031
A/D	x	$\overline{RD}/\overline{WR}$	0002 0000 through 0002 00FF	20000	8051X

A/D = Address/Data
A = Address
x = Don't care

Section 3
Microprocessor Data

INTRODUCTION 3-1.

This section contains microprocessor data which may be useful during operation of the Troubleshooter. This information includes descriptions of 8044- and 8051-family signals and pin assignments.

MICROPROCESSOR SIGNALS 3-2.

Table 3-1 lists all of the 8044- and 8051-family microprocessor signals and provides a brief description of each signal. Refer to the microprocessor manufacturer's literature for complete information.

Figure 3-1 shows the 8044- and 8051-family pin assignments.

Table 3-1. Signal Descriptions

SIGNAL NAME	DESCRIPTION
ALE/$\overline{\text{PROG}}$	Address Latch Enable/Program. Latches addresses into external memory. The Program pulse for programming microprocessors with internal EPROM is applied to this pin. The Pod does not test microprocessors in the program mode.
$\overline{\text{EA}}$/VPP	When $\overline{\text{EA}}$/VPP is low, all instructions come from External Addresses. When high, some addresses (less than 0FFF in the 8051 series and less than 1FFF in the 8052 series) execute from internal ROM. Also receives the programming pulse for EPROM versions of the microprocessor. The Pod does not test microprocessors in the program mode.
PORT 0 (P0.0-P0.7)	Port 0 is an eight-bit, open-drain I/O port. When using external memory, it is multiplexed between low-order address and data. During program verification, Port 0 is used for data output. The Pod does not test microprocessors in the program mode.

Table 3-1. Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
PORT 1 (P1.0-P1.7)	Port 1 is an eight-bit I/O port. Two of Port 1's pins are used for secondary purposes in some microprocessors as described below.
T2 (P1.0)	Trigger input to timer 2. (8032 and 8052 only.)
T2EX (P1.1)	External input to timer 2. (8032 and 8052 only.)
$\overline{RTS}$ (P1.6)	Request To Send signals that the microprocessor is ready to transmit data (in a non-looping configuration). (8044 family.)
$\overline{CTS}$ (P1.7)	Clear To Send signals to the microprocessor that the receiving station is ready to send data (in a non-looping configuration). (8044 family.)
	Port 1 is also used as the low-order addresses during program verification. The Pod does not test microprocessors in the program mode.
PORT 2 (P2.0-P2.7)	Port 2 is an eight-bit I/O port that is used for high-order addresses when using external memory, and high-order addresses and control information during program verification.
PORT 3 (P3.0-P3.7)	Port 3 is an eight-bit I/O port. Pins assigned to Port 3 also have secondary functions as described below.
RXD (P3.0) I/$\overline{O}$	RXD is the serial port's receiver data input during asynchronous communications, and its data input/output during synchronous communications (8051 family). Also functions as the data direction control in non-looping configurations (8044 family).
TXD (P3.1) DATA	TXD is the serial port's transmitter data output during asynchronous communications, and its clock output during synchronous communications (8051 family). Also functions as the data input/output in non-looping configurations (8044 family).
$\overline{INT0}$ (P3.2)	$\overline{INT0}$ is the interrupt 0 input or the gate control input for counter 0.
$\overline{INT1}$ (P3.3)	$\overline{INT1}$ is the interrupt 1 input or the gate control input for counter 1.
T0 (P3.4)	T0 is the input to counter 0.
T1 (P3.5)	T1 is the input to counter 1.
$\overline{WR}$ (P3.6)	$\overline{WR}$ (Write Control) latches the data from Port 0 into external memory.
$\overline{RD}$ (P3.7)	$\overline{RD}$ (Read Control) enables external data memory for Port 0.
$\overline{PSEN}$	Program Store Enable enables external Program Memory on the bus. $\overline{PSEN}$ is high during internal operation.
RST	The Reset signal. A high on this line resets the microprocessor.

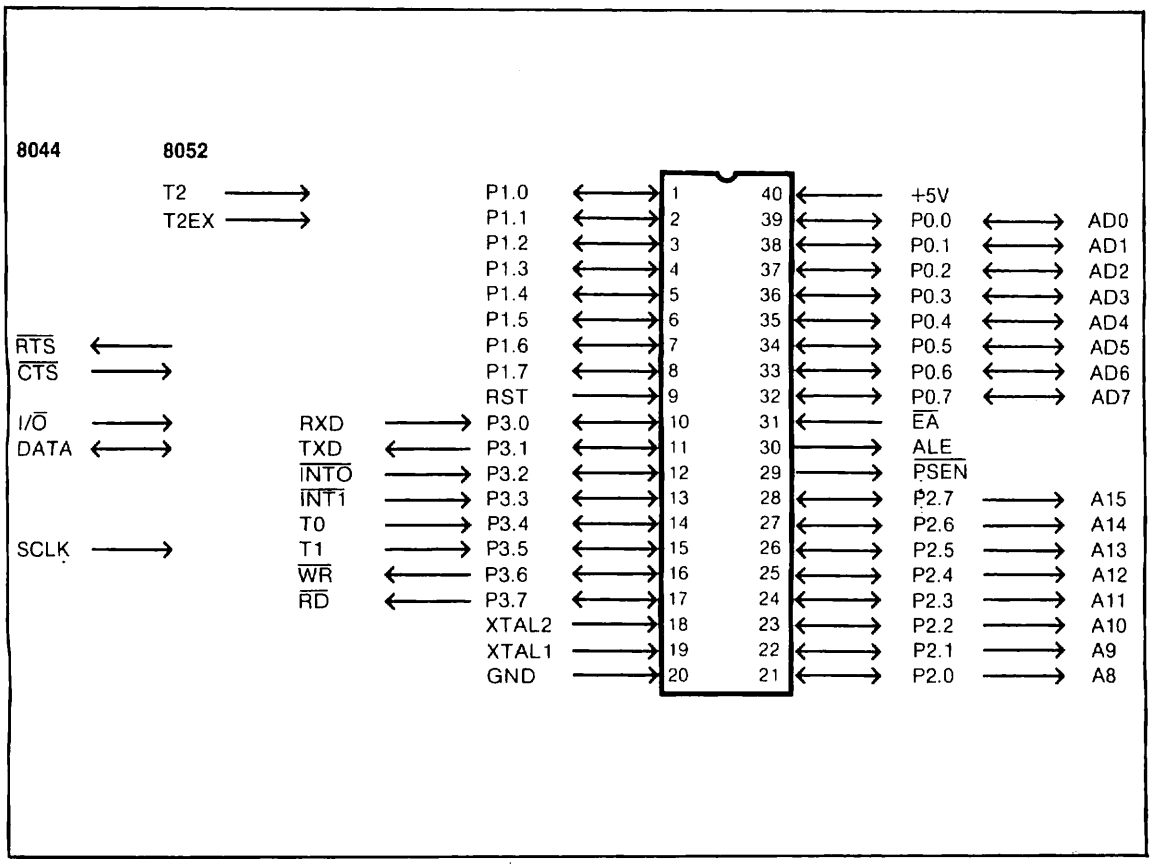


Figure 3-1. 8044 and 8051 Family Pin Assignments

Section 4
Operating Information

INTRODUCTION 4-1.

This section contains information which pertains to operating the Troubleshooter with systems employing the 8044- or 8051-microprocessor families. Information provided is in addition to and complements that provided in the Troubleshooter Operator and Programming manuals, and covers such items as the following:

- Definition and bit assignment of status lines.
- Definition of forcing and interrupt lines.
- Bit assignment of control lines.
- Address space assignment.
- Access to program memory, RAM, registers, and ports.
- Characteristics of Learn and Bus Test operations.
- Marginal UUT problems.
- Quick-Looping Read and Write functions.
- The Quick RAM test.
- The Quick ROM test.
- The Block Memory tests.

STATUS/CONTROL LINES 4-2.

Introduction 4-3.

The Pod provides the interface between the general architecture of the Troubleshooter and the specific features of the microprocessor. Part of this task is assigning the status and control lines to a consistent bit map to be used in status messages and read/write operations.

These assignments are described in the following paragraphs, and are summarized in Tables 4-1 and 4-2 and on a decal on the bottom of the Pod.

Table 4-1. Status Line Bit Assignments

BIT	8051 INPUT	DESCRIPTION
7	PWR FAIL	UUT Power Fail
6 *	MPU FAIL	UUT Processor RAM Fail
5	$\overline{EA}$	External Address Line
4 *	RST	Reset Line
3 **	$\overline{INT0}$ (P3.2)	Interrupt 0 Line
2 **	$\overline{INT1}$ (P3.3)	Interrupt 1 Line
1	-	
0	-	

* Forcing Line
** Defined by DIP Switch Setting

Table 4-2. Control Line Bit Assignments

BIT	8051 OUTPUT	DESCRIPTION
7	$\overline{PSEN}$	Program Store Enable
6	-	
5 **	$\overline{WR}$ (P3.6)	External Data Write
4 **	$\overline{RD}$ (P3.7)	External Data Read
3	-	
2	-	
1	-	
0	ALE	Address Latch Enable

** Defined by DIP Switch Setting

Status Line Bit Assignments 4-4.

The bits assigned to the specific status lines are shown in Table 4-1 and on the decal on the back of the Pod. A Read Status operation reads the actual levels on the pins or bits. However an active forcing line is reported as a high bit, regardless of its actual logic level. For example, if the Reset line is active (high), an active Reset is reported as *ACTIVE FORCE LINE, STS BTS 0001 0000.* An $\overline{INT0}$ interrupt (pin 12, active low), on the other hand, if the Troubleshooter's active interrupt trap is enabled, is reported as *ACTIVE INTERRUPT, STS BTS 0000 1000,* but a Read Status operation will report XXXX 0100.

NOTE

The Power Fail and Microprocessor Fail bits do not represent particular 8044 or 8051 signals. PWR FAIL is generated within the Pod whenever UUT voltage levels deviate from acceptable limits. MPU FAIL is generated whenever the microprocessor's internal RAM fails a RAM test which is conducted during every Pod reset cycle. MPU FAIL remains in effect until cleared by a successful Pod Reset.

Control Line Bit Assignments 4-5.

The Pod does not allow the Troubleshooter to write directly to any of the microprocessor control lines with the Write Control or Data Toggle Control functions. However, all 8044 and 8051 control lines are tested by means of the Troubleshooter Bus Test.

If a control line cannot be driven when performing a Bus Test, the Troubleshooter displays the message *CTL ERR XXXXXXXX-LOOP?* The string *XXXXXXXX* represents a binary map that identifies which lines can or cannot be driven. A 1 indicates the corresponding line cannot be driven, while a 0 indicates the corresponding line can be driven. Control line bit assignments are shown in Table 4-2 and on the decal on the back of the Pod.

For example, if the $\overline{\text{PSEN}}$ line of an 8051 cannot be driven, but all other control lines can be driven, the Troubleshooter displays the message *CTL ERR 10000000 LOOP?* (the $\overline{\text{PSEN}}$ line is represented by bit number 7).

FORCING AND INTERRUPT LINES 4-6.

Several Troubleshooter messages are used to indicate errors and conditions associated with forcing lines and interrupts. Forcing lines are those lines which, when active, force the microprocessor into some specific action. Forcing lines for the 8044- and 8051-family are RST and MPU FAIL.

The RST and interrupt lines are functionally enabled only when the Troubleshooter is operated in the RUN UUT mode (where the Pod emulates the UUT microprocessor). In all other operating modes, the Troubleshooter monitors the status lines and, when active, causes active forcing or interrupt line messages to appear on the Troubleshooter display.

NOTE

The RST line is active high. The interrupt lines are active low.

MPU FAIL denotes a failed test on the microprocessor's internal RAM. The RAM is tested during each Pod Reset cycle, and failures are reported as a forcing line error. Microprocessor RAM failures are most likely due to a defective microprocessor. When a MPU FAIL forcing line error is reported, try to correct the error by substituting the microprocessor that is plugged into the Pod's microprocessor socket for a known-good device. If a MPU FAIL forcing line error persists, refer to Troubleshooting a Defective Pod in Section 6.

The reporting of active forcing lines or active interrupt lines may be disabled with the Setup operation on the Troubleshooter. If the Reset line is driven high but the *TRAP ACTIVE FORCE LINE* is disabled, for example, the Pod reports such a condition to the Troubleshooter, but the Troubleshooter does not report the condition to the operator.

ADDRESS SPACE ASSIGNMENT 4-7.

Introduction 4-8.

The 8044- and 8051-family microprocessors are capable of addressing some or all of the memory and I/O functions listed and described briefly in Table 4-3. Additional information about access to the various address space is provided below. Address space assignments are summarized in Table 4-3.

Table 4-3. Address Space Assignments

LOWER ADDRESS	UPPER ADDRESS	DESCRIPTION OF SPACE
00 0000	00 FFFF	External program memory
01 0000	01 FFFF	External data memory (16-bit address)
02 0000	02 00FF	External data memory (8-bit address)
03 0000	03 00FF	Internal data RAM
04 0080	04 00FF	Internal Special Function Registers
05 0000	05 00FF	Int. bit (flags & spcl func) addresses
06 0000	06 FFFF	Internal program ROM
07 0000	07 FFFF	* Pod's program ROM
08 0000	08 FFFF	* Pod's external read/write space
09 0000	09 00FF	* Pod's variable RAM (int. to processor)
10 0000	10 FFFF	Quick looping ext. program ROM
11 0000	11 FFFF	Quick looping ext. RAM (16-bit address)
12 0000	12 00FF	Quick looping ext. RAM (8-bit addresss)
2X 0000	2X XXXX	Quick RAM test
3X 0000	3X XXXX	Quick ROM test
4X 0000	4X XXXX	Block write and verify
F0 0000	F0 00XX	Special Addresses

* Enabled only when self test bit is disabled, for troubleshooting Pod.

External Memory 4-9.

The external program memory address space (00 0000 - 00 FFFF) is an address space that normally only consists of Read-Only Memory. Read operations in this space exercise the $\overline{PSEN}$ control line. An attempted Write operation in this address space will yield data drivability errors for all data lines.

The 64K byte RAM address space (01 0000 - 01 FFFF) is used for variable data and memory-mapped I/O devices. Read or write operations in this area will exercise the $\overline{RD}$ or $\overline{WR}$ control lines. The high-order address is on Port 2.

The 256 byte RAM address space (02 000 - 02 00FF) is identical to the 64K byte space above, except that only the lower eight bits of the address are sent to the UUT. Port 2 is used for I/O data. For example, a *WRITE@ 1 7FFF=AA* is the same as writing Port 2 to 7F (*WRITE @ 4 00A0=7F*), and then a *WRITE @ 2 00FF=AA*. The only difference is that port 2 is in I/O mode rather than Address mode.

Internal RAM, Registers, and Stack 4-10.

The microprocessor's internal RAM address space (03 000 - 03 00FF) may be used to load the microprocessor's RAM prior to a RUN UUT command.

Read and write operations in the address range of 03 0000 to 03 007F actually use the Pod's RAM, instead of the RAM internal to the microprocessor. Read and write operations in the address range of 03 0080 to 03 00FF use the UUT microprocessor's internal RAM (although only the 8032 and 8052 contain RAM in those locations). All of the data in this 03 XXXX address space is copied into the microprocessor's internal RAM (if applicable) at the beginning of a RUN UUT command, and transferred back to the Pod's RAM when the RUN UUT terminates.

The internal Special Function Register address space (04 0080 - 04 00FF) contains special function registers for the microprocessor's internal peripheral devices, control, and status. Since writing to some of the registers in this address range might affect Pod operation, that data is set aside until a RUN UUT command is executed and then it is loaded into the correct special function register. Note too that only registers whose actions are harmless to the Pod are writable in this address range. The xx80 - xxFF addressing scheme corresponds to the standard 8044- and 8051-family addresses as defined by the microprocessor's manufacturer.

Table 4-4 contains a list of the special function registers.

The internal bit address space (05 0000 - 05 FFFF) contains bit addresses for special function registers, internal data RAM, and individual port lines. Bits addressed in this space are mapped into specific bits in the 04 0000 - 04 00FF address spaces according to the manufacturer's bit addressing scheme. As with the 04 00xx address range, only some bits are actually written during a Write operation. Others are deferred and written just prior to beginning a RUN UUT command.

NOTE

If a UUT has a port with both input and output pins, then the bit mode (5 XXXX addresses) should be used rather than the byte mode (4 XXXX). Then, only the bit addressed will be tested for drivablility.

Table 4-5 shows the addresses for individual port lines. Table 4-6 shows the special addresses and bit assignments for internal data.

Internal Program ROM 4-11.

The Internal Program ROM address space (06 0000 - 06 FFFF) makes use of an undocumented feature of the 8051 to provide access to the internal program data of the microprocessor. By pulling the $\overline{EA}$ line high, it allows the Pod to read the contents of the microprocessor ROM. It is provided for the convenience of those who can use it, but with the understanding that it might not always work. An attempted Write operation to this area will yield data drivability errors on all data lines.

Diagnostic Addresses 4-12.

The Pod's program ROM (07 0000 - 07 FFFF), the Pod external read/write space (08 0000 - 08 FFFF), and the Pod microprocessor's internal RAM (09 0000 - 09 00FF) are accessible when the self test is disabled for troubleshooting a defective Pod. Refer to Section 6, Troubleshooting, for detailed information about using these addresses.

Quick-Looping Test Addresses 4-13.

Addresses in the 10 0000 to 12 00FF range are the same as 00 0000 to 02 00FF, except that Read or Write commands repeat until another command is received. A system fault byte is sent after the first repetition of the operation, and sync pulses are generated on each repetition, if they are enabled.

Operations to addresses 10 0000 - 10 FFFF perform quick-looping functions on the UUT's external program ROM at 00 0000 through 00 FFFF.

Operations to addresses 11 0000 - 11 FFFF perform quick-looping functions on the UUT's external (16-bit address) data space at 01 0000 through 01 FFFF.

Operations to addresses 12 0000 - 12 00FF perform quick-looping functions on the UUT's external data space (8-bit address) at 02 0000 through 02 00FF.

Table 4-4. Special Function Registers

NAME	ADDRESS	ACTION
P0	04 0080	Port 0
SP	04 0081	+Stack Pointer
DPL	04 0082	+Data Pointer (Low Byte)
DPH	04 0083	+Data Pointer (High Byte)
PCON	04 0087	◊ +Power Control
TCON	04 0088	Timer/Counter Control
TMOD	04 0089	Timer/Counter Mode Control
TL0	04 008A	Timer/Counter 0 (Low Byte)
TL1	04 008B	Timer/Counter 1 (Low Byte)
TH0	04 008C	Timer/Counter 0 (High Byte)
TH1	04 008D	Timer/Counter 1 (High Byte)
P1	04 0090	Port 1
SCON	04 0098	Serial Control
SBUF	04 0099	Serial Data Buffer
P2	04 00A0	Port 2
IE	04 00A8	+Interrupt Enable Control
P3	04 00B0	Port 3
IP	04 00B8	Interrupt Priority Control
T2CON	04 00C8 *	Timer/Counter 2 Control
STS	04 00C8 **	Status/Command
SMD	04 00C9 **	Serial Mode
RCAP2L	04 00CA *	Timer/Counter 2 Capture Register (Low Byte)
RCB	04 00CA **	Received Control Byte
RCAP2H	04 00CB *	Timer/Counter 2 Capture Register (High Byte)
RBL	04 00CB **	Receive Buffer Length
TL2	04 00CC *	Timer/Counter 2 (Low Byte)
RBS	04 00CC **	Receive Buffer Start Address
TH2	04 00CD *	Timer/Counter 2 (High Byte)
RFL	04 00CD **	Received Field Length
STAD	04 00CE **	Station Address
DMACNT	04 00CF **	DMA Count
PSW	04 00D0	+Program Status Word
NSNR	04 00D8 **	Send/Receive Count
SIUST	04 00D9 **	SIU State Counter
TCB	04 00DA **	Transmit Control Byte
TBL	04 00DB **	Transmit Buffer Length
TBS	04 00DC **	Transmit Buffer Start Address
FIFO1	04 00DD **	FIFO (Byte 1)
FIFO2	04 00DE **	FIFO (Byte 2)
FIFO3	04 00DF **	FIFO (Byte 3)
ACC	04 00E0	+Accumulator (A Register)
B	04 00F0	B Register

* unique to 8052/8032

** unique to 8044/8344/8744

+ Data written to these registers is not actually entered into that register until a RUN UUT operation is performed.

◊ The PD (Powerdown) and IDL (Idle Mode) control bits may not be accessed from the Troubleshooter.

Table 4-5. Port Access Addresses

PORT	BYTE	BIT ADDRESS	PORT	BYTE	BIT ADDRESS
Port 0	4 0080		Port 2	4 00A0	
0.0		50080	2.0		500A0
0.1		50081	2.1		500A1
0.2		50082	2.2		500A2
0.3		50083	2.3		500A3
0.4		50084	2.4		500A4
0.5		50085	2.5		500A5
0.6		50086	2.6		500A6
0.7		50087	2.7		500A7
Port 1	4 0090		Port 3	4 00B0	
1.0		50090	3.0		500B0
1.1		50091	3.1		500B1
1.2		50092	3.2		500B2
1.3		50093	3.3		500B3
1.4		50094	3.4		500B4
1.5		50095	3.5		500B5
1.6		50096	3.6		500B6
1.7		50097	3.7		500B7

For example, a *READ @ 11 5528* command initiates a Read function at address 01 5528, then continues repeating that Read operation until another command is received.

Quick ROM and RAM Test Addresses 4-14.

The Quick RAM Test addresses (2X 0000 - 2X FFFF) and the Quick ROM Test addresses (3X 0000 - 3X FFFF) are used to control the Quick ROM and RAM Tests described below.

Block Write and Verify Addresses 4-15.

The Block Write and Verify addresses (4X 0000 - 4X FFFF) are used to fill blocks of memory with data and verify the contents. See Block Memory Tests.

NOTE

Illegal addresses are mapped into the 1 XXXX or 3 XXXX address space. After reporting the error, the operation is performed using the new address.

Special Addresses 4-16.

These addresses (F0 0000 - F0 FFFF) control Pod functions described later under Special Pod Functions.

QUICK-LOOPING READ AND WRITE FUNCTIONS 4-17.

The 8051 Pod provides Quick-Looping read and write functions with repetition rates which are considerably faster than the repetition rate of the ordinary Troubleshooter Looping function that is selected by pressing the LOOP key. Because of the increased repetition rate, the Quick Looping functions are particularly useful for enhanced viewing of signal traces on an oscilloscope that is synchronized to the TRIGGER OUTPUT pulse (available on the rear panel of the Troubleshooter).

Table 4-6. Special Function Bit Addresses

MNEMONIC	ADDRESS	SPECIAL FUNCTION
	05 0000 - 05 007F	Boolean variables
P0.0-7	05 0080 - 05 0087	Port 0
TCON.0-7	05 0088 - 05 008F	Timer Control Register
IT0	05 0088	Interrupt 0 Type Control Bit
IE0	05 0089	Interrupt 0 Edge Flag
IT1	05 008A	Interrupt 1 Type Control Bit
IE1	05 008B	Interrupt 1 Edge Flag
TR0	05 008C	Timer 0 Run Control Bit
TF0	05 008D	Timer 0 Overflow Flag
TR1	05 008E	Timer 1 Run Control Bit
TF1	05 008F	Timer 1 Overflow Flag
P1.0-7	05 0090 - 05 0097	Port 1
SMOD.0-7	05 0098 - 05 009F	Serial Mode Register
RI	05 0098	Receive Interrupt Flag
TI	05 0099	Transmit Interrupt Flag
RB8	05 009A	Receive Bit 8
TB8	05 009B	Transmit Bit 8
REN	05 009C	Receive Enable
SM2	05 009D	Serial Mode Control Bit 2
SM1	05 009E	Serial Mode Control Bit 1
SM0	05 009F	Serial Mode Control Bit 0
P2.0-7	05 00A0 - 05 00A7	Port 2
IE.0-7	05 00A8 - 05 00AF	Interrupt Enable Register
EX0	05 00A8	Enable External Interrupt 0
ET0	05 00A9	Enable Timer 0 Interrupt
EX1	05 00AA	Enable External Interrupt 1
ET1	05 00AB	Enable Timer 1 Interrupt
ES	05 00AC	Enable Serial Port Interrupt
ET2	05 00AD	Enable Timer 2 Interrupt (8052)
EA	05 00AF	Enable All Interrupts
P3.0-7	05 00B0 - 05 00B7	Port 3
RXD	05 00B0	Serial Port Receive Pin
TXD	05 00B1	Serial Port Transmit Pin
$\overline{INT0}$	05 00B2	Interrupt 0 Input Pin
$\overline{INT1}$	05 00B3	Interrupt 1 Input Pin
T0	05 00B4	Timer/Counter 0 External Flag
T1	05 00B5	Timer/Counter 1 External Flag
$\overline{WR}$	05 00B6	Write Pin for External Memory
$\overline{RD}$	05 00B7	Read Pin for External Memory

Table 4-6. Special Function Bit Address (cont)

MNEMONIC ADDRESS		SPECIAL FUNCTION
IP.0-7	05 00B8 - 05 00BF	Interrupt Priority Register
PX0	05 00B8	Priority of Ext. Interrupt 0
PT0	05 00B9	Priority of Timer 0 Interrupt
PX1	05 00BA	Priority of Ext. Interrupt 1
PT1	05 00BB	Priority of Timer 1 Interrupt
PS	05 00BC	Priority of Serial Port Int.
PT2	05 00BD	Priority of Timer 2 int. (8052)
T2CON	05 00C8 - 05 00CF	Timer 2 Control Register (8052)
CP/$\overline{RL2}$	05 00C8	Capture/Reload Flag
C/$\overline{T2}$	05 00C9	Timer or Counter Select 2
TR2	05 00CA	Start/Stop Control 2
EXEN2	05 00CB	Timer 2 External Enable Flag
TCLK	05 00CC	Transmit Clock Flag
RCLK	05 00CD	Receive Clock Flag
EXF2	05 00CE	Timer 2 External Flag
TF2	05 00CF	Timer 2 Overflow Flag
STS.0-7	05 00C8 - 05 00CF	Status Register (8044)
RBP	05 00C8	Receive Buffer Protect
AM	05 00C9	Auto Mode/Addressed Mode
OPB	05 00CA	Optional Poll Bit
BOV	05 00CB	Receive Inf. Buffer Overflow
SI	05 00CC	Serial Interface Unit Interrupt
RTS	05 00CD	Request to Send
RBE	05 00CE	Receive Buffer Empty
TBF	05 00CF	Transmit Buffer Full
PSW.0-7	05 00D0 - 05 00D7	Program Status Word
P	05 00D0	Parity Flag
OV	05 00D2	Overflow Flag
RS0	05 00D3	Register Bank Select Bit 0
RS1	05 00D4	Register Bank Select Bit 1
F0	05 00D5	Flag 0
AC	05 00D6	Auxiliary Carry Flag
CY	05 00D7	Carry Flag
NSNR.0-7	05 00D8 - 05 00DF	Send/Receive Count Reg (8044)
SER	05 00D8	Sequence Error Received
NR0-NR2	05 00D9 - 05 00DB	Receive Sequence Counter
SES	05 00DC	Sequence Error Send
NS0-NS2	05 00DD - 05 00DF	Send Sequence Counter
ACC.0-7	05 00E0 - 05 00E7	Accumulator
B.0-7	05 00F0 - 05 00F7	B Register

Unlike the ordinary Troubleshooter Looping function, the software that controls the Quick Looping functions resides in the Pod and not the Troubleshooter. The operator selects these functions by using the special addresses listed in Table 4-3. For example, a read operation at address 10 0F00 causes the Pod to perform the Quick-Looping read operation at address 00 0F00.

It should be noted that the diagnostics performed by the Pod during the execution of the Quick-Looping read or write operations are less rigorous than the diagnostics performed during the execution of the ordinary Looping function. The Pod reports to the Troubleshooter any UUT system errors detected during the first iteration only. Subsequent UUT system errors are not reported.

If both error reporting and the Quick Looping functions are desired, you may apply the ordinary Troubleshooter Looping function to the Quick-Looping read or write, such as *READ @ 10 2000 LOOP*. The Troubleshooter will command read operations at address 10 2000 at the normal looping speed with full error reporting. For every ordinary read operation, the Pod will interject a few Quick-Looping read operations (with no error reporting) which will enhance oscilloscope viewing.

CAUTION

To prevent possible damage to the probe or the UUT, do not use the probe to generate stimulus pulses while a Quick-Looping function is being performed if the UUT crystal or clock frequency is less than 1 MHz.

The reason for the preceding caution is that the combination of the high repetition rate of a Quick-Looping function and a slow UUT clock (below 1 MHz) greatly increases the duty cycle of the probe. If the duty cycle is excessively high, the probe stimulus pulses can cause damage to the probe or the UUT. Note that the response capability of the probe, such as logic level reading with the Read Probe operation, is unaffected by high duty cycles.

SPECIAL ADDRESSES 4-18.

Addresses in the F0 XXXX range are special addresses, used to control internal Pod functions. The special addresses are described below, and summarized in Table 4-7. Note that some of these address are for Read operation only; attempts to Write data to them will yield drivability errors.

NOTE

Some of the Special Addresses are read-only locations. Attempting to write to them will result in drivability errors.

Self Test Failure Code (F0 0000) 4-19.

This special address contains the result code from the most recent Pod self test. If the Pod passes the self test, this special address contains the value FF. Other values denote specific errors that were detected during the self test program.

For complete information on deciphering the self test failure codes, refer to Section 6. Self Test Failure codes are summarized in Tables 6-2 and 6-3.

Status and Error Reporting Group (F0 0020 - F0 0026) 4-20.

The following group of special addresses contains summaries of errors and status. When reading these addresses, the normal dummy operation is not performed, thereby preserving the accuracy of the values.

Table 4-7. Special Addresses

ADDRESSES	FUNCTION	ADDRESSES	FUNCTION
F0 0000	Self Test Fail Code		
F0 0020	Last Error Summary	F0 0050	Port 0 Bus Test
F0 0022	Last Control Errors	F0 0051	Port 1 Bus Test
F0 0024	Last Force Line Report	F0 0052	Port 2 Bus Test
F0 0026	Last Status	F0 0053	Port 3 Bus Test
F0 0028	Error Summary Mask		
F0 002A	Control Error Mask	F0 0060	Refresh Enable
F0 002C	Force Line Error Mask		
		F0 0062	DIP Switch Override Mask
F0 0040	Last High Address Errors	F0 0064	DIP Switch Override Data
F0 0042	Last Low Address Errors		
F0 0044	Last Data Errors		

LAST ERROR SUMMARY (F0 0020) 4-21.

This special address contains a description of the last encountered error. This information, also referred to as "the fault byte", is returned 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 shown in Figure 4-1.

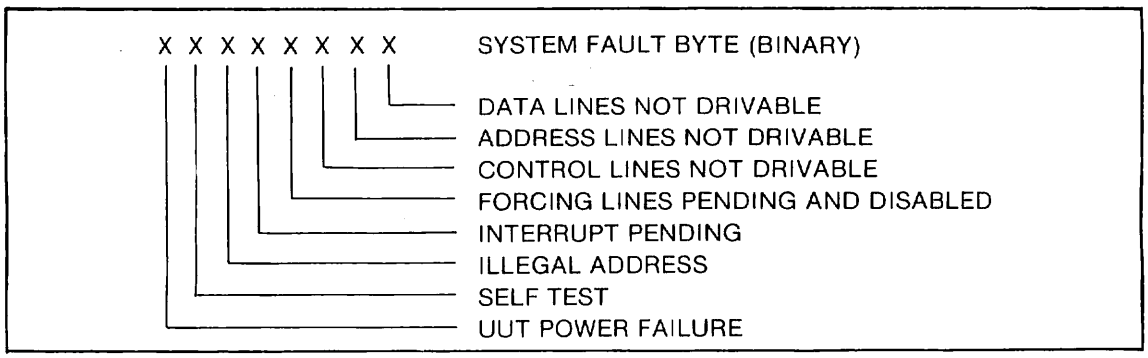


Figure 4-1. System Fault Byte

For example: If the Pod UUT connector is plugged into a UUT, and all error reporting is inhibited, performing a *READ @ F0 0020 = 18* indicates that:

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

LAST CONTROL ERRORS (F0 0022) 4-22.

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

READ @ F0 0022 = 80 OK

shows that the Pod was not able to drive bit 7, the $\overline{\text{PSEN}}$ line.

LAST FORCE LINE REPORT (F0 0024) 4-23.

This special address contains a bit map of any forcing lines which were detected as active during the last UUT operation. Refer to Table 4-1, for bit assignments for the two forcing lines, RST and MPU FAIL.

LAST STATUS (F0 0026) 4-24.

The status word from the immediately previous Pod operation may be read at this address. 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 4-1 or the Pod decal (on the bottom of the Pod) for status line bit assignments.

For example,

READ @ F0 0026 = 08

shows $\overline{\text{INT0}}$ to be the only high level status line. Compare this to the *READ @ STS* example under Status Line Bit Assignments.

Error Reporting Masks (F0 0028 - F0 002C) 4-25.

These masks control the reporting of Error, 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.

ERROR SUMMARY MASK (F0 0028) 4-26.

The reporting of any individual errors in the error summary may be suppressed by setting the appropriate bits of the error summary mask to zero. The bit assignments correspond to the the System Fault Byte shown in Figure 4-1.

Errors corresponding to the suppressed bits will not be reported by the Troubleshooter. The default value is FF--all errors enabled.

Setting the self-test bit (bit 6) to 0 disables the Pod's internal self test and makes the Pod's internal addresses valid spaces. This function is used when troubleshooting a defective Pod.

CONTROL ERROR MASK (F0 002A) 4-27.

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

For example, a certain 8051 UUT may not allow the processor to drive the $\overline{\text{PSEN}}$ line low. If this is considered normal, performing a *WRITE @ F0 002A = 7F* will inhibit the reporting of $\overline{\text{PSEN}}$ line drivability errors during BUS TEST, while allowing drivability error reporting for all of the other control lines.

FORCELINE ERROR MASK (F0 002C) 4-28.

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 4-1, Status Line

Bit Assignments. The complete error summary can be read from the Last Force Line Report special address. Errors corresponding to the suppressed bits will not be reported by the Troubleshooter.

Default value = FF

Address and Data Errors (F0 0040 - F0 0044) 4-29.

This group of special addresses contains maps of drivability errors which were detected during the previous UUT operation. If reporting of these errors has been disabled using the Troubleshooter's Setup commands, reading these special addresses will reveal the source of address and data errors.

A bit set to 1 indicates a drivability error in the corresponding line. A 0 indicates a successful test on that line.

LAST HIGH ADDRESS ERRORS (F0 0040) 4-30.

A map of drivability errors detected in the high address bits during the last UUT operation.

LAST LOW ADDRESS ERRORS (F0 0042) 4-31.

A map of drivability errors detected in the low address bits during the last UUT operation.

LAST DATA ERRORS (F0 0044) 4-32.

A map of drivability errors detected in the data bits during the last UUT operation.

Bus Tests (F0 0050 - F0 0053) 4-33.

These special addresses contain a mask that specifies which pins of a port are to be tested for drivability. Output pins to be tested are specified by writing a mask to the address with the pin(s) set to 0. Input pins are set to 1.

Reading these special addresses causes a drivability test to be performed on the specified output pins. The result returned by the Read operation is a map of any pins which could not be driven properly. A 0 denotes a line which has been determined to be drivable. A 1 indicates a line where a drivability error was detected.

PORT 0 BUS TEST (F0 0050) 4-34.

Output drivability test for port 0.

NOTE

The UUT must provide external pull-up resistors on the Port 0 lines for the drivability test to work correctly. Port 0 has open-collector output drivers, and a high output will make them float if they are not pulled up.

PORT 1 BUS TEST (F0 0051) 4-35.

Output drivability test for port 1.

PORT 2 BUS TEST (F0 0052) 4-36.

Output drivability test for port 2.

PORT 3 BUS TEST (F0 0053) 4-37.

Output drivability test for port 3.

Refresh Enable (F0 0060) 4-38.

This special address allows the Pod to provide transparent read operations for the purposes of refreshing dynamic RAM or displays, keyboards, etc. in the UUT. Any non-zero value written to this address will initiate continuous program read operations at address 0000 as well as data read operations at 10000 whenever a read or write operation is done in the Pod.

The default is DISABLED (F0 0060 = 00)--no transparent read operations.

Mode Switch Manipulation (F0 0062 - F0 0064) 4-39.

Two special addresses are provided to allow the operator or a Troubleshooter program to remotely control the DIP Mode Switch located on top of the Pod. Refer to the label on the top of the Pod and Table 2-1, for identification of switch positions.

MODE SWITCH OVERRIDE MASK (F0 0062) 4-40.

This special address contains a mask that specifies which of the switch settings that are defined in Special Address F0 0064 will actually be used to override the Mode Switches. Only those switch positions identified by a 1 in this mask will be used to override the Pod's hardware Mode Switches. Writing new data to this special address is the same as changing the switch settings.

For example, if *WRITE @ F0 0062=02* is used with F0 0064=43 then only one bit (bit 1) of the data in F0 0064 will be used to override the Pod's hardware switches--that specifying port 2 to be I/O.

The default is 00, all switch overrides disabled.

MODE SWITCH OVERRIDE DATA (F0 0064) 4-41.

The Mode Switch Override special address contains a map of switch settings to be used instead of the configuration selected by the Mode Switches on the top of the Pod. Switch positions defined by this special address (and enabled by the mask in special address F0 0062) have precedence over the hardware switches. Data bit positions 0-7 correspond to the switch numbers 1-8 (i.e., bit 4 specifies switch 5). A bit that is written to a 1 specifies a switch that is enabled, and 0 specifies a switch that is disabled (open).

NOTE

Only switches 1 through 5 may be defined. Switch 6 (UUT Connection), switch 7 (Clock Source), and switch 8 (External Clock) are hardware connections and may not be overridden. Bits 5, 6, and 7 are ignored when set.

For example, a *WRITE @ F0 0064=43* configures the switches for Port 0 and 2 to be I/O, port 3 (pins 2,3, and 6) to be $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{WR}}$, and $\overline{\text{RD}}$ respectively. Bit 6 is ignored.

Quick Memory Tests (2X XXXX, 3X XXXX, F0 20XX, F0 30XX) 4-42.

The Quick Memory tests are controlled by software inside the Pod, rather than by software inside the Troubleshooter. They execute much faster than the corresponding tests that are built into the Troubleshooter.

The Quick Memory tests are controlled by writing setup information into special addresses as described below. Appendix A contains a program that may be used to automate these procedures and make the Quick Memory tests appear to the operator to work like the built-in Troubleshooter memory tests.

QUICK RAM TEST ADDRESSES (2X XXXX, F0 20XX) 4-43.

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 is available in two variations: the normal RAM test and a pattern verification test.

- The normal 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 pattern verification test simply verifies that memory contains expected data. It should be used following a normal Quick RAM Test to verify that the memory still contains the correct data after a longer period than is checked by the normal RAM test. It is provided primarily for testing dynamic RAM memory to assure that the memory is retaining information properly. If problems with dynamic RAM are suspected, it is suggested that the pattern verification test be used to follow the normal RAM test.

The starting and ending addresses for the Quick RAM Test are specified in a different manner than for the usual RAM Test. The starting and ending addresses are specified by writing to special addresses. The starting address is defined by a *WRITE @ 2X XXXX=0,* where *X XXXX* is the address to be used to start the Quick RAM test. The ending address, address increment, and test specification are defined by a *WRITE @ 2Y YYYY=ZN,* where *Y YYYY* is the desired ending address, *Z* is the desired increment, and N is the test specification. If *Z* is omitted or specified as 0, the address increment defaults to 1. N may be either 1 (normal Quick RAM test) or 2 (pattern verification test). The ending address must be greater than the starting address.

For example, to specify a normal Quick RAM test over the external RAM addresses 5000 through 5FFF with the default address increment of 1, do the following two operations:

WRITE @ 21 5000=0
WRITE @ 21 5FFF=1

To follow that test with a pattern verification test over the same address space, rewrite the ending address with the new specification:

WRITE @ 21 5FFF=2

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 byte indicating the status of the test or the test results. The status codes and their meanings are shown in Table 4-8.

Table 4-8. Quick RAM Test Addresses and Status Codes

OPERATION	PARAMETERS
WRITE @ 2X XXXX=0 *WRITE @ 2Y YYYY=ZN*	X XXXX = Start Address Y YYYY = End Address Z = Increment N = Test Specification: 1 = Perform RAM Test 2 = Pattern Check
READ @ ENTER	Returns Status Code: 00 = No test requested A0 = Aborted, new command entered A1 = Aborted, illegal data in command A2 = Aborted, illegal address in command B0 = Busy, read/write check B1 = Busy, address decoding check C0 = Complete, no errors F0 = Failed, read/write error F1 = Failed, address decoding error

READ-ONLY ADDRESS	FUNCTION
F0 2000	Start address, LSB
F0 2001	Start address, 2nd byte
F0 2002	Start address, MSB
F0 2004	End address, LSB
F0 2005	End address, 2nd byte
F0 2006	End address, MSB
F0 2008	Error address, LSB
F0 2009	Error address, 2nd byte
F0 200A	Error address, MSB
F0 200C	Expected data
F0 200E	Actual data
F0 2012	Hex mask - bad data bits, or address decode failed bit number
F0 20F0	Most recent code

Read-only special addresses in the F0 20XX range contain addition information about the Quick RAM Test, including records of addresses used and errors. The special addresses for the Quick RAM test are described in Table 4-8.

The Quick RAM Test is only valid in the RAM Address spaces 01 0000 through 02 00FF.

For more information about the test results, the operator may specify read operations at the other special addresses listed in Table 4-8. Make sure that you first specify the *READ @ ENTER* 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.

For example, if any error is reported by the status byte, you can find the Least-Significant Byte of the address where the error occurred by *READ @ F0 2008*. You can get a hex mask of any bad data bits by *READ @ F0 2012*.

QUICK ROM TEST ADDRESSES (3X XXXX, F0 30XX) 4-44.

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 starting and ending addresses are specified by writing to special addresses. The starting address is defined by a *WRITE @ 3X XXXX=0*, where *X XXXX* is the address to be used to start the Quick ROM test. The ending address and address increment are defined by a *WRITE @ 3Y YYYY=Z1*, where *Y YYYY* is the desired ending address and *Z* is the optional increment. If *Z* is not specified or is specified as 0, the increment will default to 1. The ending address must be greater than the starting address.

For example, to specify a Quick ROM test over the internal program ROM addresses 0000 through 0FFF, do the following two operations:

WRITE @ 36 0000=0
WRITE @ 36 0FFF=1

The 3X XXXX portion of the address denotes a Quick ROM test. The X6 XXXX portion indicates operations on the internal ROM address space.

The Quick ROM 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.

The Quick ROM Test is only valid in address spaces 0000 through 300FF and 06 0000 through 6 FFFF.

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 byte indicating the status of the test or the test results. The status codes and their meanings are shown in Table 4-9.

Read-only special addresses in the F0 30XX range contain addition information about the Quick ROM Test, including records of addresses used and errors. The special addresses for the Quick ROM test are described in Table 4-9.

For more information about the test results, the operator may specify read operations at the special addresses listed in Table 4-9. You should first *READ @ ENTER* 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.

Table 4-9. Quick ROM Test Addresses and Status Codes

OPERATION	PARAMETERS
WRITE @ 3X XXXX=0 *WRITE @ 3Y YYYY=ZI*	X XXXX = Start Address Y YYYY = End Address Z = Increment
READ @ ENTER	Returns Status Code: 00 = No test requested A0 = Aborted, new command entered A1 = Aborted, illegal data in command A2 = Aborted, illegal address in command B0 = Busy, read/write check C0 = Complete, no errors C1 = Complete, inactive bits detected

READ-ONLY ADDRESS	FUNCTION
F0 3000	Start address, LSB
F0 3001	Start address, 2nd byte
F0 3002	Start address, MSB
F0 3004	End address, LSB
F0 3005	End address, 2nd byte
F0 3006	End address, MSB
F0 300C	Checksum LSB
F0 300D	Checksum MSB
F0 300E	Hex mask - inactive bits
F0 30F0	Most recent code

Block Memory Tests (4X XXXX, F0 40XX) 4-45.

The Block Memory Tests allow the user to fill blocks of memory with data and then verify accuracy of the contents. The Block Memory Tests are controlled by writing setup information into special addresses as described below.

The Block Memory Tests are much faster than the Troubleshooter's normal memory tests. In addition, they allow the user to customize special memory tests, such as might be desirable when testing a memory-mapped video display.

Appendix A contains a program that may be used to automate these procedures and make the Block Memory Tests appear to the operator to work like the built-in Troubleshooter memory tests.

Three variations of the Block Memory Test are available. The variations are specified when writing the ending address (see below).

- The Fill Test (specified by 1) will write the data that is contained in the starting address to all of the addresses in the block.
- The Verify Test (specified by 2) will read data from all of the addresses in the block an compare each one to the data contained in the starting address. Errors will be reported via the special addresses described below.
- The Fill and Verify Test (specified by 3) combines the Fill Test and the Verify Test into one step.

The Block Memory Test is specified in a manner similar to the Quick RAM Test. The data to be written to all addresses in the block is first written to the starting address, then the starting and ending addresses are specified by writing to special addresses. The starting address is defined by a *WRITE @ 4X XXXX=0,* where *X XXXX* is the address to be used to start the test. The ending address, address increment, and test specification are defined by a *WRITE @ 4Y YYYY=ZN,* where *Y YYYY* is the desired ending address, *Z* is the optional increment, and *N* is the test specification. If *Z* is not specified or is specified as 0, the increment will default to 1. N may be either 1 (fill block), 2 (verify block), or 3 (fill and verify block). The ending address must be greater than the starting address.

For example, to specify a Memory Fill for the external data memory (16-bit address) 0000 through 0FFF with an increment of 2 and data AA, do the following two operations:

WRITE @ 01 0000=AA (Write data AA to starting address)

WRITE @ 41 0000=0 (Specify starting address)

WRITE @ 41 0FFF=21 (Specify ending address, increment, and test)

The Block Memory 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.

Block Memory Tests are valid only in the Data Memory address space 01 0000 through 03 00FF.

Special addresses in the range of F0 40XX contain additional information about the Block Memory Test. These read-only locations contain records of the addresses used, errors, and other information. The special addresses for the Block Memory Test are described in Table 4-10.

Table 4-10. Block Memory Test Addresses and Status Codes

OPERATION	PARAMETERS
WRITE @ X XXXX=DD	X XXXX= Start Address DD = Fill Data
WRITE @ 4X XXXX=0 *WRITE @ 4Y YYYY=ZN*	X XXXX = Start Address Y YYYY = End Address Z = Increment N = Test Specification: 1 = Fill Memory 2 = Verify 3 = Fill and Verify
READ @ ENTER	Returns Status Code: 00 = No test requested A0 = Aborted, new command entered A1 = Aborted, illegal data in command A2 = Aborted, illegal address in command B0 = Busy, filling B1 = Busy, verifying C0 = Complete, no errors F0 = Failed, verify

READ-ONLY ADDRESS	FUNCTION
F0 4000	Start address, LSB
F0 4001	Start address, 2nd byte
F0 4002	Start address, MSB
F0 4004	End address, LSB
F0 4005	End address, 2nd byte
F0 4006	End address, MSB
F0 4008	Error address, LSB
F0 4009	Error address, 2nd byte
F0 400A	Error address, MSB
F0 400C	Expected data
F0 400E	Actual data
F0 40F0	Most recent code

To determine if the Quick Block 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 byte indicating the status of the test or the test results. The status codes and their meanings are shown in Table 4-10.

For more information about the test results, the operator may specify read operations at the special addresses listed in Table 4-10. You should first *READ @ ENTER* 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.

CHARACTERISTICS OF THE LEARN OPERATION 4-46.

The Learn operation of the Troubleshooter is designed to operate on bus-oriented systems employing RAM, ROM, and I/O that is external to the microprocessor. While the Learn operation of the Troubleshooter may be used with the 8044 and 8051 families of microprocessors, the results are not the same as those achieved with other types of microprocessors.

By default, the Learn function will evaluate only external address spaces as defined by the Mode Switch configuration. The Learn function may also be forced to evaluate internal address spaces. Such an operation might be necessary, for example, to get a checksum from internal ROM. Mode switch settings and the resultant learnable address blocks are described in Table 2-2, Learnable External Address Blocks.

I/O addresses reported by the Learn operation are those addresses which contain read/writable bits, but do not otherwise qualify as RAM.

I/O is reported for the special function registers (4 0080 - 4 00FF) and the bit address registers (5 0000 - 5 00FF) if those address ranges are specified.

NOTE

If I/O ports cause the Learn operation to stop because of drivability errors on input lines, the reporting of drivability errors may be disabled in the troubleshooter Setup operation by setting the SET-TRAP ADDR ERR and/or the SET-TRAP DATA ERR messages to NO. However, if driveability error reporting is disabled to allow the Learn operation to be performed, be sure to enable the error reporting when performing operations other than Learn.

CHARACTERISTICS OF BUS TEST 4-47.

Bus testing is dependent on the configuration switch settings for the lines being tested. If Port 0 and Port 2 are configured to be address/data lines, they are tested for drivability. If P3.6 and P3.7 are used as $\overline{RD}$ and $\overline{WR}$, they are also tested for drivability. $\overline{PSEN}$ and ALE are always tested for drivability.

The default address for the Bus Test is also determined by the configuration switch settings. If Port 0 is set to Address/Data and the $\overline{RD}$ and $\overline{WR}$ lines are enabled, then the address is 2 0000. If Port 2 is set to Addresses then the Bus Test address is 1 0000. If the switches are all set for I/O, or if switch 5 is set for I/O (and therefore, there is no $\overline{RD}$ and $\overline{WR}$ lines), then the bus test is performed at 3 0000, the internal RAM address space.

RUN UUT Limitations 4-48.

CONFIGURATION 4-49.

The 8044's and 8051's are versatile microprocessors with many modes of operation. Because of this versatility, and because there is a serious lack of external clues as to what the microprocessor is doing or what it will do next, it is necessary to have the user declare what his UUT configuration is. The UUT/Pod configuration is accomplished through two means: the DIP switches and through a special address meant for overriding the DIP switch settings. This allows the user to test any UUT, regardless of configuration, since it is within his control to change the Pod's idea of the configuration at any time during testing.

Because of the use of DIP switches and/or special addresses to configure the Pod there is an important limitation on RUN UUT. That is, RUN UUT will only work if the UUT does not require that the Pod change the configuration of Port 0 or Port 2 during the course of the RUN UUT operation.

SYNCHRONOUS RESETS 4-50.

One further limitation on RUN UUT is that it will not work if the UUT depends on a synchronous Reset. This is because the Pod must drop out of Run UUT any time it is Reset in order to resynchronize its state counter. The state counter is essential for buffer operation in RUN UUT. The result is that when the Pod experiences a UUT Reset while performing RUN UUT, the Pod drops out of RUN UUT, takes an extra 100 μsec or so to resynchronize the state counter, and then jumps back into RUN UUT at the Reset address.

PROGRAMMING MODES 4-51.

The EPROM programming modes are not supported for RUN UUT.

ACTIVE ALE SIGNAL 4-52.

During all UUT testing, and as long as the UUT power supply is operable, the ALE signal is continuously present. This signal is provided to accommodate those UUTs which depend on ALE in order to remain active.

NOTE

The generation of the ALE output signal is suspended whenever the UUT power level falls to an unsafe level. This feature protects sensitive devices within the UUT which may otherwise be damaged by this signal.

PROBE SYNCHRONIZATION MODES 4-53.

Two synchronization modes are provided by the Pod, address sync and data sync. The modes synchronize the Troubleshooter probe operation to internal microprocessor events. When enabled, the sync output is active low once during each UUT access. The leading edge of the sync pulse begins at a time slightly prior to the event of interest so that the probe may be activated during the period of interest. The trailing edge of the sync pulse occurs at the moment when address or data is valid. If a probe is being used to capture data, then data is sampled on the trailing (rising) edge of the sync pulse.

Address sync is active from the falling edge of ALE during the cycle prior until the falling edge of ALE during UUTON and is gated to the sync output. Data sync may originate from several signal sources: $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, or UUTON. The source of the data sync signal depends on the type of access in progress. If any other sync mode other than address or data is specified, the sync output of the Pod is disabled.

MARGINAL UUT PROBLEMS 4-54.

Introduction 4-55.

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 with the actual 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.

UUT Operating Speed and Memory Access 4-56.

Some UUTs are designed to 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.

UUT Noise Levels 4-57.

As long as 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 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.

Bus Loading 4-58.

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.

Clock Loading 4-59.

The Pod has a clock oscillator which replaces the UUT microprocessor oscillator. The Pod clock oscillator is designed to be less sensitive to capacitive loading than the UUT microprocessor oscillator. However, if the UUT has a clock source other than a crystal, it will experience greater than normal loading due to the capacitance of the Pod and its cable. While this loading will rarely have any effect on clock operation, it may make marginal clock sources easier to detect.

POD DRIVE CAPABILITY 4-60.

As a driving source on the UUT bus, the Pod generally provides equal to or better than normal 8044- and 8051-family current drive capability. However, lines P1.0 - P1.7 and P3.0 - P3.5 are not buffered and have approximately 100-ohm of series resistance (due to the protection network) between the microprocessor and the UUT. All Pod inputs and outputs (except the clock and RST) are TTL-compatible.

DRIVABILITY TESTING 4-61.

I/O Ports 4-62.

Data drivability is tested for all WRITE operations to the ports Port 0, Port 1, Port 2 and Port 3. This applies to all the byte and bit addresses listed in table 4-5. In each case, the least-significant bit of the port corresponds to data bit 0 and the most-significant bit to data bit 7. For example if a *WRITE @ 4 0090 = 00* resulted in data error 40, then it was P1.6 which had the problem.

Writes to the bit addresses will report data drivability errors as though a byte address had been used. For example, if the same error above were detected by a bit address, it would have happened as *WRITE @ 5 0096 = 00*, resulting in the same data error, 40. The data error refers to the port as a 8-bit object even though a 1-bit access was performed, and only one bit was tested.

Special Function Registers 4-63.

Only WRITE operations to registers which correspond to the I/O ports result in drivability testing. Other special function registers which may result in active outputs (such as SBUF), do not result in drivability tests being performed on those outputs. It is suggested that all port lines be tested for drivability by accessing the ports directly prior to performing functional tests using the other special function registers.

UUT POWER DETECTION 4-64.

A power sensing circuit within the Pod sends a power fail message to the Troubleshooter whenever the +5V power supply in the UUT drops below 4.5V or increases above 5.5V. This circuit causes the Troubleshooter to display a *BAD POWER SUPPLY* error message.

The *BAD POWER SUPPLY* message may be suppressed by changing the Setup command *SET - TRAP BAD POWER SUPPLY? YES* to *NO.*

Also, any time that the UUT power supply drops below about 3.5V, all active Pod outputs are disabled or driven to their high logic level. This feature has been incorporated to protect UUT circuits from possibly being damaged by Pod outputs when the power supply drops below safe operating limits. When proper power levels are restored to the UUT, the Pod outputs will return to normal, and the Troubleshooter will be ready for additional testing.

CLOCK SOURCES, SETUPS, AND PROBLEMS 4-65.

The 8051 oscillator has several different modes of operation, and for correct operation of the Pod the user must configure the Pod for his particular UUT using the mode switches. The position of switch 7, Clock Source, is determined by whether the UUT has a separate clock generator driving the processor, or a crystal across pins 18 and 19. Switch 8, External Clock, is set depending on whether pin 18 (XTAL2) is driven from an external source or pin 19 (XTAL1).

NOTE

If the UUT has a gate that is driven from pin 18 of the microprocessor and a crystal across pins 18 and 19, the switches should be configured with switch 7 set to XTAL and switch 8 set to XTAL1. This will make the Pod capable of driving a CMOS gate from pin 18. A TTL gate will load the oscillator circuit and cause unpredictable behavior.

EFFECTS OF THE CONFIGURATION SWITCHES 4-66.

Introduction 4-67.

The procedure for setting the configuration switches is explained in Section 2. Each of the switch settings has widespread effects on learnable addresses, port characteristics, and other facets of Pod operation. The effects of each of the configuration switches is described here in detail and summarized in Table 4-11. Table 2-2 shows the settings of the configuration switches.

Table 4-11. Effects of Configuration Switches

SWITCH	AFFECTED POD CHARACTERISTIC
Port 0	Default LEARN addresses Default BUS TEST address Address lines tested by BUS TEST Port 0 configuration during READ and WRITE Port 0 configuration during RUN UUT
Port 2	Default LEARN addresses Default BUS TEST address Address lines tested by BUS TEST Port 2 configuration during READ and WRITE Port 2 configuration during RUN UUT
P3.2	Generation of active interrupt report
P3.3	Generation of active interrupt report
P3.6 and P3.7	Default LEARN addresses Default BUS TEST address Control lines tested by BUS TEST
UUT Connection	Reset line behavior Forcing line report
CLK Source	Clock/oscillator circuitry
External CLK	XTAL1 and XTAL2 connections to clock/ oscillator circuitry

Switch 1 (Port 0) 4-68.

DEFAULT LEARN ADDRESSES 4-69.

If Port 0 is declared to be I/O, no external memory devices are possible. In this case there are no default LEARN addresses supplied. If Port 0 is declared to be ADDR/DATA then the default LEARN addresses depend on the setting of the Port 2 and the P3.6/P3.7 switches.

DEFAULT BUS TEST ADDRESS 4-70.

If Port 0 is declared to be I/O, no external bus is considered to exist. In this case the default BUS TEST address is 3 0000--the internal RAM space. If Port 0 is declared to be ADDR/DATA the default BUS TEST address depends on the setting of the Port 2 and P3.6/P3.7 switches. Depending on those settings, the default BUS TEST address may be 1 0000, 2 0000, or 3 0000.

ADDRESS LINES TESTED BY BUS TEST 4-71.

The low-order address bus is only checked for drivability if Port 0 is declared as ADDR/DATA.

PORT 0 CONFIGURATION DURING READ AND WRITE 4-72.

Port 0 will latch an I/O output on its lines and will keep that data there throughout testing of addresses which do not affect Port 0 only if Port 0 is declared as I/O. If it is declared ADDR/DATA, I/O outputs will not remain latched on Port 0. The port is instead tri-stated when not in use as address or data.

PORT 0 CONFIGURATION DURING RUN UUT 4-73.

Port 0 will behave as only an open-drain I/O port or as an address/data bus, depending on the switch setting, during RUN UUT. A UUT which attempts to utilize both modes in its program, may not be able to perform RUN UUT correctly.

Switch 2 (Port 2) 4-74.

DEFAULT LEARN ADDRESSES 4-75.

The default LEARN addresses depend on the settings of the Port 0, Port 2, and P3.6/P3.7 switches. Port 2 must be declared as ADDR in order for the external program ROM and 16-bit address RAM to be included as default LEARN addresses.

DEFAULT BUS TEST ADDRESS 4-76.

With Port 0 declared as ADDR/DATA and P3.6/P3.7 declared as $\overline{RD}/\overline{WR}$, if Port 2 is declared as ADDR, then the default BUS TEST address is 1 0000. If Port 2 is declared as I/O then the BUS TEST address is 2 0000.

ADDRESS LINES TESTED BY BUS TEST 4-77.

The high-order address bus is only checked for drivability if Port 2 is declared as ADDR.

PORT 2 CONFIGURATION DURING READ AND WRITE 4-78.

Port 2 will latch high-order addresses or I/O outputs on its lines throughout all UUT testing. However, high data on these lines is buffered differently depending upon the setting of the Port 2 switch. If Port 2 is declared as I/O then high outputs only source current on high-going transitions and only for two clock periods. The high output is then left to be driven through a 22K ohm pull-up resistor. If Port 2 is declared as ADDR, then high outputs source current continuously.

PORT 2 CONFIGURATION DURING RUN UUT 4-79.

Port 2 will behave as only a quasi-bidirectional I/O port or as an address bus, depending on the switch setting, during RUN UUT. A UUT which attempts to utilize both modes in its program, may not be able to perform RUN UUT correctly.

Switch 3 (P3.2) 4-80.

GENERATION OF ACTIVE INTERRUPT REPORT 4-81.

When P3.2 is declared as $\overline{INT0}$, a low value sensed on the line will cause an active interrupt report. When declared as I/O, values on this line do not affect the active interrupt report.

Switch 4 (P3.3) 4-82.

GENERATION OF ACTIVE INTERRUPT REPORT 4-83.

When P3.3 is declared as $\overline{INT1}$, a low value sensed on the line will cause an active interrupt report. When declared as I/O, values on this line do not affect the active interrupt report.

Switch 5 (P3.6/P3.7) 4-84.

DEFAULT LEARN ADDRESSES 4-85.

When declared as I/O, no external RAM space is included as default LEARN addresses. If this switch declares P3.6/P3.7 to be $\overline{RD}/\overline{WR}$, then one or both of the external RAM address spaces may be included as default LEARN addresses, depending on the setting of the Port 0 and Port 2 switches.

DEFAULT BUS TEST ADDRESS 4-86.

When P3.6/P3.7 are declared as $\overline{RD}/\overline{WR}$, one of the external RAM addresses 1 0000 or 2 0000 may be used as the default BUS TEST address. If this switch declares the lines to be I/O, then the internal RAM space, 3 0000, is used as the default BUS TEST address.

CONTROL LINES TESTED BY BUS TEST 4-87.

The $\overline{RD}$ and $\overline{WR}$ control lines are only tested for drivability if this switch declares P3.6/P3.7 to be $\overline{RD}/\overline{WR}$. If they are declared to be I/O then they are not tested for drivability.

Switch 6 (UUT Connection) 4-88.

RESET LINE BEHAVIOR 4-89.

When set to the "Clip-on" position, this switch causes the RST line to be permanently pulled high.

FORCING LINE REPORT 4-90.

With the switch set to the "Clip-on" position RST is held high. This is the active state for this line which is a forcing line. In order to avoid reporting an active forcing line on every UUT operation, the RST line is omitted from the forcing line report.

Switch 7 (CLK Source) 4-91.

CLOCK/OSCILLATOR CIRCUITRY 4-92.

When set to the XTAL position, the clock circuitry behaves as an oscillator. When set to the TTL position, the clock circuitry behaves as a buffer.

Switch 8 (External CLK) 4-93.

XTAL1 AND XTAL2 CONNECTIONS TO CLOCK/OSCILLATOR CIRCUITRY 4-94.

When the CLK source switch is set to the TTL position, one of the XTAL lines is used as an input and the other is not used. The External CLK switch allow the XTAL lines to be swapped so that the proper one gets the TTL input.

Section 5
Theory of Operation

INTRODUCTION 5-1.

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.

GENERAL POD OPERATION 5-2.

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.

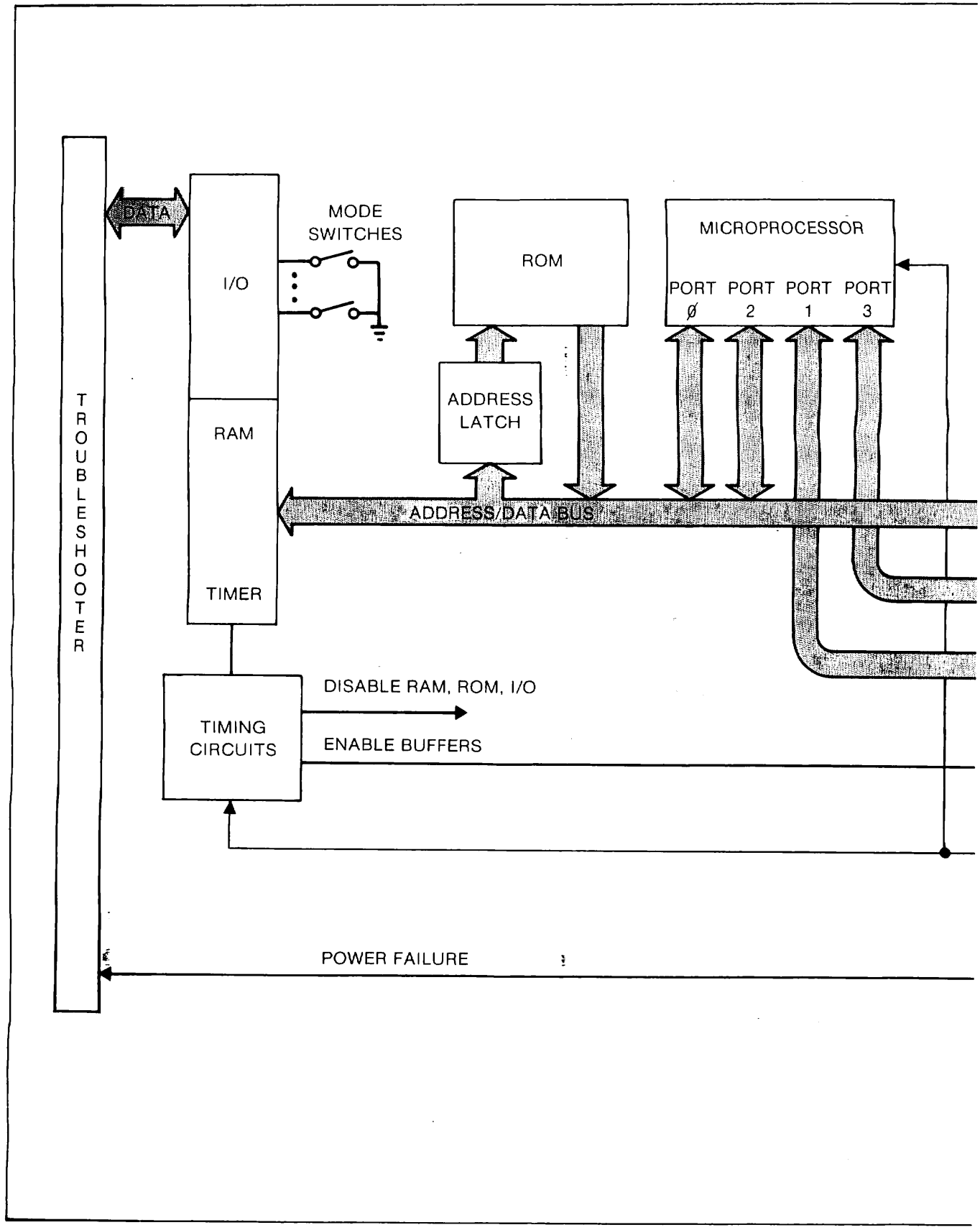
The Pod may be divided into four major areas:

- Processor Section
- UUT Interface Section
- Timing and Control Section
- UUT Power Sensing

Each section is described in the following paragraphs.

Processor Section 5-3.

The Processor Section, shown in Figure 5-1, is made up of a microprocessor, RAM, ROM, an I/O interface to the Troubleshooter, and various latches and buffers. These elements, along with timing components, constitute a small microsystem which receives Troubleshooter commands and directs all Pod operations during execution.


TROUBLESHOOTER
DATA
I/O
RAM
TIMER
MODE
SWITCHES
ROM
MICROPROCESSOR
PORT Ø
PORT 2
PORT 1
PORT 3
ADDRESS
LATCH
ADDRESS/DATA BUS
TIMING
CIRCUITS
DISABLE RAM, ROM, I/O
ENABLE BUFFERS
POWER FAILURE

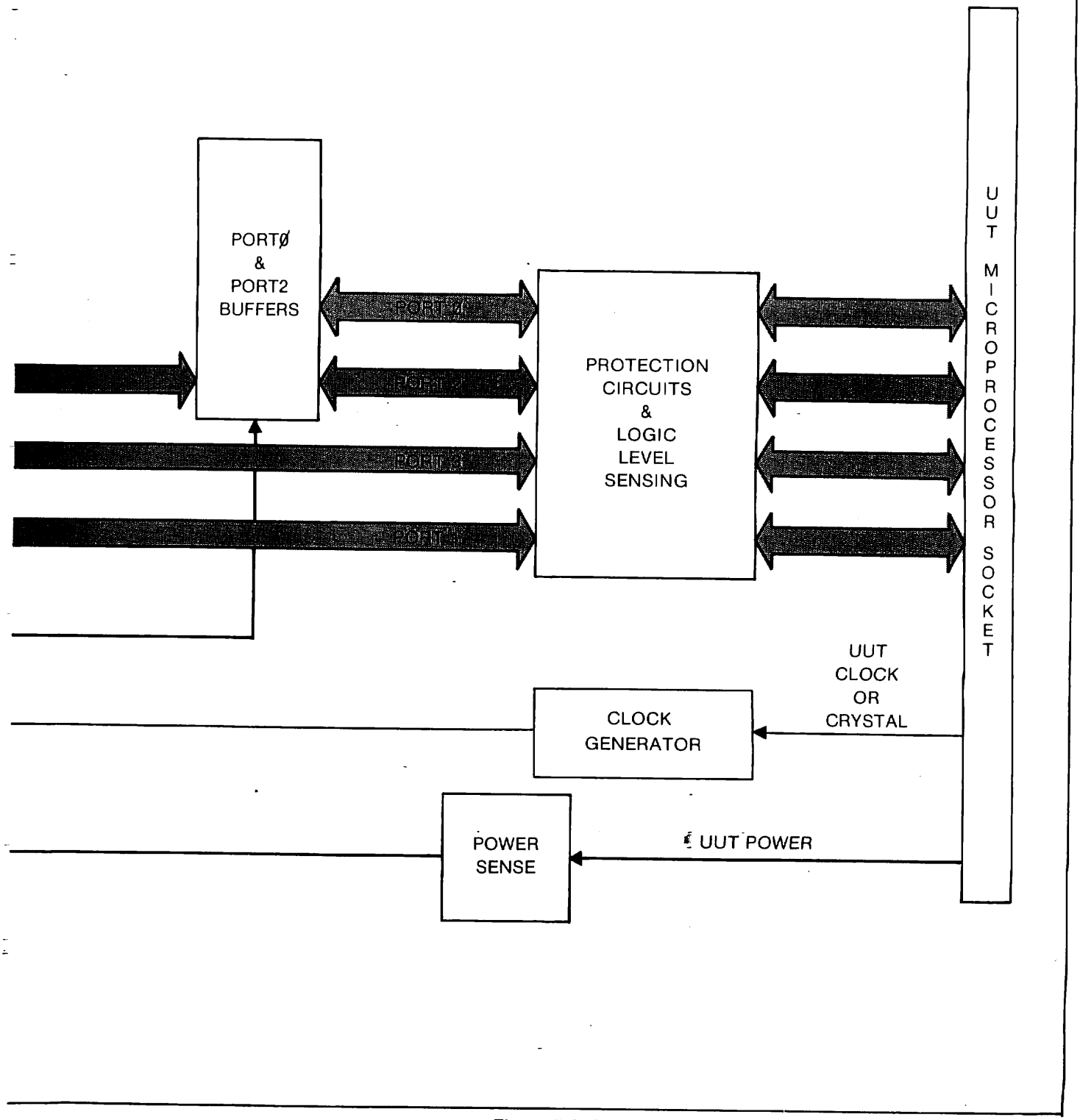


Figure 5-1. General Block Diagram

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 RST, 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.

UUT Interface Section 5-4.

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.

Timing and Control Section 5-5.

The Timing and Control Section, shown in Figure 5-1, controls the buffers to the Pod's internal RAM, ROM, and I/O, and to the UUT. When a UUT access occurs, the internal data bus is disabled and the buffers to the UUT are enabled. During alternate times, the UUT buffers are disabled and the internal data bus is enabled.

The Timing and Control section consists of an interval counter which, when preset by the microprocessor, determines the time when the microprocessor switches between addressing the Pod's internal devices and the UUT's devices. The microprocessor sends information to the internal ports (Ports B and D) which determines whether an operation is a port access, program read, or a data read or write.

If the microprocessor has sent the Run UUT command through Port D, then the buffers are switched from internal context to UUT context 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.

UUT Power Sensing 5-6.

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 3.5V, all active Pod outputs are disabled or written to their high 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.

DETAILED BLOCK DIAGRAM DESCRIPTION 5-7.

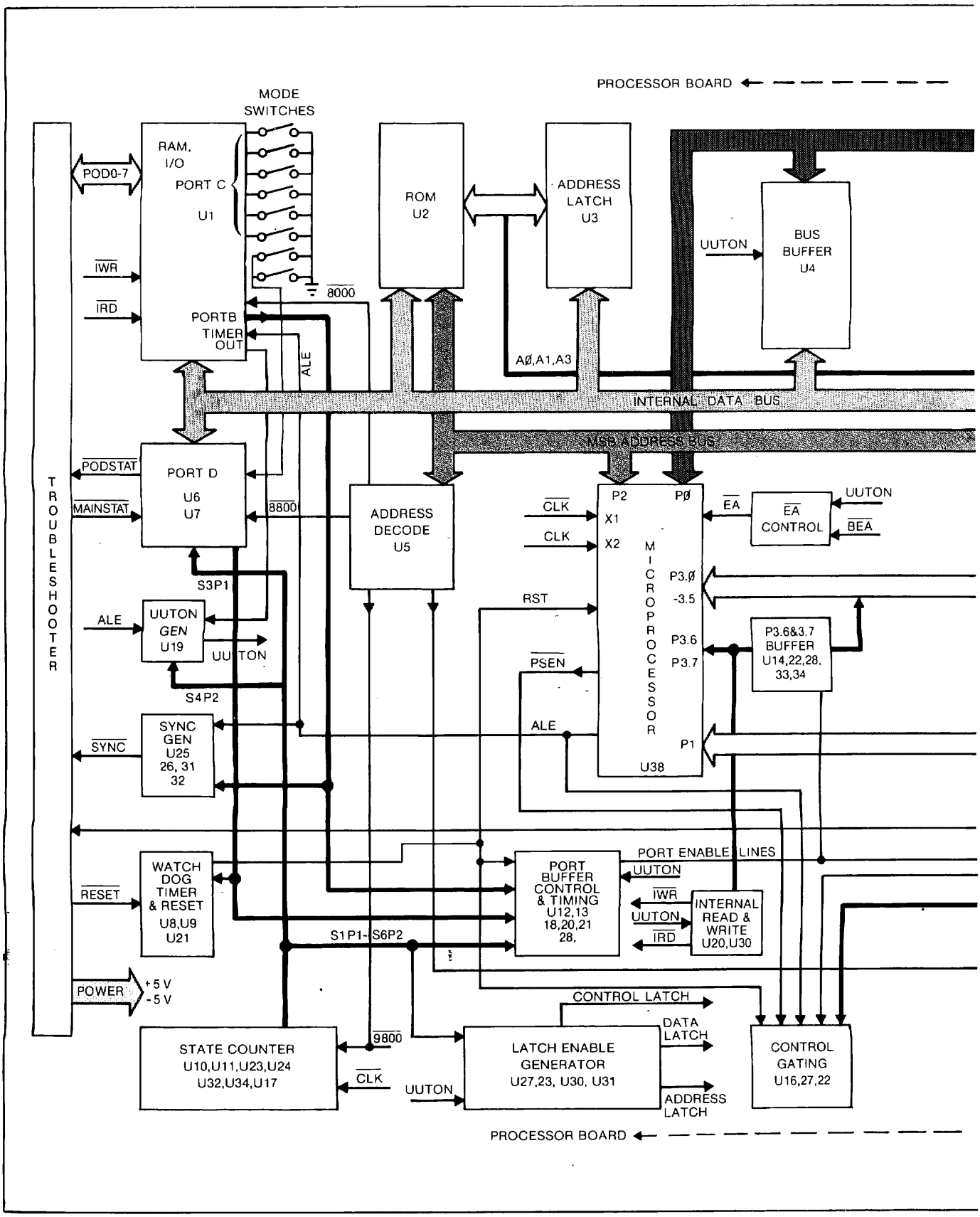
Each major section is described in the following paragraphs along with a separate description of the self test circuit. Figure 5-2 is a detailed illustration of the Pod operation.

Detailed Description of the Processor Section 5-8.

The microprocessor (U38, in ZIF socket J3) along with ROM (U2), low order address latch (U3), and address decoder (U5) form a small microprocessor system which is the kernal of the Pod. The RAM, I/O, and Timer (U1) handle the eight I/O lines to the Troubleshooter on Port A, along with the first six mode switch inputs on port C. The interval timer for switching the buffers from internal context to UUT context is also on this device. Port B puts out control signals for the type of access to take place: the Port mode, program reads, and other functions. Port D consists of an input buffer (U6) and an output latch (U7). The handshake lines $\overline{\text{MAINSTAT}}$ and $\overline{\text{PODSTAT}}$ are bit 0 of this port.

All communication between the Pod and the Troubleshooter uses the handshake protocol shown in Figure 5-3. $\overline{\text{MAINSTAT}}$ 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.

The watchdog timer and reset circuit consists of a 18-bit counter which, if it reaches full count, allows the $\overline{\text{PODRESET}}$ signal from the Troubleshooter to reach the microprocessor. While the Pod is operating correctly, the Watchdog timer is reset by WDRST from port D periodically. A flip-flop (U21) latches a PODRESET so that the Pod can respond correctly to the Troubleshooter when it sends a RESET signal. If the Pod's software gets into an infinite loop or otherwise operates improperly, then the counter allows the Troubleshooter to reset the microprocessor. In normal Run UUT operation, the UUT Reset line is gated to the microprocessor. If the clip-on option is enabled, the UUT processor is held Reset and a Reset from the UUT will have no effect.


PROCESSOR BOARD
MODE SWITCHES
RAM, I/O
PORT C
U1
POD0-7
IWR
IRD
PORTB
TIMER OUT
ALE
8000
ROM U2
ADDRESS LATCH U3
UUTON
BUS BUFFER U4
A0,A1,A3
INTERNAL DATA BUS
MSB ADDRESS BUS
TROUBLESHOOTER
PODSTAT
MAINSTAT
PORT D U6 U7
8800
ADDRESS DECODE U5
CLK
CLK
X1
X2
P2
P0
MICROPROCESSOR
U38
EA
EA CONTROL
UUTON
BEA
P3.0 -3.5
P3.6
P3.7
P3.6&3.7 BUFFER U14,22,28, 33,34
RST
PSEN
ALE
P1
S3P1
ALE
UUTON GEN U19
UUTON
S4P2
SYNC
SYNC GEN U25 26, 31 32
RESET
WATCH DOG TIMER & RESET U8,U9 U21
S1P1-S6P2
PORT BUFFER CONTROL & TIMING U12,13 18,20,21 28,
PORT ENABLE LINES
UUTON
IWR
UUTON
IRD
INTERNAL READ & WRITE U20,U30
POWER
+5 V
-5 V
CONTROL LATCH
DATA LATCH
STATE COUNTER U10,U11,U23,U24 U32,U34,U17
9800
CLK
UUTON
LATCH ENABLE GENERATOR U27,23, U30, U31
ADDRESS LATCH
CONTROL GATING U16,27,22
PROCESSOR BOARD

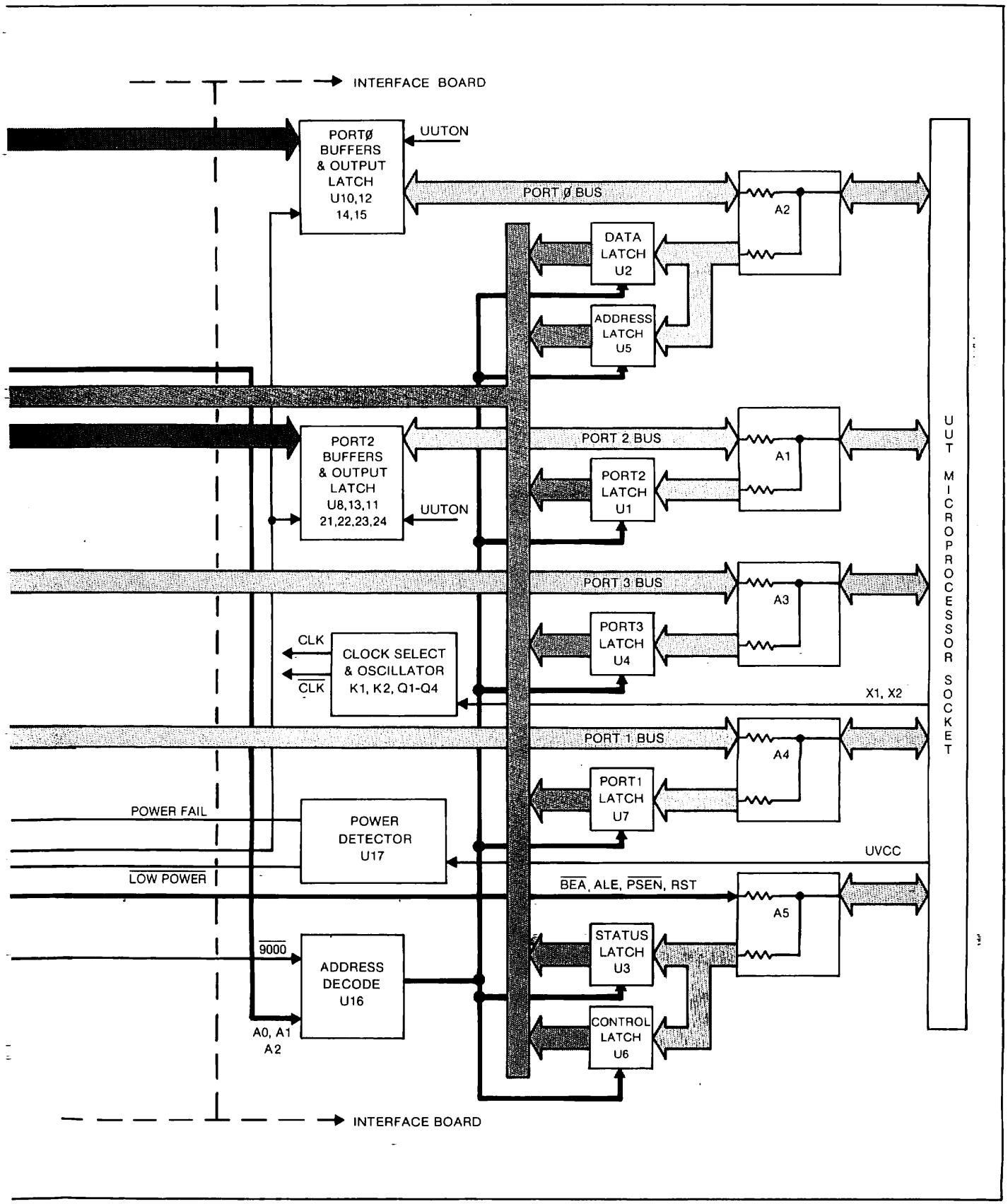


Figure 5-2. Detailed Block Diagram

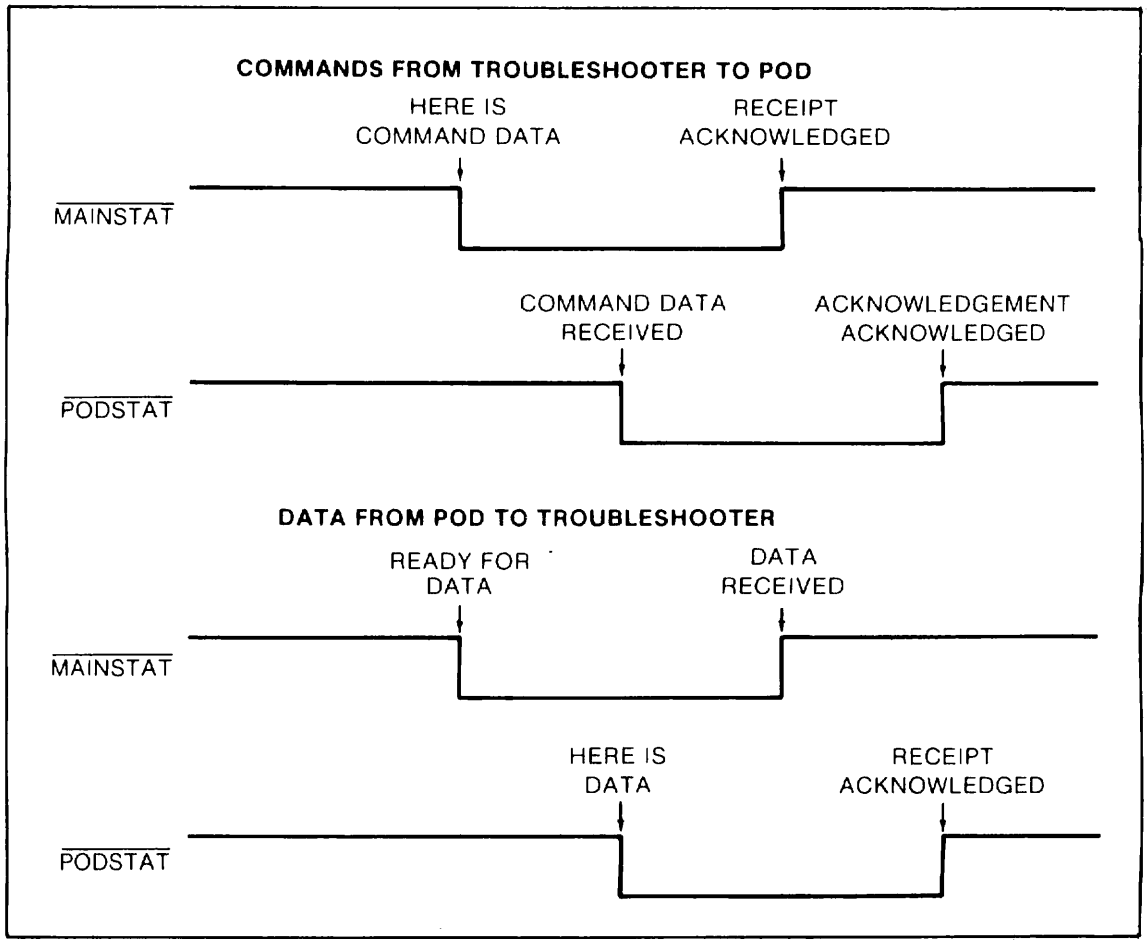


Figure 5-3. Handshake Signals

The state counter (U10 and U11) keeps track of which state the microprocessor is in internally. The result is used to control the timing of bus cycles and buffer direction. The state counter is resynchronized after each UUT access. The port buffer and control timing consists of two programmed-logic arrays (U12 and U13) which convert the timing signals derived from the state counter and the control signals from port B and D, which tell which type of operation is to happen, into the buffer control signals.

P3.6 and P3.7 buffers consist of U14, U28, and U33. These signals, which are used as read and write internally, are latched to the UUT to simulate the actual processor in the port mode.

To simulate the microprocessor's quasi-bidirectional port structure, in which the pull-up current of the outputs is increased during the first two clock cycles before being allowed to stabilize, the buffers in U33 are driven high for two clock cycles, then tristated.

The selected sync mode is enabled by the signals on the lower three bits (I/O Cycle, ADRSYNC, and DSYNC) of port B (U1). Address sync is generated by a flip-flop (U19), clocked by EN(able)UUTON, reset by the delayed falling edge of ALE, gated with UUTON (U27). D(ata)SYNC depends on what type of instruction is being executed (see timing diagrams). The gates U32, U26, and U25 arbitrate the period used for data sync.

Internal read and write strobes ($\overline{IRD}$ and $\overline{IWR}$) are generated by flip-flops U20 and U30. These are disabled during UUTON.

The microprocessor $\overline{EA}$ line is gated by U22, U23, and U32. The Pod controls the state of $\overline{EA}$ during testing and lets the UUT control it during Run UUT.

NOTE

During normal operation, the $\overline{EA}$ line is kept low, providing access to external memory. If an attempt is made to read the microprocessor's internal memory, the $\overline{EA}$ line will be switched high. Even though the manufacturer does not recommend that the state of $\overline{EA}$ be switched during operation this way, it works reliably with current products. If, in the future, you find microprocessors that will not allow you to read internal memory, you will need to verify it separately (using a ROM reader) using the manufacturer's recommended techniques.

Detailed Description of the UUT Interface Section 5-9.

Each UUT interface line is protected by a 3K 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 A5 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 U1-U7 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.

The buffers and output latch for Port 0 consist of a bidirectional bus buffer (U10), which is used for multiplexed address/data type bus accesses, an output buffer (U12) to provide the 00 address during transparent reads, and a latch and open-collector buffer (U14 and U15), which provides the data for a port output operation.

The buffers and output latch for Port 2 consist of several devices. U8 is an input buffer, enabled only during read port operations. U11 is a output buffer for providing the 00 address during a transparent read. U13 and U21 through U24 are an output latch and buffer which provides high addresses or port data.

When a high is written to a Port 2 bit in the I/O mode, the buffer is enabled 2 clock cycles, then the output lines are tri-stated. This procedure simulates the "quasi bi-directional port" characteristics of the microprocessor.

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.

Detailed Description of the Timing and Control Section 5-10.

Timing and control functions govern the internal Pod cycles and the UUT access (UUTON) cycles. When in the Pod context, the microprocessor performs Pod functions; exchanging information with the Troubleshooter and performing tasks dictated by Troubleshooter commands. During UUT context, the microprocessor performs UUT functions as it would if it were plugged into the UUT directly. Signal timing is shown in Figures 5-4 through 5-7.

POD CONTEXT 5-11.

The Pod context is primarily devoted to exchanging information with the Troubleshooter and performing tasks dictated by Troubleshooter commands. In this state, the microprocessor 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.

UUT CONTEXT 5-12.

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

The bus buffer (U4) is disabled during UUT accesses to keep the Pod inactive.

UUT Power Sensing 5-13.

Comparator U17 and its associated circuitry monitor the UUT power supply voltage. U17 sends an error signal (PWRFAIL) to the Pod and the Troubleshooter if the UUT power supply voltage is not between 4.5V (nominal) and 5.5V (nominal). This signal indicates that the UUT power supply is operating improperly. To prevent possible damage to the UUT, U17 also inhibits all Pod outputs from sourcing more than a few microamps if the UUT power supply voltage falls below approximately 3.5V (LOWPWR).

Detailed Description of the Self Test Circuit 5-14.

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.

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.

Oscillator Y1 provides the microprocessor with a 6.5 MHz clock signal during Pod self test.

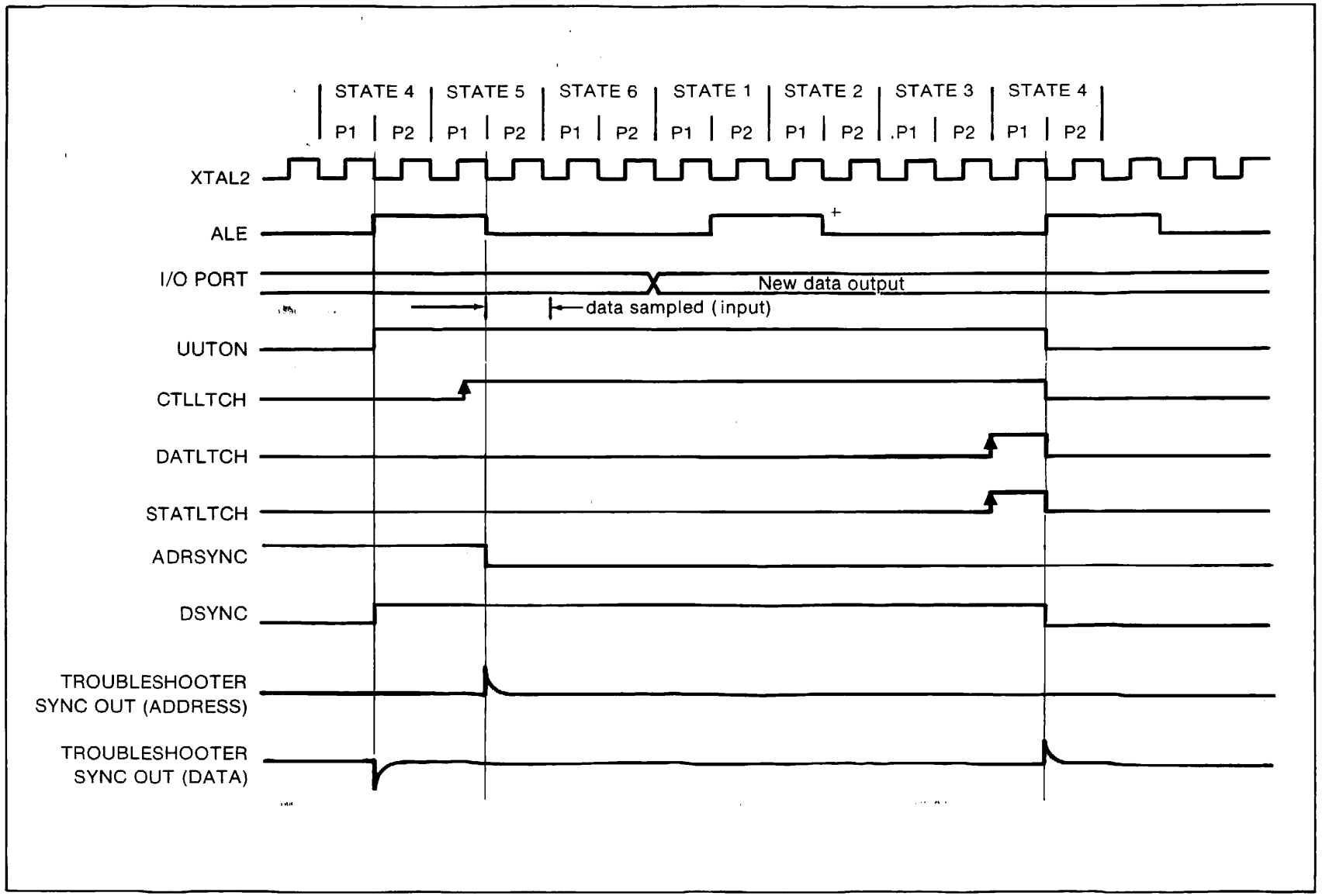


Figure 5-4. Latch Times for Data and Status During Port 1 and Port 3 Accesses

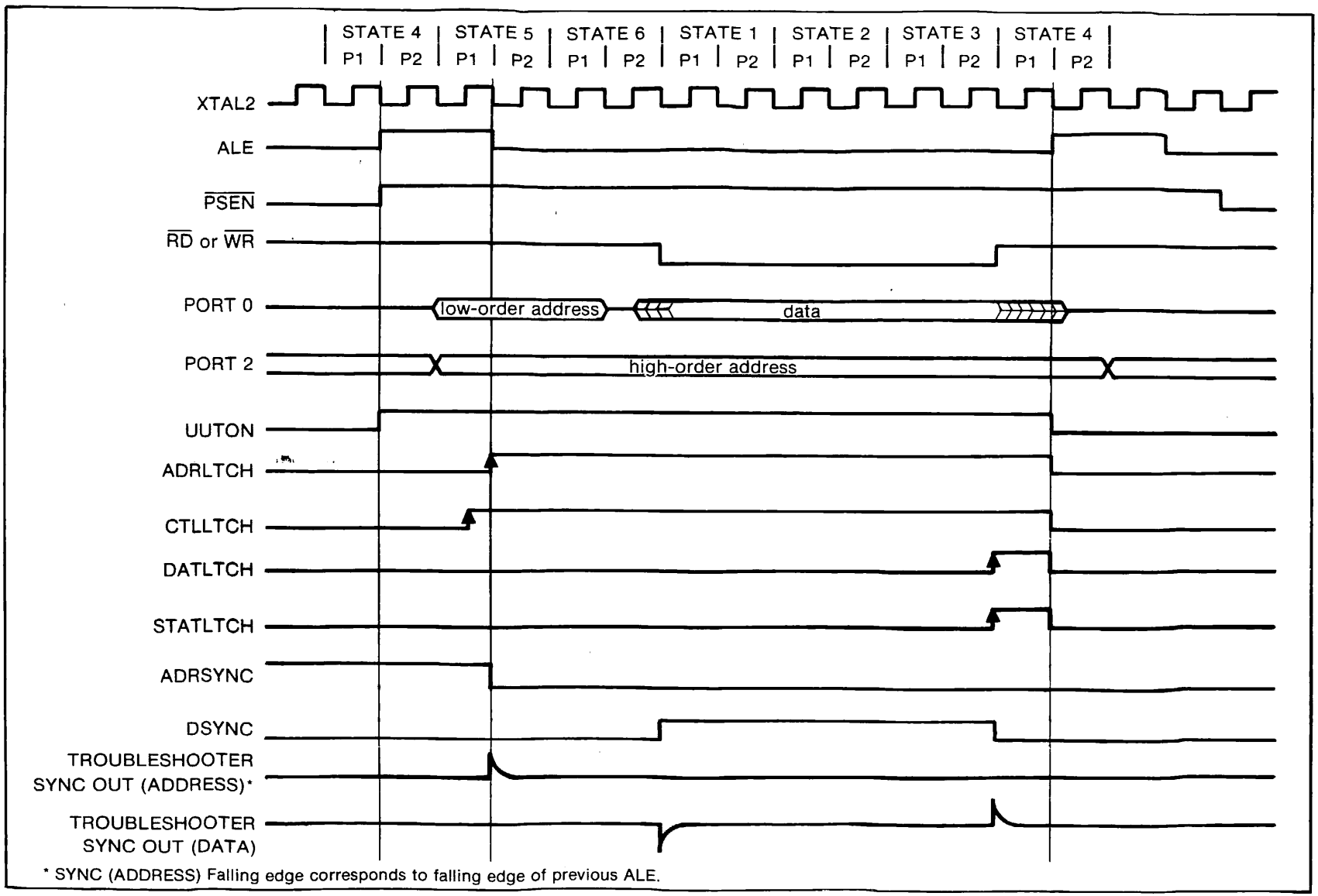


Figure 5-5. Latch Times for Data, Status, and Control During 16-Bit Address External Data Access

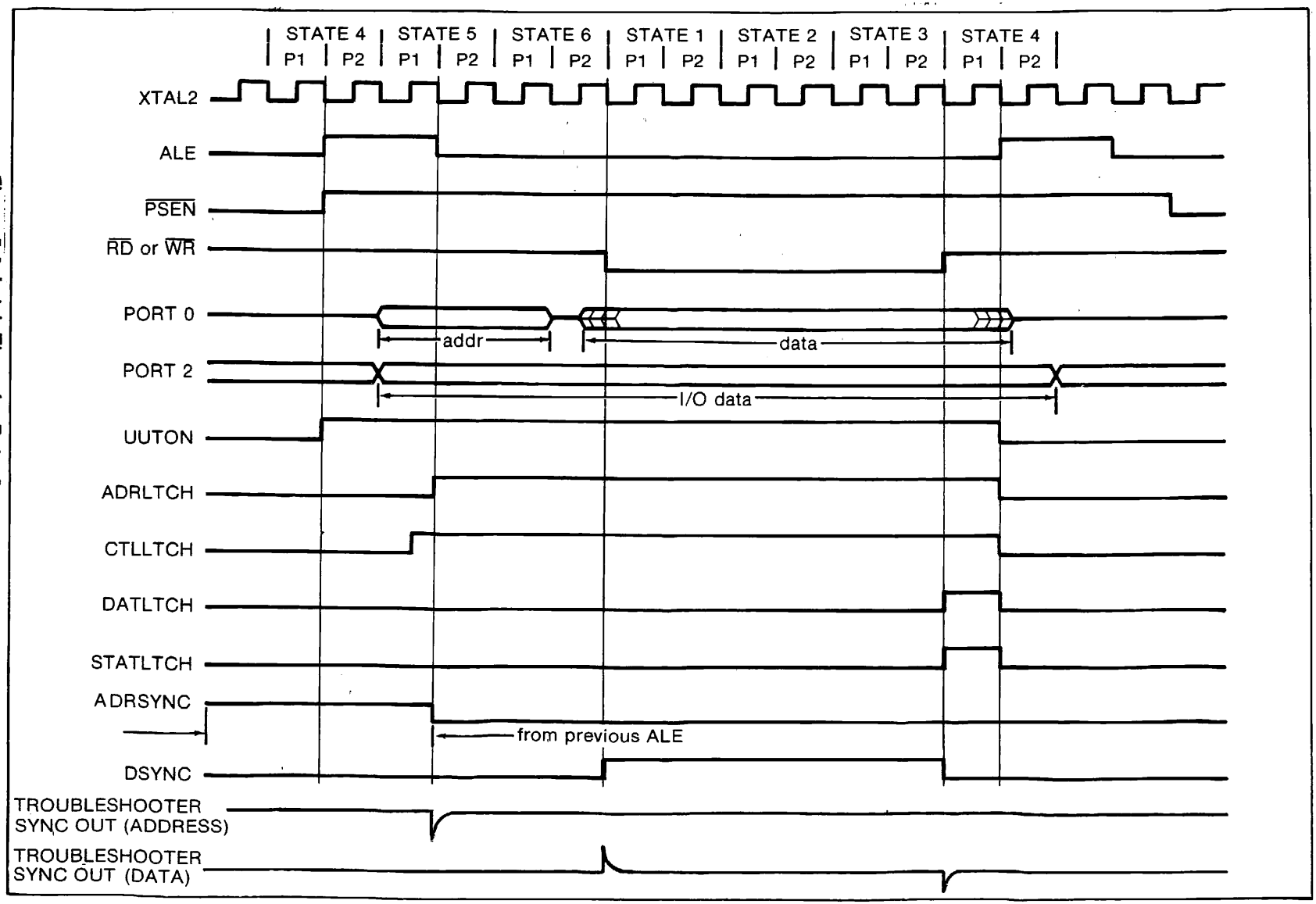


Figure 5-6. Latch Times for Data, Status, and Control During 8-Bit Address External Data Access

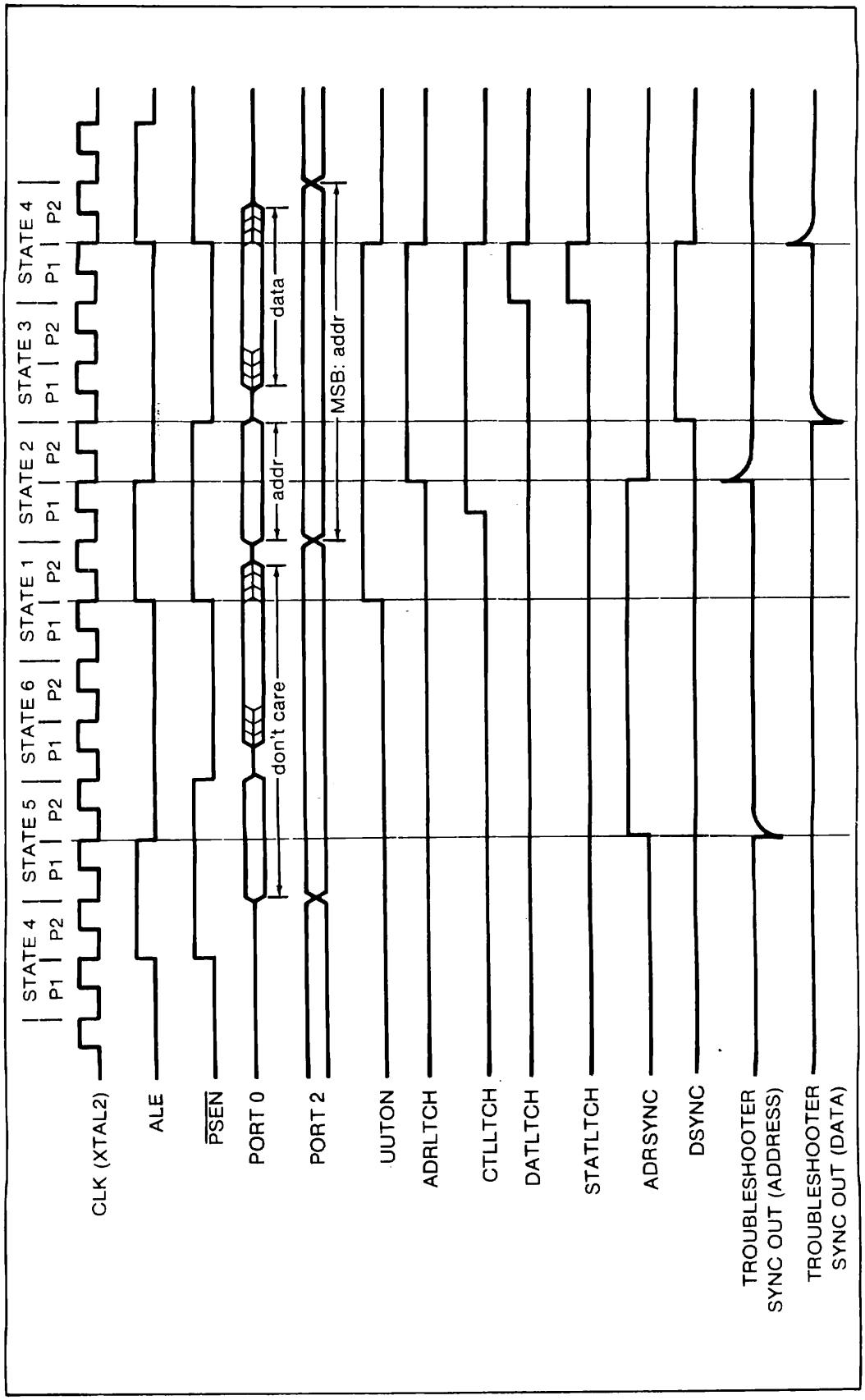
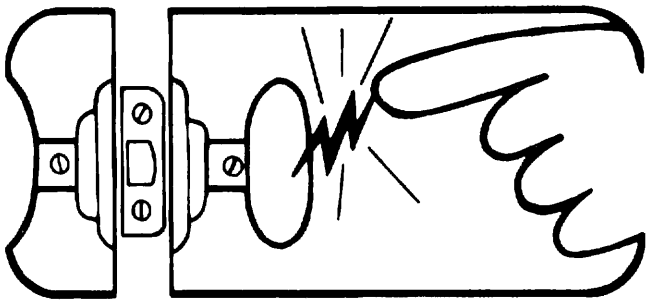


Figure 5-7. Latch Times for Data, Status, and Control During External Program Fetch

static awareness

A Message From

John Fluke Mfg. Co., Inc.

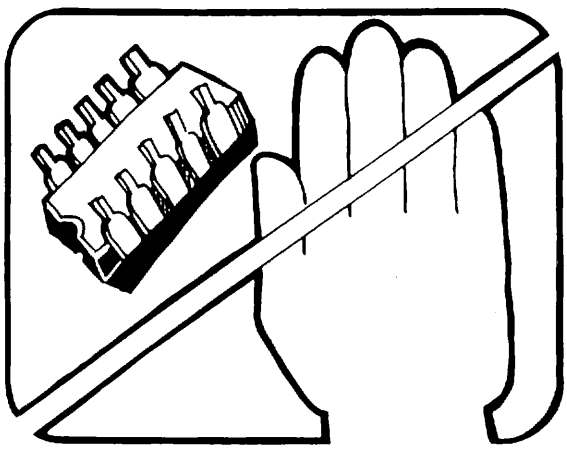


Some semiconductors and custom IC's can be damaged by electrostatic discharge during handling. This notice explains how you can minimize the chances of destroying such devices by:

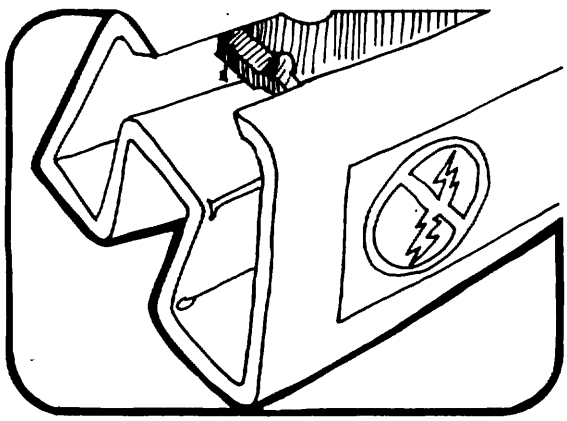
1. Knowing that there is a problem.
2. Learning the guidelines for handling them.
3. Using the procedures, and packaging and bench techniques that are recommended.

The Static Sensitive (S.S.) devices are identified in the Fluke technical manual parts list with the symbol "⊗"

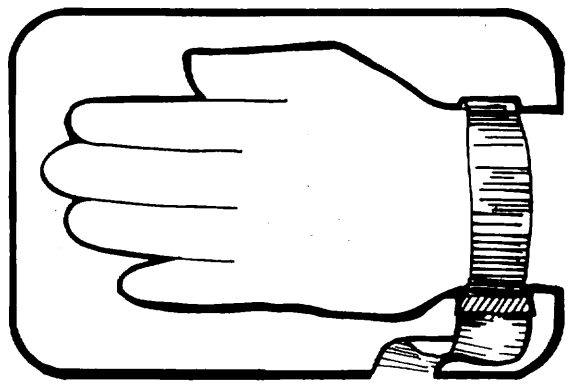
The following practices should be followed to minimize damage to S.S. devices.



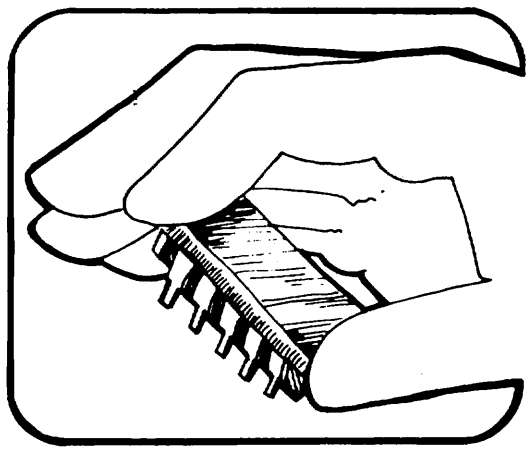
1. MINIMIZE HANDLING



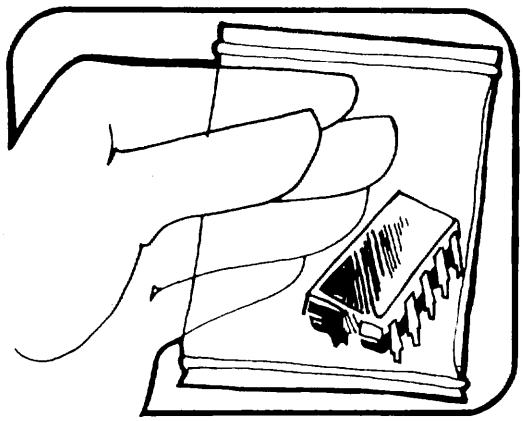
2. KEEP PARTS IN ORIGINAL CONTAINERS UNTIL *READY FOR USE.*



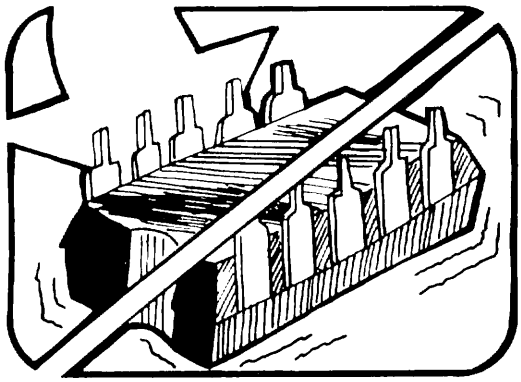
3. DISCHARGE PERSONAL STATIC BEFORE HANDLING DEVICES. USE A HIGH RESISTANCE GROUNDING WRIST STRAP.



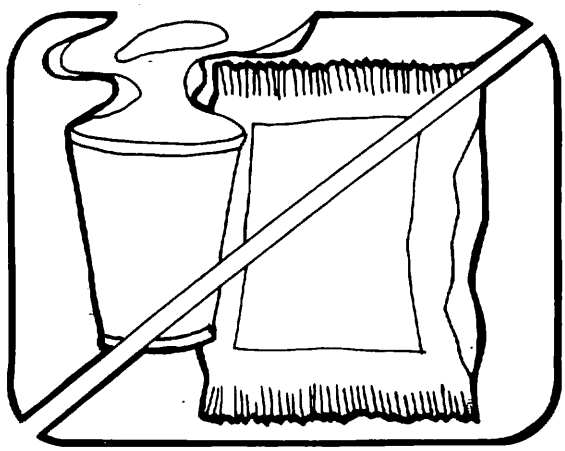
4. HANDLE S.S. DEVICES BY THE BODY



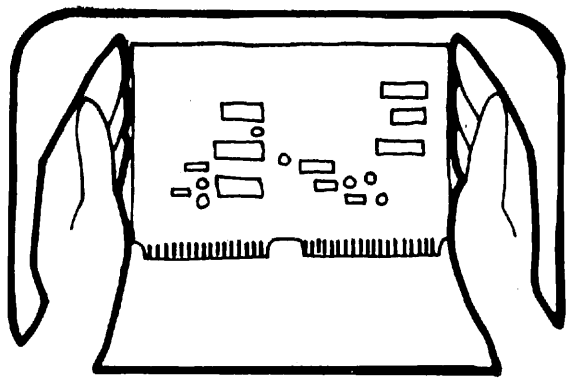
5. USE STATIC SHIELDING CONTAINERS FOR HANDLING AND TRANSPORT



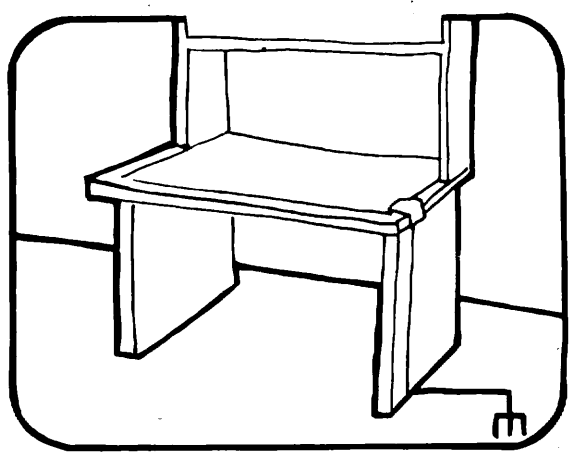
6. DO NOT SLIDE S.S. DEVICES OVER ANY SURFACE



7. AVOID PLASTIC, VINYL AND STYROFOAM® IN WORK AREA



8. WHEN REMOVING PLUG-IN ASSEMBLIES, HANDLE ONLY BY NON-CONDUCTIVE EDGES AND NEVER TOUCH OPEN EDGE CONNECTOR EXCEPT AT STATIC-FREE WORK STATION. PLACING SHORTING STRIPS ON EDGE CONNECTOR HELPS TO PROTECT INSTALLED SS DEVICES.



9. HANDLE S.S. DEVICES ONLY AT A STATIC-FREE WORK STATION
10. ONLY ANTI-STATIC TYPE SOLDER-SUCKERS SHOULD BE USED.
11. ONLY GROUNDED TIP SOLDERING IRONS SHOULD BE USED.

A complete line of static shielding bags and accessories is available from Fluke Parts Department, Telephone 800-526-4731 or write to:

JOHN FLUKE MFG. CO., INC.
PARTS DEPT. M/S 86
9028 EVERGREEN WAY
EVERETT, WA 98204

PORTIONS REPRINTED
WITH PERMISSION FROM TEKTRONIX, INC.
AND GENERAL DYNAMICS, POMONA DIV.

• Dow Chemical

J0089D-07U8604/SE EN Litho in U.S.A.
Rev. 1 MAR 86

Section 6
Troubleshooting

INTRODUCTION 6-1.

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 8051 Interface Pod is only designed to be used with a Troubleshooter 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 used with the 8044- and 8051-family microprocessors. If your Pod is demonstrating such symptoms, contact a Fluke Technical Service Center for advice.

WARRANTY AND FACTORY SERVICE 6-2.

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.

GETTING STARTED 6-3.

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 ICs or PCB assemblies by their connectors.
- Wear a static ground strap when performing repair work.
- Use conductive foam to store replacement or removed ICs.
- 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.

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

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-8051
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.

TROUBLESHOOTING A DEFECTIVE POD 6-5.

Introduction 6-6.

This section describes what to do if the Troubleshooter displays the *POD SELF TEST 8051 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.

Interpreting the Self Test Failure Codes 6-7.

INTRODUCTION 6-8.

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.

When a self test is done, two self test sequences are actually performed: the standard self test that is controlled by the Troubleshooter and the enhanced self test that is built into the Pod. The enhanced self test provides more thorough evaluation of the Pod than is provided by the standard self test.

Whenever the Pod self test is performed and the Troubleshooter displays the message *POD SELF TEST 8051 FAIL xx* where xx equals 01, 02, or 03, then the Pod failed the standard self test. Refer to Table 6-2, Standard Self Test Failure Codes. Whenever the Pod self test if performed and the Troubleshooter displays the message *POD SELF TEST 8051 FAIL 00*, then the Pod failed the enhanced self test. Refer to Recreating the Enhanced Self Test Routines in Table 6-4.

RECREATING THE STANDARD SELF TEST FAILURE CODES 6-9.

To determine which of the enhanced self test segments failed, do a read at the special address F0 0000 to find the Enhanced Self Test Failure Code. The Enhanced Self Test Failure Codes are summarized in Table 6-3.

Table 6-2. Standard Self Test Failure Codes

TEST ROUTINE/ FAILURE CODE	POD OPERATION	OPERATOR ACTIONS TO RECREATE TEST
00	Reset Pod *READ @ 0FF0=0F*	*WRITE @ F0 0028 = BF, READ @ F0 0FF0* If a power fail error message occurs, check the power detection circuits.
01	*WRITE @ 0FF0=F0*	*WRITE @ 0FF0=F0*
02	Test Control Line	BUS TEST
03	Send command to enable all Pod enable lines and verify that a Pod timeout occurs. This timeout is transparent to the operator.	Cannot recreate.

The individual portions that failed may be recreated by the user. This can make it easier to locate the source of the Pod failure. Table 6-4 contains procedures that a user may use to duplicate the enhanced self-test routines.

Make sure the Pod's ribbon cable is plugged into the pod's self test socket for all of the following operations. Begin with *WRITE @ F0 0028 = BF*. This operation prevents the Troubleshooter from initiating the self-test routine and enables the special address ranges that allow access to the internals of the Pod. Then, use the indicated procedures to recreate the failed test.

SELF TEST DIAGNOSTIC (ADDRESS F0 0000) 6-10.

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 8051 Pod in Section 4 of this manual.)

A *READ* @ operation at the special address F0 0000 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 described in Table 6-3.

A *WRITE @ F0 0028=BF* operation will disable the Pod self test and allow the Pod self test socket to function as a simple UUT. This write operation also enables the 7 xxxx, 8 xxxx, and 9 00xx internal addresses. Table 6-5 describes some of the Pod's internal addresses which may be used while Troubleshooting a Pod.

Table 6-3. Enhanced Self Test Failure Codes

ERROR CODES	DESCRIPTION
01	Program ROM checksum failed.
02	Processor RAM failed.
03	8155 RAM failed.
04	8155 I/O or timer failed.
05	State counter failed.
06	Watchdog timer failed.
07	UUTON circuit failed.
08	Status buffer failed.
09	UUT read/write test failed. Address, data, or control errors are reported.
0A	Port 0 test failed. Data errors are reported.
0B	Port 1 test failed. Data errors are reported.
0C	Port 2 test failed. Data errors are reported.
0D	Port 3 test failed. Data errors are reported.
FF	Self Test completed OK. No errors found.

Preparation for Troubleshooting a Defective Pod 6-11.

CAUTION

Any adjustment, maintenance, or repair of the opened Pod under voltage shall be avoided as far as possible and, if done, shall be carried out only by a skilled person who is aware of the hazards 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 ICs that are loose in their sockets. Replace components if necessary.

Table 6-4. Recreating the Enhanced Self Test Routines

CODE	DESCRIPTION	PROCEDURE
01	Program ROM checksum failed.	*READ @ 73FFE* and *73FFF* to obtain the ROM checksum. Then perform *ROM TEST @ 70000 - 73FFD SIG xxxx* where "*xxxx*" is the checksum you just read.
02	Processor RAM failed.	Although reads and writes may be performed in the range 90000 - 900FF to test this region, beware that most of this RAM is used by the Pod for variables, stack, etc., and that operations on it will likely affect Pod operation (eg. you may crash the stack). It is not recommended that one attempt to duplicate the RAM test in this address space.
03	8155 RAM failed.	Perform *RAM SHORT or RAM LONG @ 88000 - 8807F.*
04	8155 I/O or timer failed.	*WRITE @ 88100 = 43* and then *WRITE @ 88101* to write Port A. *READ @ 88101* to verify data written to Port A. *WRITE @ 88100 = 42* to return Port A to its input state. *READ/WRITE @ 88102* to test Port B. *READ @ 88102* to check Port C and to read DIP switch settings. *READ @ 88100* and look for data bit 6 (ie. X1XXXXXX binary), the timer flag, to verify that the timer is counting.
05	State counter failed.	This test cannot be duplicated by the Pod operating on itself.
06	Watchdog timer failed.	This test cannot be duplicated by the Pod operating on itself.
07	UUTON circuit failed.	Perform *READ @ 705A0.* Perform *READ @ 05A0.* The second Read should result in data = 05. If the data instead matches that of the first Read, then a failure is indicated.
08	Status buffer failed.	Perform *WRITE @ 400B0 = FF* and *WRITE @ 40090 = FF.* Do *READ @ STS* and verify that the $\overline{EA}$ and RST signals are high. *WRITE @ 50090 = 0* then *READ @ STS* and verify that $\overline{EA}$ = 1, RST = 0. *WRITE @ 50091 = 0* and verify that *READ @ STS* results in both $\overline{EA}$ and RST being low.

Table 6-4. Recreating the Enhanced Self Test Routines (cont)

CODE	DESCRIPTION	PROCEDURE
09	UUT read/write test failed.	Perform the following: *READ @ 5555* to check address, PSEN and ALE; *READ @ 1AAAA* to check address, RD and ALE; *WRITE @ 1C33C = 55* to check address, data, WR and ALE; and *WRITE @ 17EE7 = AA* to check address, data, WR and ALE.
0A	Port 0 test failed.	Perform the following: *WRITE @ 400A0 = FF* to make sure Port 2 is high; *WRITE @ 40080 = 55* (this will yield a drivability error, since the self-test socket does not have any pullup capacity to drive the line high); *WRITE @ 40080 = 55* again (this time, it should be OK); *READ @ 400A0; WRITE @ 40080 = AA* (this will yield a drivability error again); *WRITE @ 40080 = AA; READ @ 400A0; WRITE @ 40080 = FF; WRITE @ 400A0 = 55* to check Port 0 inputs; *READ @ 40080; WRITE @ 400A0 = AA;* and *READ @ 40080.*
0B	Port 1 test failed.	Perform the following: *WRITE @ 400B0 = FF* to make sure Port 3 is high; *WRITE @ F00051 = 00* to set up to test all Port 1 lines; *READ @ F00051* to do a Bus Test on Port 1; *WRITE @ 400B0 = 55* to check Port 1 inputs; *READ @ 40090; WRITE @ 400B0 = AA;* and *READ @ 40090.* *NOTE* *Control errors will occur if Switch 5 is set to the $\overline{RD}/\overline{WR}$ position.*
0C	Port 2 test failed.	Perform the following: *WRITE @ 40080 = FF* to make sure Port 0 is high; *WRITE @ F00052 = 00* to set up to test all Port 2 lines; *READ @ F00052* to do a Bus Test on Port 2; check that switch 2 is in the I/O position; *WRITE @ 40080 = 55* to check Port 2 inputs; *READ @ 400A0; WRITE @ 40080 = AA;* and *READ @ 400A0.*
0D	Port 3 test failed.	Perform the following: *WRITE @ 40090 = FF* to make sure Port 1 is high; *WRITE @ F00053 = 00* to set up to test all Port 3 lines; *READ @ F00053* to do a Bus Test on Port 3; *WRITE @ 40090 = 55* to check Port 3 inputs; *READ @ 400B0; WRITE @ 40090 = AA;* and *READ @ 400B0.* *NOTE* *8044-family microprocessors do not have output latches on pins P3.0 and P3.1. Do not try to write data to these pins.*

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 @ F0 0028 = BF* 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 @ F0 0028 = FF*.)

Table 6-5. Diagnostic Pod Addresses

CONTROL	LOWER ADDRESS	UPPER ADDRESS	DEVICE	DESCRIPTION
$\overline{PSEN}$	0000	3FFF	U2 (27128) EPROM	Program Memory
$\overline{RD}/\overline{WR}$	8000	80FF	U1 (8155)	256 Byte RAM
"	8100		"	Control/Status Register
"	8101		"	Port A
"	8102		"	Port B
"	8103		"	Port C
"	8104		"	Timer (Low Byte)
"	8105		"	Timer (High Bits,Mode)
"	8800		U6, U7	Port D
$\overline{RD}$	9000		*U2 (74SC374)	Port 0 Data Latch
"	9001		*U5	Port 0 Address Latch
"	9002		*U7	Port 1 Latch
"	9003		*U1	Post 2 Latch
"	9004		*U4	Port 3 Latch
"	9005		*U3	Status Latch
"	9006		*U6	Control Latch
"	9800		U24	State Counter Increment

* Devices on Interface Board

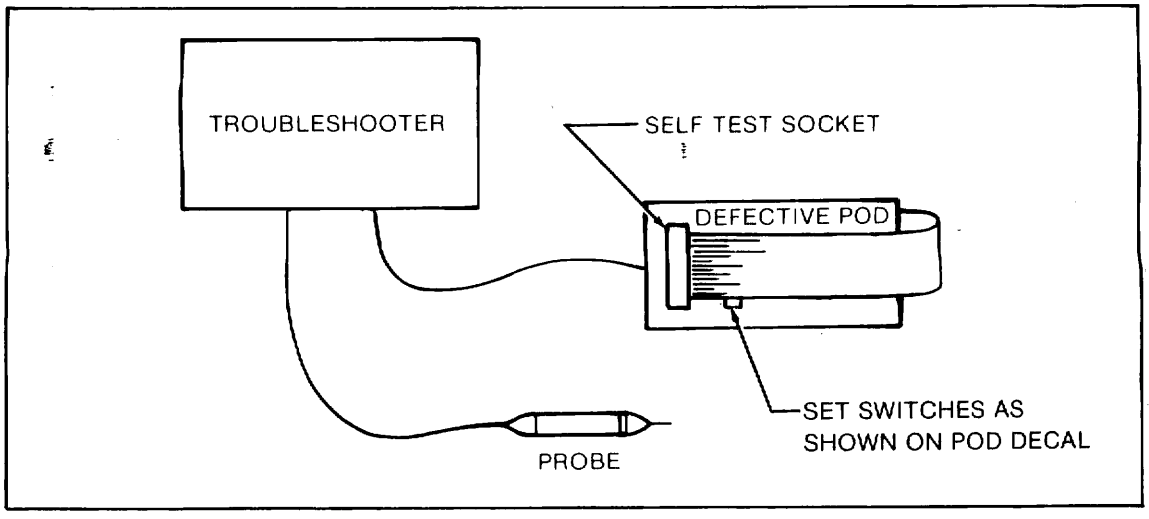


Figure 6-1. Troubleshooting a Defective Pod

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

When the Pod and the Troubleshooter are connected in this configuration (and with Special Address F0 0028 = BF 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.

TROUBLESHOOTING AN INOPERATIVE POD 6-12.

Introduction 6-13.

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 8051 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.)

Preparation for Troubleshooting an Inoperative Pod 6-14.

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 ICs 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 normally 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. 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 on connector J1 (Processor PCB) to +5V and pin 25 to ground, and pin 21 to -5 Volts.

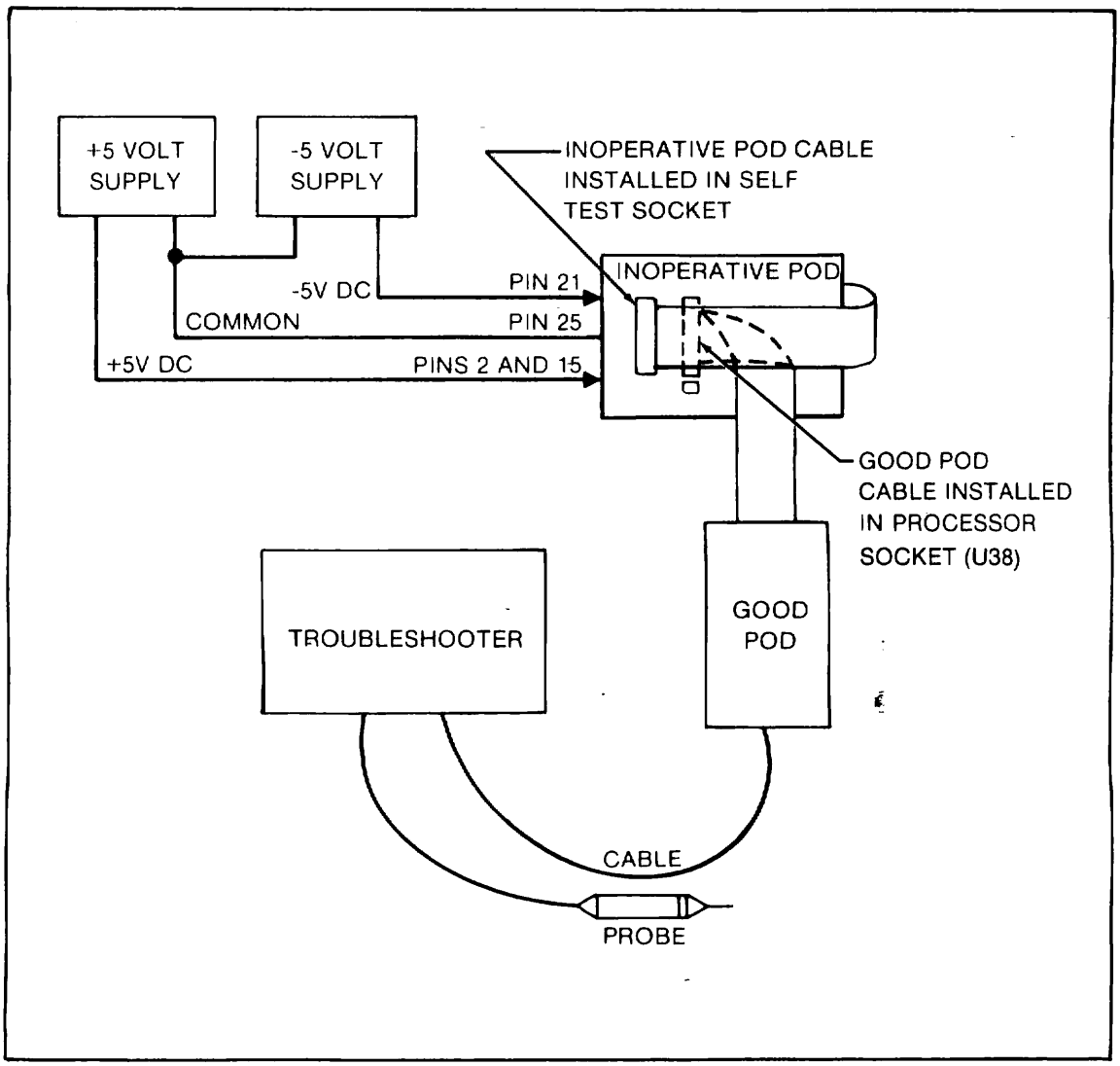


Figure 6-2. Troubleshooting an Inoperative Pod

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.

Procedure for Troubleshooting an Inoperative Pod 6-15.

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 a fault 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 processor board (the one with 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's clock signal is present at U38 pin 18. If it is not trace the clock signal to the fault. Complementary clock signals should also be present on TP1 and TP2 on the interface board.

3. Check that RST is not being held active (high) at U38 pin 9.

4. Check that the Watchdog timer is allowing a Reset from the Troubleshooter to the microprocessor. The PODRESET signal is held off if pin 13 of U31 is low. If the Troubleshooter displays any of the *ATTEMPTING RESET* messages, then the timer output U8, pin 3 should be high, and Resets from the Troubleshooter should reach the processor U38 pin 9. If they don't, trace the signal to the disabling device. Also, check for a missing clock signal on U9, pin 14 and continuous Resets to U8 pin 11 (Watch dog reset).

5. Check the address decoder (U5) by performing looping reads and checking that the corresponding outputs are low at the respective addresses.

Read Address	U5 Output Pin
1 8000	15
1 8800	14
1 9000	13
1 9800	12

6. Do a ROM test on addresses 0000 through 3FFD. The correct checksum is stored in 3FFE and 3FFF.

7. Check the operation of the state counter (U11). Pin 6 should have a pulse, occurring during every other ALE signal (TP4) for the first half of the ALE pulse. If it is not in the right place, check that a *READ @ 1 9800* increments the state counter. Do a *READ @ 1 8800* to read port D. Bit 2 is a state which should be read high when the correct state is encountered. Increment the counter and read until the bit is high.

8. Do a RAM test on addresses 1 8000 through 1 80FF to test the RAM in the 8155 Peripheral chip. If the RAM test fails, then check the internal read and internal write flip flops ($\overline{\text{IRD}}$ and $\overline{\text{IWR}}$). Do a looping *WRITE @ 1 XXXX*. U20, pin 13, should have a buffered write signal and the output pin 7 should also be active. Similarly, $\overline{\text{IRD}}$ is generated by U30. Do a looping *READ @ 1 XXXX* and see that pin 1 has a buffered read signal. Note that the output, pin 6, is gated by U26, and then inverted.

9. Check the communication with the Troubleshooter by testing the ports on the 8155. If another Troubleshooter is providing power to the inoperative Pod, the value AB should be on POD0-7, U1 pins 21-28. First *WRITE @ 1 8100=42* to set up the 8155's control register, then *READ @ 1 8101* to verify the correct value.

10. Check input and output for Port D. Write an arbitrary value to the port with a *WRITE @ 1 8800 = XX* operation and see that U7, pins 2, 5, 6, 9, 12, 15, 16, and 19, contain the same value. A *READ @ 1 8800* operation should return the value on U6, pins 2-9.

11. Check that UUTON (TP5) is low. If not, see that U19, pin 12, is receiving the ALE clock, and that pin 13 (K) has a negative pulse on the rising edge of ALE.

EXTENDED TROUBLESHOOTING PROCEDURES 6-16.

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.

Cable Lines 6-17.

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.

Timing Problems 6-17.

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.

Noise Problems 6-18.

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.

RUN UUT Limitations 6-19.

If a Pod seems to perform testing functions properly but does not perform RUN UUT operations properly, the problem may be due to limitations of the Pod design.

The Pod will not perform RUN UUT operations correctly with UUTs that use Port 0 or Port 2 as both expansion buses and as I/O ports. The Pod relies on the configuration of these lines as specified by the configuration switches and is unable to accommodate both modes during the course of a RUN UUT.

The Pod also does not support the programming or program verification modes of the 8751 and 8744 microprocessors.

OSCILLATOR CIRCUIT ADJUSTMENT PROCEDURE 6-20.

The oscillator circuit has a potentiometer (R29 on the Interface PCB Assembly) which controls the adjustment of the clock signal to the microprocessor. The following steps describe the procedure for performing the adjustment. This adjustment should be performed whenever components have been changed in the oscillator circuit.

1. Remove power from the Troubleshooter. Refer to Disassembly and disassemble the Pod (the two PCB assemblies do not need to be separated).

2. Locate Potentiometer R29 on the Interface PCB assembly (shown in Figure 6-3) and turn R29 fully counter-clockwise.

3. Insert the Pod ribbon cable into the self test socket. The self test socket will provide the crystal source for the oscillator circuit.

4. Turn on the Troubleshooter power.

5. Place an oscilloscope probe as indicated in Figure 6-3 (TP1 or TP2 on the Interface PCB assembly).

6. Slowly turn R29 clockwise until the signal as shown on the oscilloscope is +5.0V p-p (0 volt baseline).

SIMULATING POWER UP RESET 6-21.

The following sequence resets the Pod in the same manner as the power up reset. Note that this operation results in all special address masks being reset to their default values.

WRITE @ F00028 = BF
WRITE @ 90032 = 0
BUS TEST

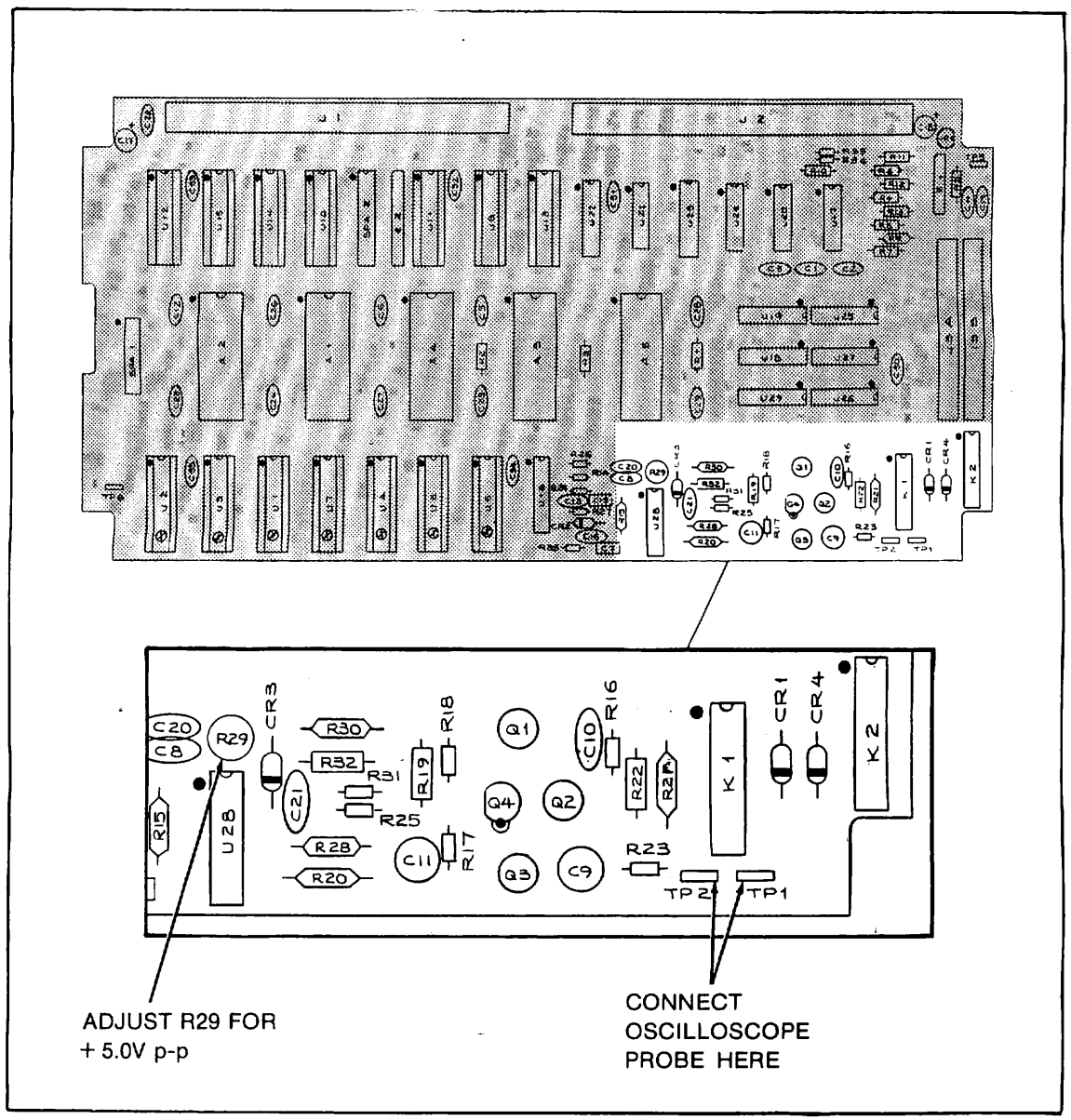


Figure 6-3. Oscillator Circuit Adjustment Location

DISASSEMBLY

6-22.

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 six Phillips screws at the corners and center of the PCB assemblies and carefully pull the boards apart at the two connectors along the side.

Section 7
List of Replaceable Parts

INTRODUCTION 7-1.

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.

HOW TO OBTAIN PARTS 7-2.

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

Items on the parts lists indicated by an asterisk (*) are subject to damage by static discharge.

MANUAL CHANGE AND BACKDATING INFORMATION 7-3.

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. 9000A-8051 FINAL ASSEMBLY
(SEE FIGURE 7-1.)

REFERENCE DESIGNATOR A->NUMERICS--->	S	----------DESCRIPTION----------	FLUKE STOCK --NO--	MFRS SPLY CODE-	MANUFACTURERS PART NUMBER --OR GENERIC TYPE--	TOT QTY	R S -Q	NOT -E
A 38	*	PROCESSOR PCB ASSEMBLY	744722	89536	744722	1		
A 39	*	INTERFACE PCB ASSEMBLY	744730	89536	744730	1		
H 1		SCREW,MACHINE,PHP,STEEL,4-40X3/4	115063	89536	115063	4		
H 2		SCREW,MACHINE,PHP SEMS,STEEL,4-40X1/4	185918	89536	185918	6		
H 3		SCREW,MACHINE,PHP,STEEL,4-40X5/8	145813	89536	145813	1		
H 4		WASHER,LOCK,INTRNL,STEEL,#4	110403	89536	110403	1		
MP 1		LABEL, STATIC CAUTION	605808	89536	605808	1		
MP 2		DECAL, POD	744649	89536	744649	1		
MP 3		DECAL, SPEC	744656	89536	744656	1		
MP 4		WARNING DECAL 8048	659813	89536	659813	1		
MP 5		DECAL, SWITCHING	744664	89536	744664	1		
MP 6		SHELL, TOP	653113	89536	653113	1		
MP 7		SHELL, BOTTOM	648881	89536	648881	1		
MP 8		ACTUATOR	582916	89536	582916	1		
MP 9		SHIELD, ALUM. MYLAR	659771	89536	659771	1		
MP 10		SLIDE COVER	653139	89536	653139	1		
MP 11		SPACER,HEX,ALUM,4-40X0.375	187575	89536	187575	1		
P 1		CABLE, POD 8086/88	607184	89536	607184	1		
TM 1		INSTRUCTION MANUAL	744698	89536	744698	1		
TM 2		REFERENCE CARD	744839	89536	744839	1		
W 1		UUT CABLE, SHIELDED	680595	89536	680595	1		

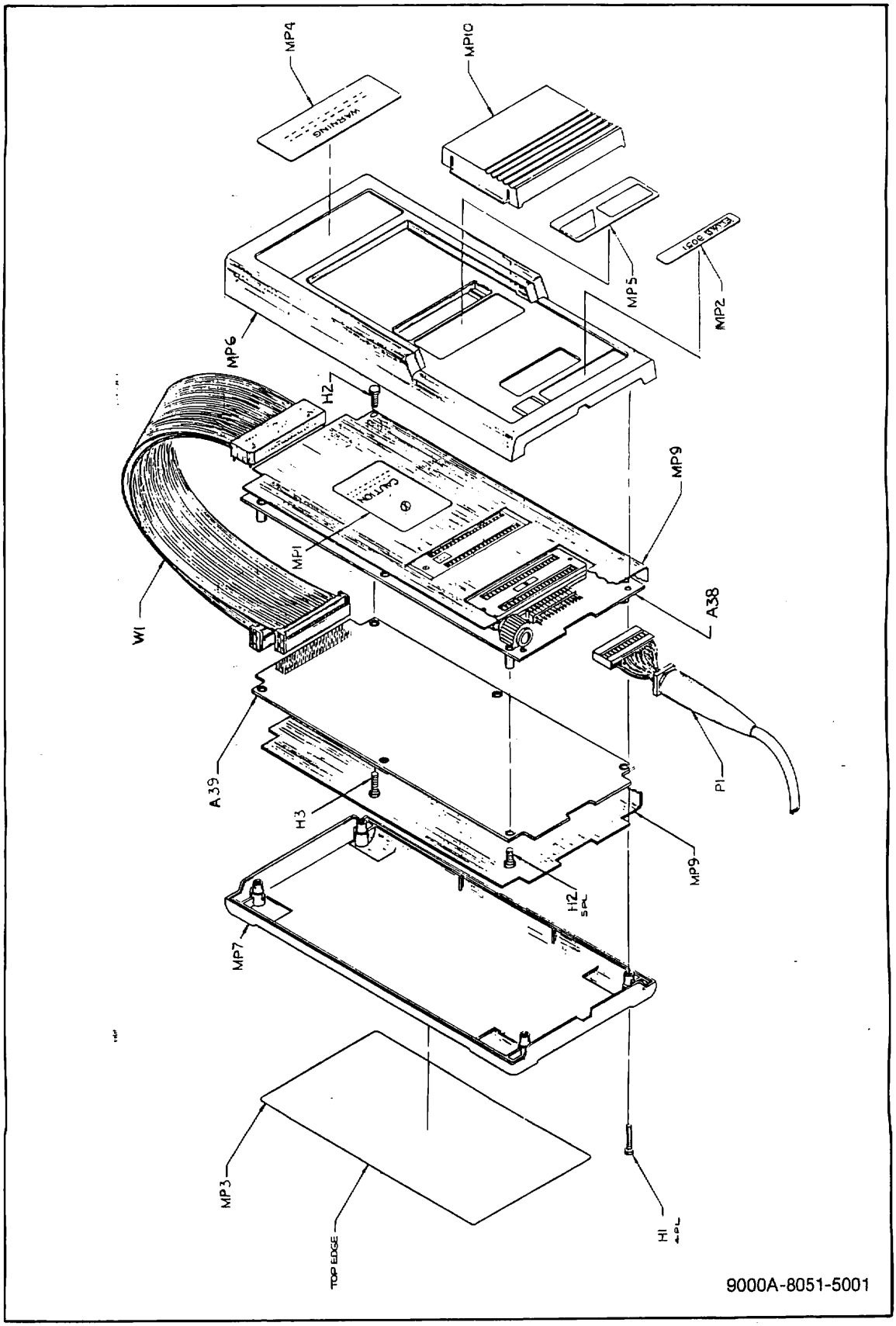


Figure 7-1. 9000A-8051 Final Assembly

TABLE 7-2. A38 PROCESSOR PCB ASSEMBLY
(SEE FIGURE 7-2.)

REFERENCE DESIGNATOR A->NUMERICS---->	S	----------DESCRIPTION----------	FLUKE STOCK --NO--	MFRS SPLY CODE-	MANUFACTURERS PART NUMBER --OR GENERIC TYPE--	TOT QTY	R S -Q	NOTE
C 1, 19		CAP,AL,10UF,+-20%,16V	602326	89536	602326	2		
C 2- 15, 17,		CAP,CER,0.22UF,+-20%,50V,Z5U	309849	71590	CW3C0C224K	16		
C 18			309849					
C 16		CAP,CER,100PF,+-2%,100V,COG	512848	51406	RPE121	1		
CR 1	*	DIODE,SI,BV= 75.0V,IO=150MA,500 MW	203323	07910	1N4448	1	1	
H 1		SPACER,SWAGED,RND,BRASS,4-40X0.340	380329	89536	380329	6		
J 1		CONN,PWB,HEADER,DIP,90,0.100,26 PIN	512590	89536	512590	1		
J 2		SOCKET, 40 PIN ZIF	585133	89536	585133	1		
J 3		SOCKET,DIP,0.100 CTR,ZIF,40 PIN	524124	89536	524124	1		
P 1, 2		CONN,POST,PWB,.025SQ,NON-INSUL,SELECT	267500	00779	87022-1	100		
Q 1, 2, 3	*	TRANSISTOR,SI,PNP,SMALL SIGNAL	195974	64713	2N3906	3	1	
R 1, 8		RES,CF,4.7K,+-5%,0.25W	348821	01121	CB4725	2		
R 3		RES,CF,39,+-5%,0.25W	340836	80031	CR251-4-5P39E	1		
R 4		RES,CF,470,+-5%,0.25W	343434	80031	CR251-4-5P470E	1		
R 5, 13		RES,CF,2.2K,+-5%,0.25W	343400	80031	CR251-4-5P2K2	2		
R 6		RES,CF,10K,+-5%,0.25W	348839	80031	CR251-4-5P10K	1		
R 7		RES,CF,1K,+-5%,0.25W	343426	80031	CR251-4-5P1K	1		
R 9		RES,MF,182,+-1%,0.125W,100PPM	289579	91637	CMF55	1		
R 10, 11		RES,CC,30K,+-5%,0.125W	740241	89536	740241	2		
R 12		RES,CF,100,+-5%,0.25W	348771	80031	CR251-4-5P100E	1		
R 14		RES,MF,806,+-1%,0.125W,100PPM	223552	91637	CMF550806F	1		
R 15		RES,MF,200,+-1%,0.125W,100PPM	245340	91637	CMF552000F	1		
R 16		RES,CF,100K,+-5%,0.25W	348920	80031	CR251-4-5P100K	1		
S 1		SWITCH MODULE, 8-POS	414490	00779	435166-5	1		
TP 1, 2, 4,		CONN,TAB,FASTON,PRESS-IN,0.110 WIDE	512889	02660	62395	4		
TP 5			512889					
U 1	*	IC,NMOS,256 X 8 W/Z I/O PORTS & TIMER	536847	34649	P8155-2	1	1	
U 2	*	27128, PROGRAMMED V1.0	734061	89536	734061	1		
U 3	*	IC,LSTTL,OCTAL D TRANSPARENT LATCHES	504514	01295	SN74LS373N	1	1	
U 4	*	IC,ALSTTL,OCTAL BUS XCVR W/3-STATE	647214	01295	SN74ALS245N	1	1	
U 5	*	IC,LSTTL,3-8 LINE DCDR W/ENABLE	407585	01295	SN74LS138N	1	1	
U 6	*	IC,LSTTL,OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	1	1	
U 7	*	IC,LSTTL,OCTAL D F/F,+EDG TRG,W/CLEAR	454892	01295	SN74LS273N	1	1	
U 8	*	IC,CMOS,14STAGE RIPPLE CARRY BIN CNTR	394486	02735	CD4020AE	1		
U 9	*	IC,LSTTL,DIV BY 2,DIV BY 8 COUNTER	525204	01295	SN74LS93N	1	1	
U 10, 11	*	IC,TTL,8BIT SER-IN,PAR-OUT R-SHFT RGS	408732	01295	SN74LS164N	2	1	
U 12	*	16L8, PROGRAMMED V1.0	734079	89536	734079	1		
U 13	*	16L8, PROGRAMMED V1.0	734087	89536	734087	1		
U 14	*	IC,FTTL,OCTAL TRNSPRNT LTCH W/3-STATE	686238	89536	686238	1	1	
U 15, 27, 34	*	IC,LSTTL,QUAD 2 INPUT OR GATE	393108	01295	SN74LS32N	3	1	
U 16	*	IC,TTL,QUAD 2 INPUT AND GATE	393066	01295	SN74LS08N	1	1	
U 17, 35	*	IC,LSTTL,HEX INVERTER	393058	01295	SN74LS04N	2	1	
U 18, 26	*	IC,LSTTL,QUAD 2 INPUT NOR GATE	393041	01295	SN74LS02N	2	1	
U 19, 24	*	IC,LSTTL,DUAL J-K F/F,+EDG TRIG	412999	01295	SN74LS109N	2	1	
U 20, 21, 30	*	IC,LSTTL,DUAL JK F/F,-EDG TRIG	414029	01295	SN74LS112N	3	1	
U 22	*	IC,STTL,TRIPLE 3 INPUT AND GATE	363457	01295	SN74S11N	1	1	
U 23, 32	*	IC,LSTTL,QUAD 2 INPUT NAND GATE	393033	01295	SN74LS00N	2	1	
U 25	*	IC,TTL,TRIPLE 3 INPUT NAND GATE	363465	01295	SN7412N	1	1	
U 28	*	IC,LSTTL,QUAD SET/RESET LATCH	404210	01295	SN74LS279N	1	1	
U 29	*	IC,LSTTL,HEX INVERTER W/SCHMT TRIG	483180	01295	SN74LS14N	1	1	
U 31	*	IC,FTTL,QUAD 2 INPUT AND GATE	650523	07263	74F08PC	1	1	
U 33	*	IC,LSTTL,QUAD BUFFER W/3-STATE OUTPUT	585273	01295	SN74LS126N	1	1	
U 36	*	IC,LSTTL,HEX BUFFER W/NOR ENABLE	483800	01295	SN74LS367N	1	1	
U 37	*	IC,COMPARATOR,HI-SPEED,14 PIN DIP	386920	18324	NE529A	1	1	
U 38	*	IC,NMOS,256 X 8 W/Z I/O PORTS & TIMER	536847	89536	536847	1		
XU 1		SOCKET,DIP,0.100 CTR,40 PIN	429282	09922	DILB40P-108	1		
XU 2		SOCKET,DIP,0.100 CTR,28 PIN	448217	91506	328-AG39D	1		
XU 3, 4, 6,		SOCKET,DIP,0.100 CTR,20 PIN	454421	09922	DILB20P-108	6		
XU 7, 12, 13			454421					
XU 8		SOCKET,DIP,0.100 CTR,16 PIN	276535	91506	316-AG39D	1		
Y 1		OSCILLATOR,6.5MHZ,TTL CLOCK	586933	89536	586933	1	1	
Z 1		RES,NET,DIP,16 PIN,8 RES,1K,+-5%	358119	01121	314	1		
Z 2		RES,NET,SIP,10 PIN,9 RES,10K,+-2%	414003	80031	95081002CL	1		
Z 3		RES,NET,SIP,10 PIN,9 RES,2K,+-2%	446880	89536	446880	1		

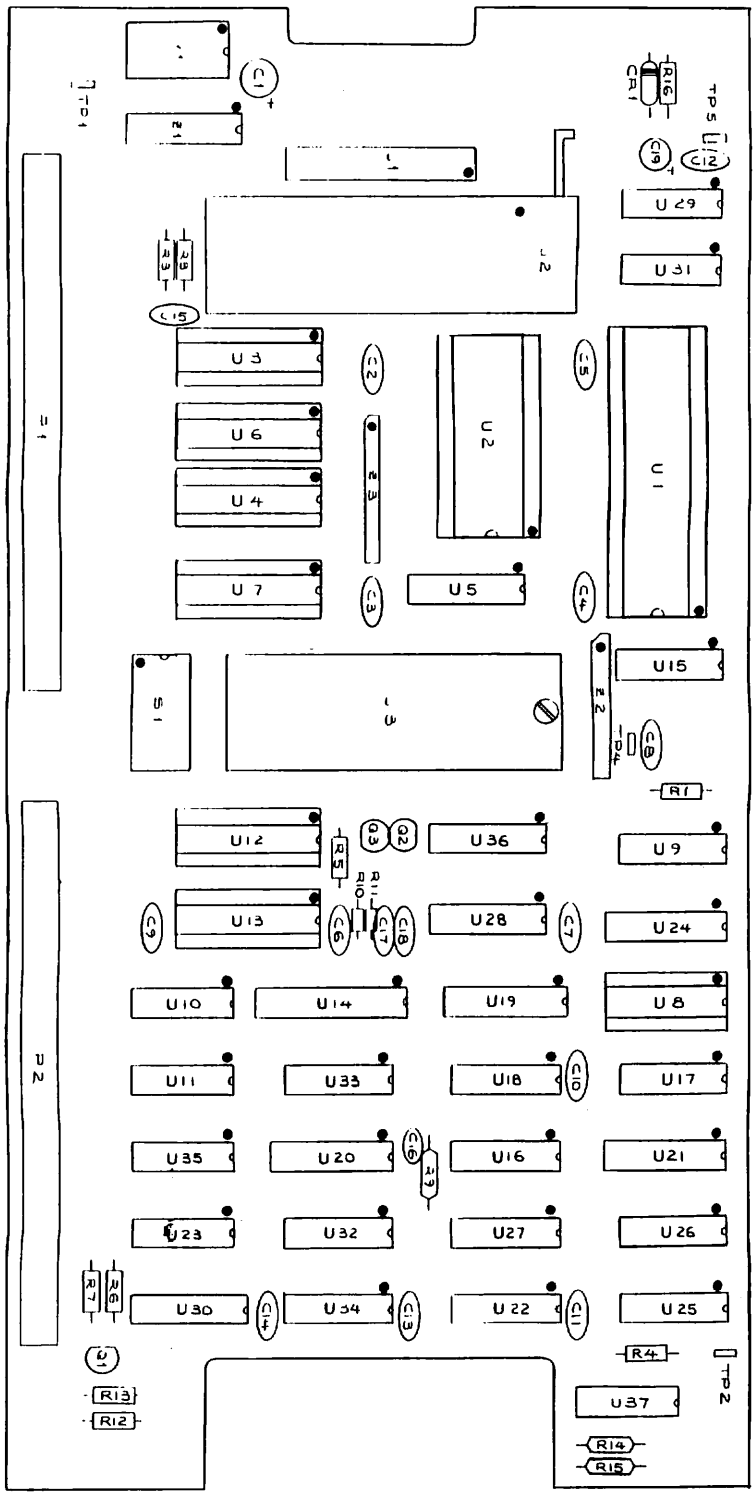


Figure 7-2. A38 Processor PCB Assembly

TABLE 7-3. A39 INTERFACE PCB ASSEMBLY
(SEE FIGURE 7-3.)

REFERENCE DESIGNATOR A->NUMERICS--->	S	DESCRIPTION	FLUKE STOCK --NO--	MFRS SPLY CODE-	MANUFACTURERS PART NUMBER ---OR GENERIC TYPE---	TOT QTY	R S -Q	NOT -E
A 1, 2, 5	*	HYBRID, 700(, TESTED	582189	89536	582189	3		
A 3, 4	*	HYBRID, 3K(, TESTED	582247	89536	582247	2		
C 1, 3, 4,		CAP,CER,0.22UF,+-20%,50V,Z5U	309849	71590	CW3C0C224K	27		
C 5, 6, 8,			309849					
C 10, 12, 13,			309849					
C 16, 19- 35			309849					
C 7, 15		CAP,CER,47PF,+-2%,100V,C0G	512368	89536	512368	2		
C 9, 11, 17,		CAP,TA,10UF,+-20%,15V	193623	56289	196D106X0015A1	4		
C 18			193623					
CR 1- 4	*	DIODE,SI,BV= 75.0V,IO=150MA,500 MW	203323	07910	1N4448	4	1	
J 1, 2		CONN,PWB,SCKT HOUSING,DIP,0.150,50POS	649848	00779	86396-5	2		
J 3		CONN,PWB,HEADE3R,DIP,0.100,40 PIN	603670	89536	603670	2		
K 1, 2		RELAY,ARMATURE,2 FORM C,5V	733063	89536	733063	2		
L 1		CHOKE CORE, 4B	321182	89536	321182	1		
Q 1	*	TRANSISTOR,SI,NPN,HI-FREQ,SMALL SIGNL	454231	04713	BRF90	1	1	
Q 2, 3	*	TRANSISTOR,SI,NPN,HI-FREQ,SMALL SIGNL	659417	04713	BRF96	2	1	
Q 4	*	TRANSISTOR,SI,NPN,SMALL SIGNAL	330803	07263	MPS6560	1	1	
R 1		RES,CF,12K,+-5%,0.25W	348847	80031	CR251-4-5P12K	1		
R 2, 3		RES,CF,22K,+-5%,0.25W	348870	80031	CR251-4-5P22K	2		
R 4, 13		RES,CF,2K,+-5%,0.25W	441469	80031	CR251-4-5P2K	2		
R 5, 6, 11		RES,CF,2.2K,+-5%,0.25W	343400	80031	CR251-4-5P2K2	3		
R 7, 8, 12		RES,CF,1K,+-5%,0.25W	343426	80031	CR251-4-5P1K	3		
R 9		RES,MF,2.43K,+-1%,0.125W,100PPM	312637	91637	CMF552431F	1		
R 10		RES,CF,16K,+-5%,0.25W	442376	80031	CR251-4-5P16K	1		
R 14		RES,CC,100,+-5%,0.125W	714469	89536	714469	1		
R 15, 30		RES,MF,15.4,+-1%,0.125W,100PPM	321299	91637	CMF5515R4F	2		
R 16, 17		RES,CC,10,+-5%,0.125W	654970	89536	654970	2		
R 18, 23, 31		RES,CC,1K,+-5%,0.125W	643932	01121	BB1025	3		
R 19, 22		RES,CF,2,+-5%,0.25W	442053	80031	CR251-4-5P2E	2		
R 20, 21		RES,MF,61.9,+-1%,0.25W,100PPM	659052	91637	CMF5561R9F	2		
R 24, 33		RES,CC,10K,+-5%,0.125W	643940	01121	BB1035	2		
R 25		RES,CC,75,+-5%,0.125W	721035	89536	721035	1		
R 26		RES,CC,300,+-5%,0.125W	512772	01121	BB3015	1		
R 27		RES,CC,100K,+-5%,0.125W	643957	89536	643957	1		
R 28		RES,MF,301,+-1%,0.125W,100PPM	267740	91637	CMF553010F	1		
R 29		RES,VAR,CERM,200,+-20%,0.5W	226050	80031	ET50W201	1		
R 32		RES,CF,220,+-5%,0.25W	342626	80031	CR251-4-5P220E	1		
R 34, 35		RES,CC,47,+-5%,0.125W	512061	01121	BB4705	2		
R 36, 37		RES,CF,4.7K,+-5%,0.25W	348821	01121	CB4725	2		
TP 1, 2, 5,		CONN,TAB,FASTON,PRESS-IN,0.110 WIDE	512889	02660	62395	4		
TP 6			512889					
U 1- 7	*	IC,CMOS,OCTAL D F/F,+EDG TRG,3-STATE	707695	89536	707695	7	1	
U 8, 10	*	IC,ALSTTL,OCTAL BUS XCVR W/3-STATE	647214	01295	SN74ALS245N	2	1	
U 11, 12	*	IC,LSTTL,OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	2	2	
U 13, 14	*	IC,FTTL,OCTAL TRNSPRNT LTCH W/3-STATE	686238	89536	686238	2	1	
U 15	*	IC,LSTTL,OCTAL BUS XCVR W/OPEN COL	723296	89536	723296	1	1	
U 16	*	IC,LSTTL,3-8 LINE DCDR W/ENABLE	407585	01295	SN74LS138N	1	1	
U 17	*	IC,COMPARATOR,QUAD,14 PIN DIP	387233	12040	LM339N	1	1	
U 18	*	IC,LSTTL,TRIPLE 3 INPUT NAND GATE	393074	01295	SN74LS10N	1	1	
U 19, 27	*	IC,LSTTL,QUAD 2 INPUT NAND GATE	393033	01295	SN74LS00N	2		
U 20	*	IC,TTL,QUAD 2 INPUT AND GATE	393066	01295	SN74LS08N	1	1	
U 21, 24	*	IC,LSTTL,QUAD BUFFER W/3-STATE OUTPUT	585273	01295	SN74LS126N	2	1	
U 22, 23	*	IC,LSTTL,QUAD SET/RESET LATCH	404210	01295	SN74LS279N	2	1	
U 25	*	IC,FTTL,QUAD 2 INPUT XOR GATE	707273	89536	707273	1	1	
U 26	*	IC,CMOS,HEX SCHMITT TRIGGER	723320	89536	723320	1	1	
U 28	*	IC,ARRAY,5 TRANS,NPN,3 ISO,2 DIFF CON	248906	12040	LM3046N	1	1	
U 29	*	IC,LSTTL,QUAD 2 INPUT NOR GATE	393041	01295	SN74LS02N	1	1	
VR 1	*	IC, 1.22V,100 PPM T.C.,BANDGAP REF	452771	89536	452771	1	1	
XU 1- 8, 10-		SOCKET,DIP,0.100 CTR,20 PIN	454421	09922	DILB20P-108	14		
XU 15			454421					
XVR 1		SPACER,MOUNT,NYLON,	175125	89536	175125	1		
Z 1	*	RES NET THICK FILM ASSY,TESTED-9000	583476	89536	583476	1	1	
Z 2		RES,NET,SIP,10 PIN,9 RES,22K,+-2%	574442	89536	574442	1		

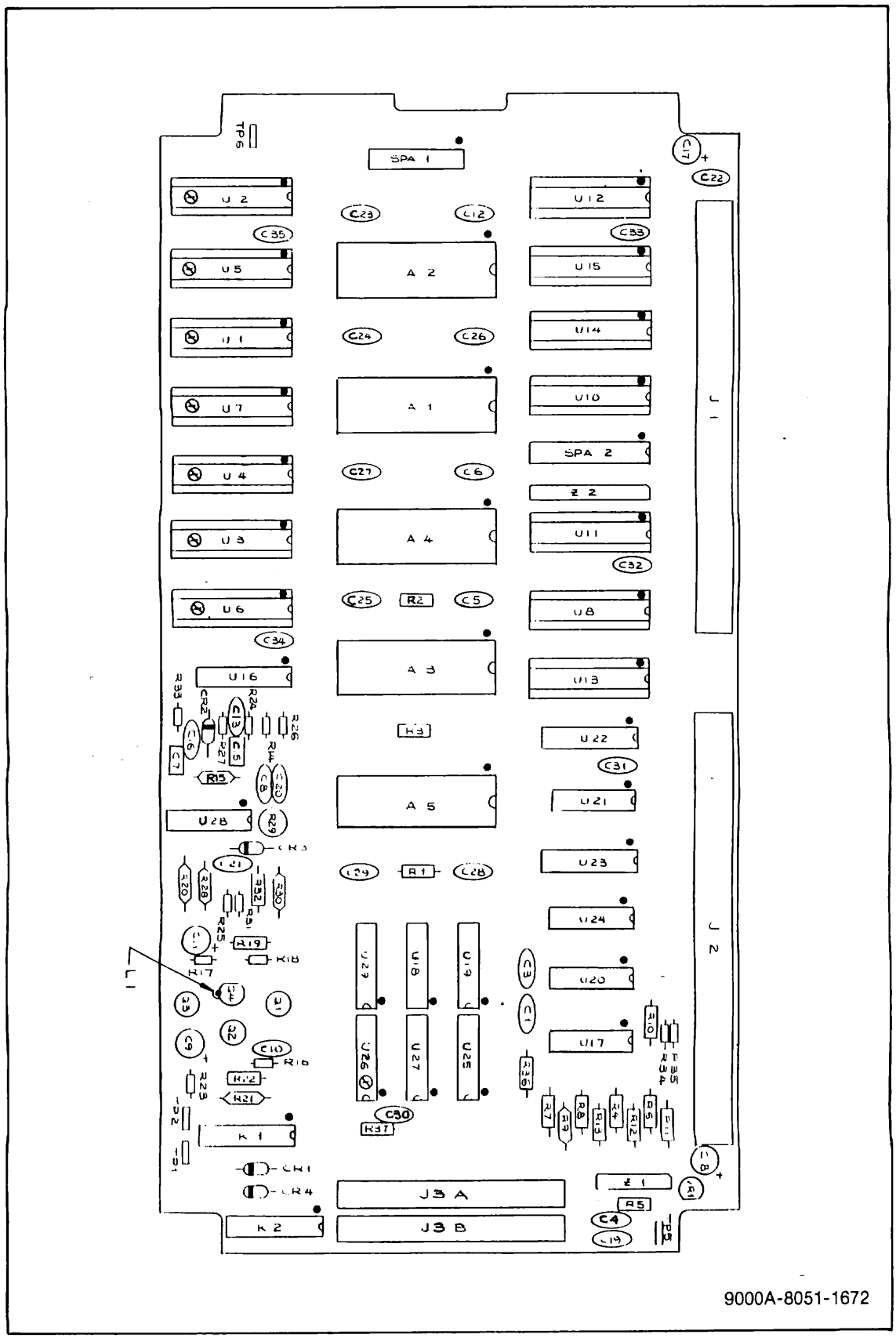


Figure 7-3. A39 Interface PCB Assembly

Table 7-4. 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	
A38	Processor PCB	744722	X															
A39	Interface PCB	744730	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.

Appendix A
Compiled Programs For the 8051 Pod

USING THE 9010A LANGUAGE COMPILER PROGRAM A-1.

Introduction A-2.

The 9010A Language Compiler is a microcomputer program that creates test programs for the Troubleshooter. It creates these test programs from source files that are created and edited on the microcomputer. 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 a Fluke Sale Office for information about the 9010A Language Compiler.

Before using the compiler to create 9010A programs for use with the 8051 Pod, you will need to create a new Pod data file. The simple procedure that is used to create the Pod data file is listed below.

Creating a New Pod Data File A-3.

A Pod data file is a simple ASCII file that you create using the text editor on your host computer system. The procedure is a follows:

1. Using the editor, create a new file named 80???.POD.
2. Copy one of the following groups into the file. Use the group that corresponds to the way that the Pod's mode switches are set (see the Note below).

If your UUT uses an external 16-bit address (called the "8031 mode", regardless of the microprocessor used), copy:

```
!
! 8031 Pod data file
! (used with 16-bit address external data memory)
!
FORCELN unused = 1
busadr = 10000
uutadr = 0000
```

If your UUT uses no external RAM memory (switch 5 set to "I/O", called the "8051 mode"), copy:

```
!
! 8051 Pod data file
! (used with no external RAM)
!
FORCELN unused = 1
busadr = 30000
uutadr = 0000
```

If your UUT uses 8-bit external RAM (the "8051X mode"), copy:

```
!
! 8051X Pod data file
! (used with 8-bit address external data memory)
!
FORCELN unused = 1
busadr = 20000
uutadr = 0000
```

3. Save this new file as file either 8031.POD, 8051.POD, or 8051X.POD on the disk, depending upon which text group you used.

NOTE

The 8051 Pod has several possible BUSTEST addresses, depending on the configuration switches. If both Port 0 and Port 2 are used as bus, and P3.6 and P3.7 are used as $\overline{RD}$ and $\overline{WR}$ respectively, then a BUSTEST address of 1 0000 is used and the Pod is considered to be in the 8031 mode. If both P3.6 or P3.7 are used as I/O, then the BUSTEST is performed at the internal address of 3 0000 and the Pod is said to be in the 8051 mode. If Port 2 is not declared as addresses, Port 0 is a bus, and RD and WR are used, then BUSTEST begins at 20000, the 8 bit address space. This last variation is the 8051X mode.

Using the Pod Data File A-4.

The new Pod data file can now be used with the compiler as described in the 9010A Language Compiler manual.

Verifying the Pod Data File A-5.

The VERIFY program (which is supplied on the 9LC disk with the 9010A Language Compiler) verifies the integrity of files on the disk. It is used to detect files which have been corrupted, which it does by calculating a checksum for each file and comparing that checksum to the one contained in the VERIFY.DAT file (also on the 9LC disk).

If you would like to add your new Pod data file to the list of files that are checked by the VERIFY program, do the following steps:

1. Edit file VERIFY.DAT (supplied on the 9LC disk) and add the following line to the end of the file:

   80???.POD DDDD

   80???.POD is the name of the new pod data file and DDDD is a dummy checksum for the file. (You'll replace the dummy checksum with a real one later.)

2. Save the modified VERIFY.DAT file on the disk.

3. Run the VERIFY program. The last two messages that it reports should be:

   File 80???.POD error - signature is CCCC, should be DDDD

   zz files tested - 1 bad signatures, 0 missing files

   80???.POD is the name of the new Pod data file, CCCC is the correct checksum for the Pod data file, and zz is the number of files tested.

4. Write down the correct checksum for the Pod data file (CCCC).

5. Re-edit the file VERIFY.DAT and replace the dummy checksum that you entered before (DDDD) with the correct checksum (CCCC).

6. Run the VERIFY program again to confirm that all changes have been made satisfactorily. The last two messages that it reports should now be:

   File 80???.POD verified

   zz files tested -- no errors

A USEFUL QUICK TEST PROGRAM A-6.

The compiler source program listed in Figure A-1 will make the use of the 8051 Pod's Quick Functions appear to operate 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 the normal built-in tests, rather than by writing information to several special addresses.

NOTE

The program is also shown in standard form in Figure A-1. That program may be used on any Troubleshooter without the compiler. It is entered line-by-line as shown, then saved on magnetic tape. Refer to the 9010A Programmers Manual if you need help.

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 available Quick Functions, then prompts for address information to use with the selected test.

```
                SAMPLE SOURCE FILE - QUICK TEST UTILITY

include "8031.pod"              ! The Pod data file has bustest address are used
        DECLARATIONS            ! 10000 only if Port 0 & 2 and RD/WR
        assign rega to key
        assign regb to incr
        assign reg8 to highaddr
        assign reg9 to lowaddr

SETUP INFORMATION
        pod - 8031

program Main
        dpy QUICK 8051 OPERATIONS                ! scroll through tests
        execute delay
ramtest:
        dpy QUICK RAM TEST (Y-N)?key
        if key = 0 goto romtest
        execute quickram                         ! do the RAM test
romtest:
        dpy QUICK ROM TEST (Y-N)?key
        if key = 0 goto fill
        execute quickrom                         ! do the ROM test
```

Figure A-1. Compiler Source Program

```
fill:
        dpy QUICK FILL OR VERIFY (Y-N)?key
        if key = 0 goto ramtest
        execute filltest                        ! do the fill or verify

program quickram                                ! Quick RAM test
        dpy ADDRESS INCREMENT? /key
        incr = key shl 4
        incr = incr or 1
        dpy BEGINNING ADDRESS? /key
        lowaddr = key or 200000                 ! mask in special 2XXXXX address
        dpy ENDING ADDRESS? /key                ! for Quick RAM test
        highaddr = key or 200000
        write @ lowaddr = 0                     ! Start Ramtest at beginning
        write @ highaddr = incr                 ! write at ending address

stat_lp:
        read @ ADR                              ! READ @ ENTER
        if DAT and F0 = F0 goto ram_err         ! if an error occurs
        if DAT and FF = C0 goto ram_ok          ! if ramtest passes
        if DAT and F0 = A0 goto abort           ! if test is aborted
        dpy BUSY                                ! else display status
        execute delay
        goto stat_lp
abort:
        dpy TEST ABORTED, ILLEGAL ENTRY
        goto ram_end
ram_err:                                        ! get address of error
        read @ F0200A                           ! byte 3 of address
        reg1 = DAT shl 8                        ! move to MSB
        read @ F02009                           ! byte 1 of address
        reg1 = reg1 or DAT shl 8                ! move to middle byte
        read @ F02008                           ! get byte 0 of address
        reg1 = reg1 or DAT
        read @  F020F0                          ! get error code
        if DAT = F0 goto rdwr_err
        if DAT = F1 goto dcd_err
        goto stat_lp
rdwr_err:
        dpy FAILED RD/WR ERROR @ $1
        goto ram_end
dcd_err:
        dpy FAILED, DECODING ERROR @ $1
        goto ram_end
ram_ok:
        dpy RAM OK
ram_end:
        stop


program quickrom                                ! Quick ROM test
        dpy ADDRESS INCREMENT? /key             ! get address increment
        incr =  key shl 4
        incr = incr or 1
        dpy BEGINNING ADDRESS? /key
        lowaddr = key or 300000                 ! mask in 3XXXXX address for
        dpy ENDING ADDRESS? /key                ! 8051 quick ROM test
        highaddr = key or 300000
        write @ lowaddr = 0                     ! starting address
        write @ highaddr = incr                 ! ending address and increment
```

Figure A-1. Compiler Source Program (cont)

```
stat_lp:
        read @ ADR                              ! READ @ ENTER
        if DAT and F0 = C0 goto rom_done
        if DAT and F0 = A0 goto abort
        dpy BUSY                                ! display status
        execute delay                           ! delay speeds test by not
        goto stat_lp                            ! interupting the Pod during
rom_done:                                       ! the quick test
        read @ ADR
        if DAT and FF = C1 goto rom_err         ! if code C1, then an error
        if DAT and FF = C0 goto chksum          ! occured
abort:
        dpy TEST ABORTED, ILLEGAL ENTRY
        goto rom_end

rom_err:
        read @ F0300E
        dpy INACTIVE BITS DETECTED $e
        goto rom_end
chksum:
        read @ F0300D                           ! display Checksum
        reg1 = DAT
        read @ F0300C
        DAT = DAT shl 8                         ! Shift and combine 2 bytes
        DAT = DAT or reg1                       ! for 16 bit checksum
        dpy CHECKSUM = $e
        goto rom_end
rom_end:
        stop

program filltest                                ! Quick fill and verify
        dpy BEGINNING ADDRESS? /lowaddr         ! get address space info
        dpy ENDING ADDRESS? /highaddr
        dpy FILL MEMORY (Y-N)?key               ! get type of test
        if key = 1 goto fill
        dpy VERIFY MEMORY (Y-N)?key
        if key = 1 goto verify
        dpy FILL AND VERIFY (Y-N)?key
        if key = 1 goto fil_ver
fill:
        reg1 = 1                                ! code 1 is fill only
        dpy FILL DATA?  /key                    ! get data and write to
        write @ lowaddr = key                   ! first location
        goto dotest
verify:
        reg1 = 2                                ! code 2 is verify only
        goto dotest
fil_ver:
        reg1 = 3                                ! 3 is fill & verify
        dpy FILL DATA? /key                     ! get data and write to
        write @ lowaddr = key                   ! first location
        goto dotest
dotest:
        dpy Address Increment? /key             ! get "Z", the address
        incr = key shl 4                        ! increment
        incr = incr or reg1
        lowaddr = lowaddr or 400000             ! mask in 4XXXXX, the
        highaddr = highaddr or 400000           ! special address for
        write @ lowaddr = 0                     ! block tests.
        write @ highaddr = incr                 ! Start test
```

Figure A-1. Compiler Source Program (cont)

```
stat_lp:
        read @ ADR                              ! READ @ ENTER for info
        if DAT and F0 = F0 goto ver_err         ! code FX means an error
        if DAT and F0 = C0 goto ver_com         ! occured
        if DAT and F0 = A0 goto abort
        dpy BUSY
        execute delay                           ! delay speeds test by
        goto stat_lp                            ! not interupting the
ver_err:                                        ! Pod during quick test
        read @ F0400A                           ! get MSB of error Addr.
        reg 1 = DAT shl 8
        read @ F04009                           ! get middle byte
        reg 1 = reg 1 or DAT shl 8
        read @ F04008                           ! get LSB error Addr.
        reg 1 = reg 1 or DAT
        dpy FAILED VERIFY @ $1
        goto fil_end
ver_com:
        dpy TEST COMPLETE, NO ERRORS
        goto fil_end
abort:
        dpy TEST ABORTED, ILLEGAL ENTRY
        goto fil_end

fil_end:
        stop

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

Figure A-1. Compiler Source Program (cont)

EQUIVALENT TROUBLESHOOTER PROGRAM

```
 1  include "8031.pod"
 *  !
 *  !  8031 pod data file - Version 1.0
 *  !
 *
 *   FORCELN UNUSED = 1
 *   busadr = 10000                          ! 10000 for 8031 mode
 *   uutadr = 0000
 *
 2   DECLARATIONS
 3   assign rega to key
 4   assign regb to incr
 5   assign reg8 to highaddr
 6   assign reg9 to lowaddr
 7
 8  SETUP INFORMATION
 9   pod - 8031
10
11  program 0
12   dpy QUICK 8051 OPERATIONS              ! scroll through tests
13   execute 4
14  0:
15   dpy QUICK RAM TEST (Y-N)?A
16   if REGA = 0 goto 1
17   execute 1                              ! do the RAM test
18  1:
19   dpy QUICK ROM TEST (Y-N)?A
20   if REGA = 0 goto 2
21   execute 2                              ! do the ROM test
22  2:
23   dpy QUICK FILL OR VERIFY (Y-N)?A
24   if REGA = 0 goto 0
25   execute 3                              ! do the fill or verify
26
27  program 1                               ! Quick RAM test
28   dpy ADDRESS INCREMENT? /A
29   REGB = REGA shl 4
30   REGB = REGB or 1
31   dpy BEGINNING ADDRESS? /A
32   REG9 = REGA or 200000                  ! mask in special 2XXXXX address
33   dpy ENDING ADDRESS? /A                 ! for Quick RAM test
34   REG8 = REGA or 200000
35   write @ REG9 = 0                       ! Start ramtest at beginning
36   write @ REG8 = REGB                    ! write at ending address
37
38  0:
39   read @ REGF                            ! READ @ ENTER
40   if REGE and F0 = F0 goto 1             ! if an error occurs
41   if REGE and FF = C0 goto 2             ! if ramtest passes
42   if REGE and F0 = A0 goto 3             ! if test is aborted
43   dpy BUSY     ! else display status
44   execute 4
45   goto 0
46  3:
47   dpy TEST ABORTED, ILLEGAL ENTRY
48   goto 4
49  1:                                      ! get address of error
50   read @ F0200A                          ! byte 3 of address
51   reg1 = REGE shl 8                      ! move to MSB
52   read @ F02009                          ! byte 1 of address
53   reg1 = reg1 or REGE shl 8              ! move to middle byte
54   read @ F02008                          ! get byte 0 of address
55   reg1 = reg1 or REGE
56   read @  F020F0                         ! get error code
```

Figure A-1. Compiler Source Program (cont)

```
 57   if REGE = F0 goto 5
 58   if REGE = F1 goto 6
 59   goto 0
 60  5:
 61   dpy FAILED RD/WR ERROR @ $1
 62   goto 4
 63  6:
 64   dpy FAILED, DECODING ERROR @ $1
 65   goto 4
 66  2:
 67   dpy RAM OK
 68  4:
 69   stop
 70
 71
 72  program 2                                  ! Quick ROM test
 73   dpy ADDRESS INCREMENT? /A                 ! get address increment
 74   REG8 =  REGA shl 4
 75   REG8 = REG8 or 1
 76   dpy BEGINNING ADDRESS? /A
 77   REG9 = REGA or 300000                     ! mask in 3XXXXX address for
 78   dpy ENDING ADDRESS? /A                    ! 8051 quick ROM test
 79   REG8 = REGA or 300000
 80   write @ REG9 = 0                          ! starting address
 81   write @ REG8 = REG8                       ! ending address and increment
 82  0:
 83   read @ REGF                               ! READ @ ENTER
 84   if REGE and F0 = C0 goto 1
 85   if REGE and F0 = A0 goto 2
 86   dpy BUSY     ! display status
 87   execute 4                                 ! delay speeds test by not
 88   goto 0                                    ! interupting the Pod during
 89  1:                                         ! the quick test
 90   read @ REGF
 91   if REGE and FF = C1 goto 3                ! if code C1, then an error
 92   if REGE and FF = C0 goto 4                ! occured
 93  2:
 94   dpy TEST ABORTED, ILLEGAL ENTRY
 95   goto 5
 96
 97  3:
 98   read @ F0300E
 99   dpy INACTIVE BITS DETECTED $e
100   goto 5
101  4:
102   read @ F0300D                             ! display Checksum
103   reg1 = REGE
104   read @ F0300C
105          REGE = REGE shl 8                  ! Shift and combine 2 bytes
106   REGE = REGE or reg1                       ! for 16 bit checksum
107   dpy CHECKSUM = $e
108   goto 5
109  5:
110   stop
111
112  program 3                                  ! Quick fill and verify
113   dpy BEGINNING ADDRESS? /9                 ! get address space info
114   dpy ENDING ADDRESS? /8
115   dpy FILL MEMORY (Y-N)?A                   ! get type of test
116   if REGA = 1 goto 0
117   dpy VERIFY MEMORY (Y-N)?A
118   if REGA = 1 goto 1
119   dpy FILL AND VERIFY (Y-N)?A
120   if REGA = 1 goto 2
```

Figure A-1. Compiler Source Program (cont)

```
121  0:
122   reg1 = 1                                  ! code 1 is fill only
123   dpy FILL DATA?  /A                        ! get data and write to
124   write @ REG9 = REGA                       ! first location
125   goto 3
126  1:
127   reg1 = 2                                  ! code 2 is verify only
128   goto 3
129  2:
130   reg1 = 3                                  ! 3 is fill & verify
131   dpy FILL DATA? /A                         ! get data and write to
132   write @ REG9 = REGA                       ! first location
133   goto 3
134  3:
135   dpy Address Increment? /A                 ! get "Z", the address
136   REGB = REGA shl 4                         ! increment
137   REGB = REGB or reg1
138   REG9 = REG9 or 400000                     ! mask in 4XXXXX, the
139   REG8 = REG8 or 400000                     ! special address for
140   write @ REG9 = 0                          ! block tests.
141   write @ REG8 = REGB                       ! Start test
142  4:
143   read @ REGF                               ! READ @ ENTER for info
144   if REGE and F0 = F0 goto 5                ! code FX means an error
145   if REGE and F0 = C0 goto 6                ! occured
146   if REGE and F0 = A0 goto 7
147   dpy BUSY
148   execute 4                                 ! delay speeds test by
149   goto 4                                    ! not interrupting the
150  5:                                         ! Pod during quick test
151   read @ F0400A                             ! get MSB of error Addr.
152   reg 1 = REGE shl 8
153   read @ F04009                             ! get middle byte
154   reg 1 = reg 1 or REGE shl 8
155   read @ F04008                             ! get LSB error Addr.
156   reg 1 = reg 1 or REGE
157   dpy FAILED VERIFY @ $1
158   goto 8
159  6:
160   dpy TEST COMPLETE, NO ERRORS
161   goto 8
162  7:
163   dpy TEST ABORTED, ILLEGAL ENTRY
164   goto 8
165
166  8:
167   stop
168
169  program 4
170   DECLARATIONS
171   assign reg1 to counter
172   REG1 = 30
173  0:
174   REG1 = REG1 dec
175   if REG1 > 0 goto 0
176
177
```

Figure A-1. Compiler Source Program (cont)

Appendix B
Testing UUTs With Soldered-In Microprocessors

INTRODUCTION B-1.

The Pod is normally used with UUTs that have their microprocessors installed in sockets. The microprocessor is removed from the UUT's socket and replaced by the Pod's UUT plug. With a few special techniques and a simple cable adapter, UUTs with soldered-in 8044- and 8051-family microprocessors may also be tested.

SPECIAL TEST PROCEDURES B-2.

This Appendix contains a description of the special procedures for using the Pod with soldered-in microprocessors and instructions for making the cable adapter that is required to connect the Pod's plug to such UUTs.

Installation B-3.

Section 2-2, Connecting the Pod to the UUT, in this manual describes how to attach the Pod plug to the UUT's microprocessor socket. When using a clip-on adapter, step 5 of that procedure should be amended as follows:

5. Connect the Pod's plug to the UUT using the clip-on adapter:

   a. Insert the pins of the Pod plug into the sockets on the adapter. Make sure that pin 1 of the Pod plug is aligned with pin 1 of the adapter.

   b. Open the adapter clip and attach it to the solder-in microprocessor. Make sure that pin 1 of the adapter is aligned with pin 1 of the microprocessor.

Figure B-1 shows how to connect the Pod plug to the soldered-in microprocessor using a clip-on adapter.

Note that the adapter must be removed before conducting another Pod self test, or an adapter must be devised for the self-test socket.

Configuration B-4.

Section 2-3, Configuration, describes how to set the mode switches on the top of the Pod to configure it to your specific UUT. When testing a UUT that has a solder-in microprocessor, set:

Switch 6 — The UUT Connection switch must be to the Clip-On position. This will provide the high logic level to the microprocessor's RST pin as described below.

Switches 7,8

For UUTs with crystal oscillators, Switch 7 should be set to XTAL and Switch 8 should be set to XTAL1.

For UUTs with external clocks, the switches should be set the same as described in paragraph 2-3.

Using the Pod With Soldered-in Microprocessors B-5.

Once the Pod is attached to the UUT with the special clip-on adapter, most test procedures and methods remain the same. The only important exception to normal Troubleshooter operation is that the RUN UUT command may not work. Some microprocessors will not do a RUN UUT properly when using a clip-on adapter. It doesn't hurt to try RUN UUT, but you should not expect it to work. There are no other special tactics required to compensate for the presence of the soldered-in microprocessor.

NOTE

The UUT's Reset function will not work during RUN UUT operations. The Pod will be holding the UUT's RST line high at all times; therefore the RST line is not available for normal Reset functions.

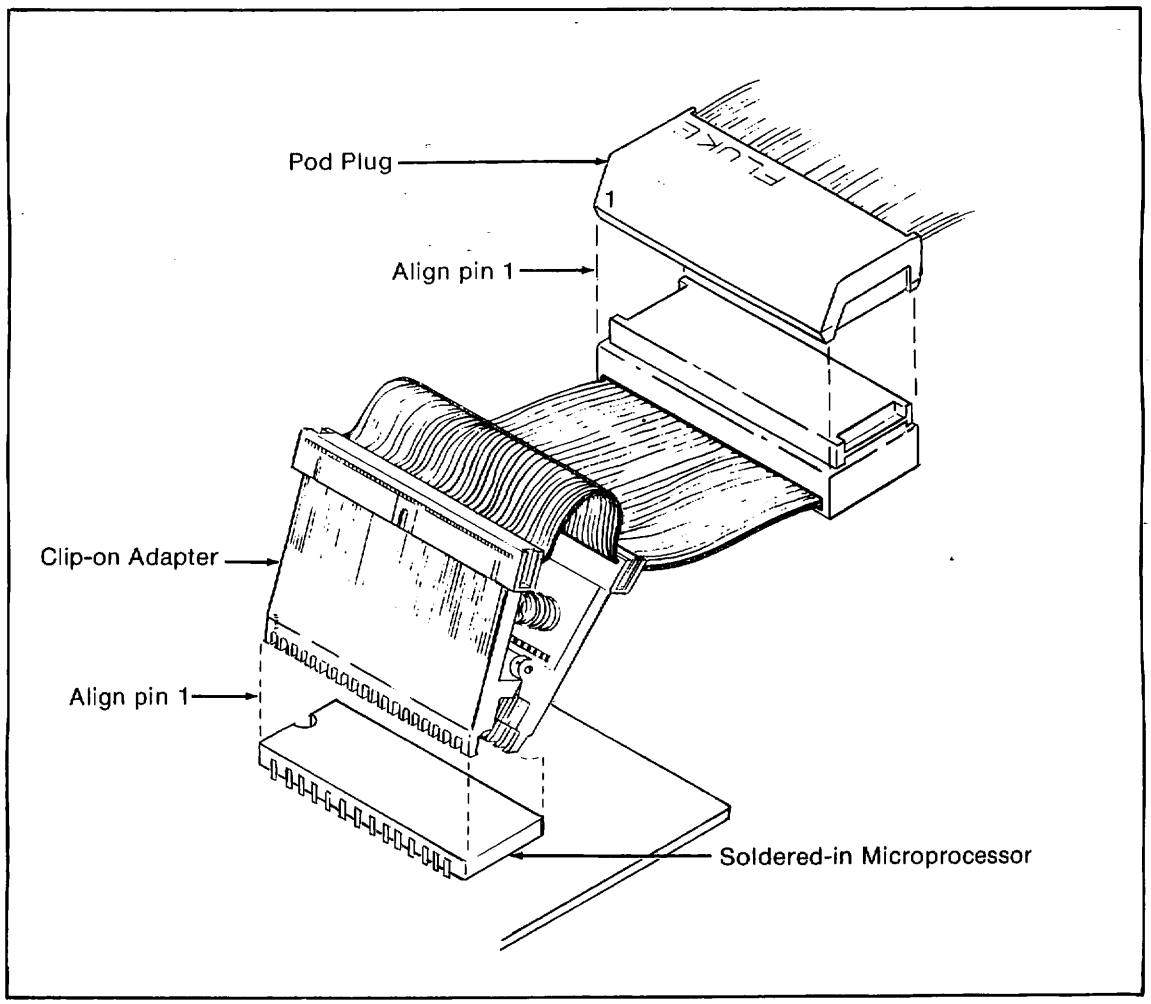


Figure B-1. Connecting the Pod to a Soldered-In Microprocessor Using a Clip-On Adapter

Theory of Operation With Soldered-in Microprocessors B-6.

When the RST line of the 8044- and 8051-family microprocessors is held to a high logic level, all of the other pins (except for the clock pins XTAL1 and XTAL2), go to a high-impedance level. This feature allows the Pod to test soldered-in UUTs by pulling RST high to disable all of the lines.

When mode switch 6 is set to the Clip-On position, the Pod will try to pull the UUT's RST line to a high level. The Pod's pull-up resistor is 1K ohm. If the UUT has a pulldown resistor on the Reset line, it should not be less than 4K ohm. If a logic gate on the UUT is driving the RST line (pin 9), the gate must either be disabled or forced to drive RST high.

Troubleshooting a Clip-On Adapter B-7.

If the Pod passes the self test, but does not appear to work properly when connected to a UUT using the clip-on adapter, begin troubleshooting the Pod by carefully checking the adapter:

1. Check the adapter for continuity from the clip-on connector contacts back to the Pod plug socket contacts. An adapter may be tested for continuity by inserting a 40-pin header into the self-test socket, clipping on to it, and initiating a self test. (Insert a 40-pin header into the self-test socket, then clip the adapter on to it. Conduct a Pod self test to locate continuity errors.)

2. Check for shorts between adjacent signal lines throughout the adapter.

3. Use an oscilloscope to check that the clock signal to the Pod is present. If the clock source is a crystal oscillator, then the clock signal should be a sine wave. If the source is TTL logic, then the signal should be a TTL square wave.

4. Use an oscilloscope to check for excessive noise on the lines. If noise is present, shield the adapter with a wrap of metal foil, grounded to pin 20.

If these procedures do not yield a solution, contact Fluke Customer Service for advice.

CLIP-ON ADAPTERS B-8.

In order to test UUTs with solder-in microprocessors you will need an adapter to connect the standard Pod plug to the soldered-in microprocessor. You may purchase an adapter from Fluke, or you may construct an adapter yourself. You may purchase the optional Fluke 9000A-201 8051 Pod Clip Adapter or you may construct an adapter yourself.

The adapter should be constructed to provide a clip-on connection to the microprocessor, and socket contacts for the Pod plug. You should keep the cable length of the adapter as short as possible to avoid a deteriorating effect on the signals. Twisted wire pairs are not required as long as the adapter is less than six inches long.

Contact a Fluke Sales Office for information about a factory-made adapter.

Appendix C
Serial Port Testing

INTRODUCTION C-1.

This Appendix describes how to test the serial port for the benefit of testing other system components which depend on exchanging serial information with the microprocessor. Testing the serial port consists of verifying that the serial lines are drivable, and of exercising the port to verify that correct data is transmitted and received.

TESTING THE SERIAL PORT LINES FOR DRIVABILITY C-2.

The Serial Input Port (RXD - P3.0) and Serial Output Port (TXD - P3.1) lines should be tested for drivability faults using the Special Function Bit Addresses (5 00B0=P3.0 and 5 00B1=P3.1). For example, if *WRITE @ 5 00B1 = 00* resulted in a data error with bit 0, that indicates a problem was encountered driving the transmit line low. Repeat with *WRITE @ 5 00B1 = 01* (or any non-zero value) to test bit 0 for drivability to high levels.

EXERCISING THE SERIAL PORT C-3.

Introduction C-4.

The serial communications port of the microprocessor is tested by writing control information to the appropriate control registers and to the serial port buffer. To set up the port for testing, you must determine what serial port mode your UUT uses and what the baud rate is, then calculate the proper control words. See the microprocessor manufacturer's data for a description of the microprocessor's serial port.

The addresses which can be used to configure the serial port and to transmit and receive data are summarized in Table C-1.

Setting Up The Serial Port For Testing C-5.

Prior to transmitting or receiving data, the serial port must be configured to send and receive serial data. Use the following procedure:

1. Set the mode with a *WRITE @ 4 0098 = XX* to load the Serial Control Register (SCON).

2. Set the mode for timer 1 or 2, if it is to be used as a baud-rate generator, with *WRITE @ 4 0089 = XX*.

3. Load the baud-rate value into 4 008D. The baud rate is determined from the microprocessor manufacturer's literature, using formulas involving the oscillator frequency and the timer reload value (TH1).

Table C-1. Serial Port Test Addresses

PROCEDURE	DESCRIPTION
Setup:	
WRITE @ 40098	Set Serial Port Mode in SCON.
WRITE @ 40089	Set Timer 1 mode in TCON.
WRITE @ 4008D	Load TH1 with auto-reload value for baud rate.
WRITE @ 5008E	Set TR1 bit to start Timer 1 operation.
To Transmit:	
WRITE @ 50099	Reset TI bit prior to transmitting first character.
WRITE @ 40099=xx	Write to SBUF to transmit.
Repeat:	
WRITE @ 40099 = xx	Write to SBUF to transmit a serial character.
READ @ 50099	Check TI bit to see that transmission is complete (data will be 1 when transmission is complete).
WRITE @ 50099	Reset TI bit prior to transmitting next character.
To Receive:	
WRITE @ 5009C	Set REN bit to enable port to receive serial data.
WRITE @ 50098	Reset RI bit prior to receiving first character.
READ @ 40099	Read SBUF.
Repeat:	
READ @ 50098	Check RI bit to verify that character is received (data will be 1 when reception is complete).
READ @ 40099	Read SBUF for received character.
WRITE @ 50098	Reset RI bit prior to receiving next character.

4. *WRITE @ 5 008E = 1* to start the timer running by setting the run-control bit in TCON.

The following example procedure describes how to set up a serial port test on a UUT that has a 7.3738 MHz clock, operates the serial port in mode 1 (10 bit frame with 1 start bit and 1 stop bit), and is connected to a terminal which receives at 9600 Baud:

1. To transmit in mode 1, *WRITE @ 4 0098 = 40* to load the Serial Control Register (SCON) with hexadecimal 40.
2. Set the mode for timer 1 to 8-bit auto-reload mode (TMOD) with *WRITE@4 0089 = 20.*

3. Load the baud-rate value into 4 008D. In this example the value is FE, so load the proper baud rate with *WRITE @ 4 008D = FE.*

4. *WRITE @ 5 008E = 1* to start the timer.

Transmitting Via The Serial Port C-6.

After the serial port is configured, any value written to the Serial Buffer (SBUF) will be automatically transmitted out.

Repeat the following procedure to transmit via the serial port:

1. *WRITE @ 50099 = 0* to reset the TI bit prior to transmitting first character and before each subsequent character.

2. Write to SBUF (*WRITE @ 40099 = XX*) to transmit a serial character.

3. Check the TI bit to see that transmission is complete by a *READ @ 50099* (data will be 1 when transmission is complete).

NOTE

At low baud rates, you should only write one byte at a time to SBUF. Otherwise, the Troubleshooter will try to overwrite SBUF before the microprocessor transmits all of the bits.

Receiving Via The Serial Port C-7.

Use the following procedure to receive data via the serial port:

1. *WRITE @ 5009C = 1* to set the REN bit to enable the port to receive serial data.

Repeat the following steps for each character to be received:

2. *WRITE @ 50098 = 0 to* reset the RI bit prior to receiving the first character and prior to receiving subsequent characters.

3. *READ @ 50098* to check the RI bit to verify that a character was received (data will be 1 when reception is complete).

4. *READ @ 40099* to Read SBUF for received character.

Appendix D
Using the Pod with a Remote 9020A

The 8051 Pod may be used with a 9020A Micro-System Troubleshooter in its remote IEEE-488- or RS-232-controlled mode. The only Pod-specific information that you need to know to use specific Pods with a remotely operated 9020A is the commands for controlling the Enableable Status Lines. There are no Enableable Status lines in the 8051 Pod, so Table 6-9, Enable Line Setup Commands, in the 9020A Operator Manual should be ammended as shown in Table D-1 below.

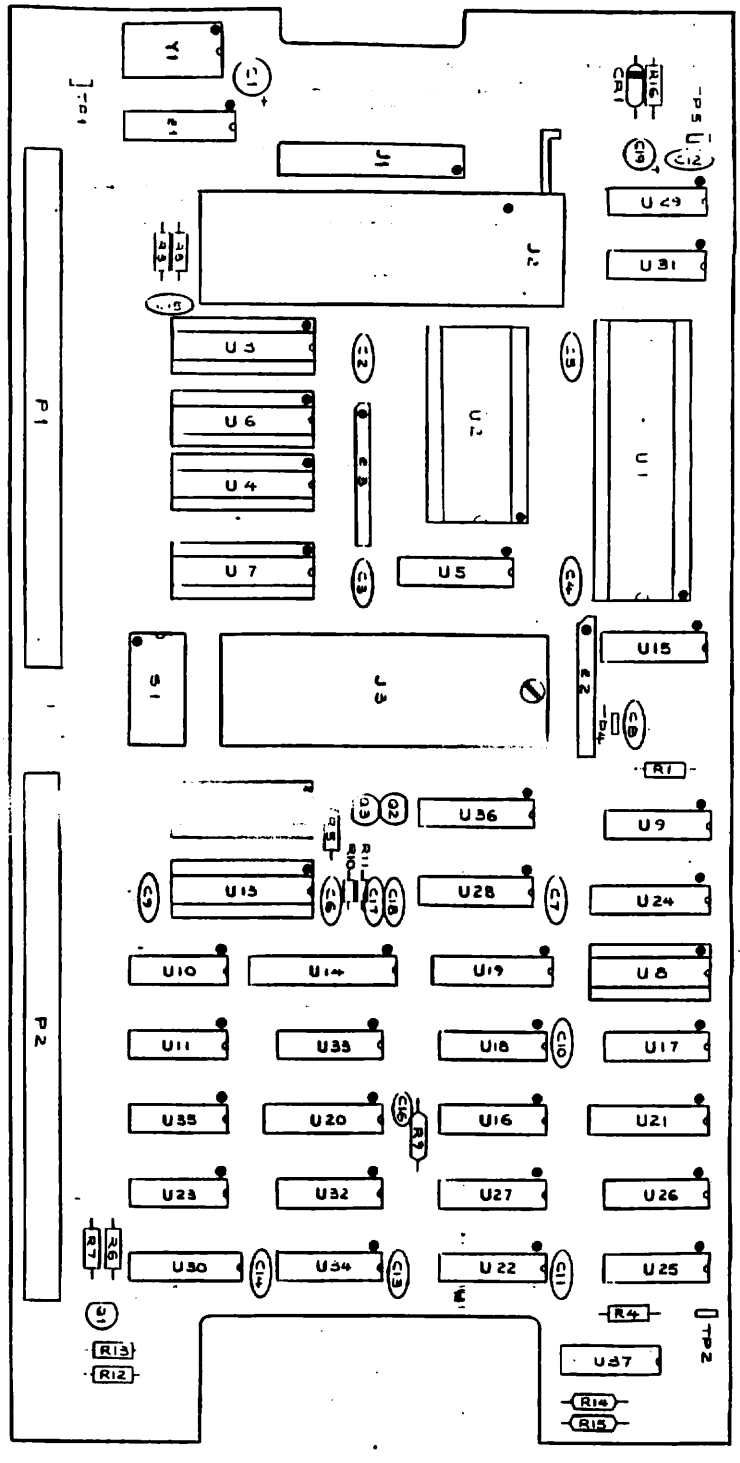
Table D-1. Enable Line Setup Commands

INTERFACE POD	SETUP COMMANDS							
	S0,8	S0,9	S0,10	S0,11	S0,12	S0,13	S0,14	S0,15
8051	Unused	*	*	*	*	*	*	*

** If this command is sent to the 9020A, it will cause a command error (status response 95).*

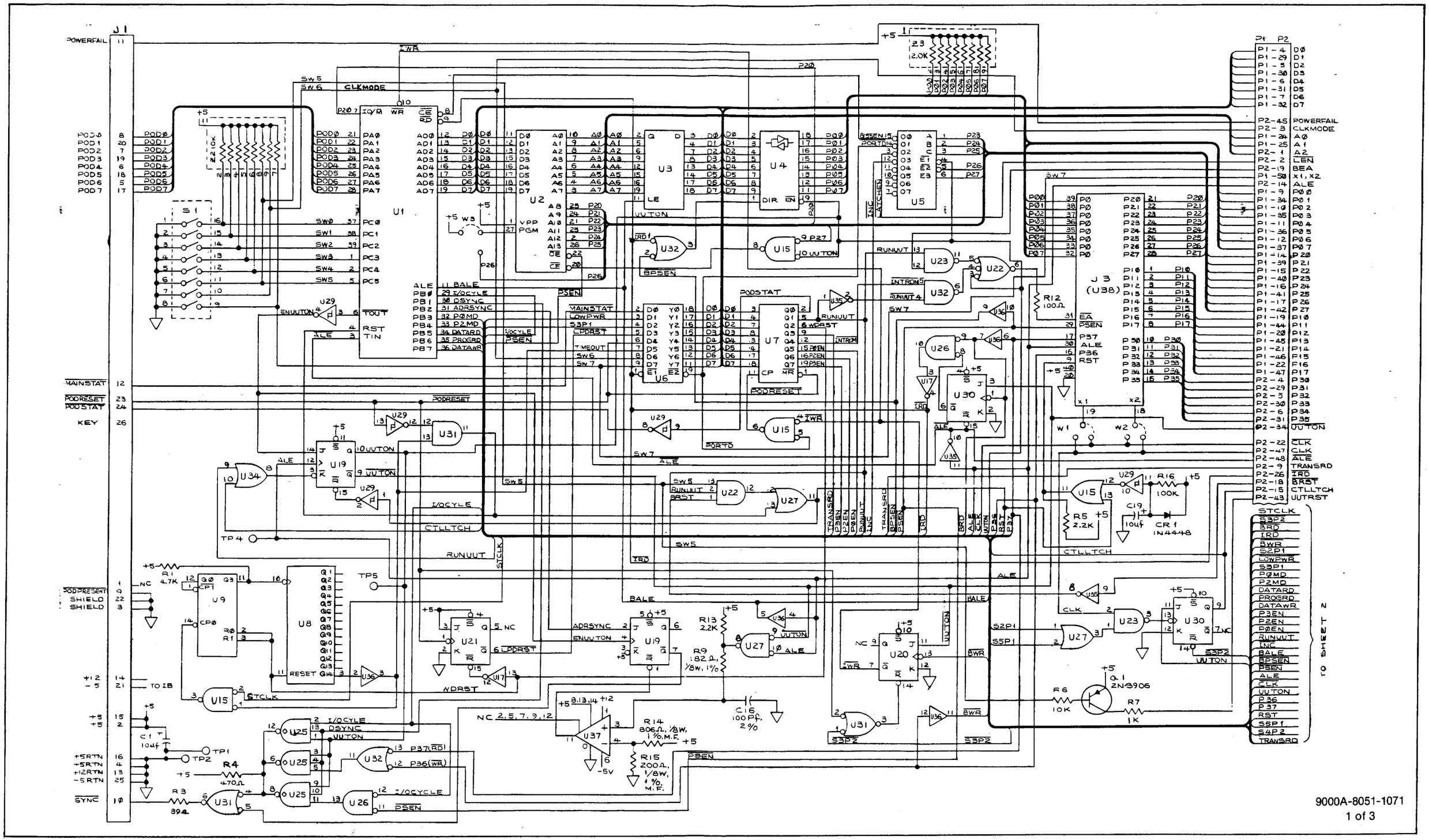
Section 8
Schematic Diagrams

TABLE OF CONTENTS



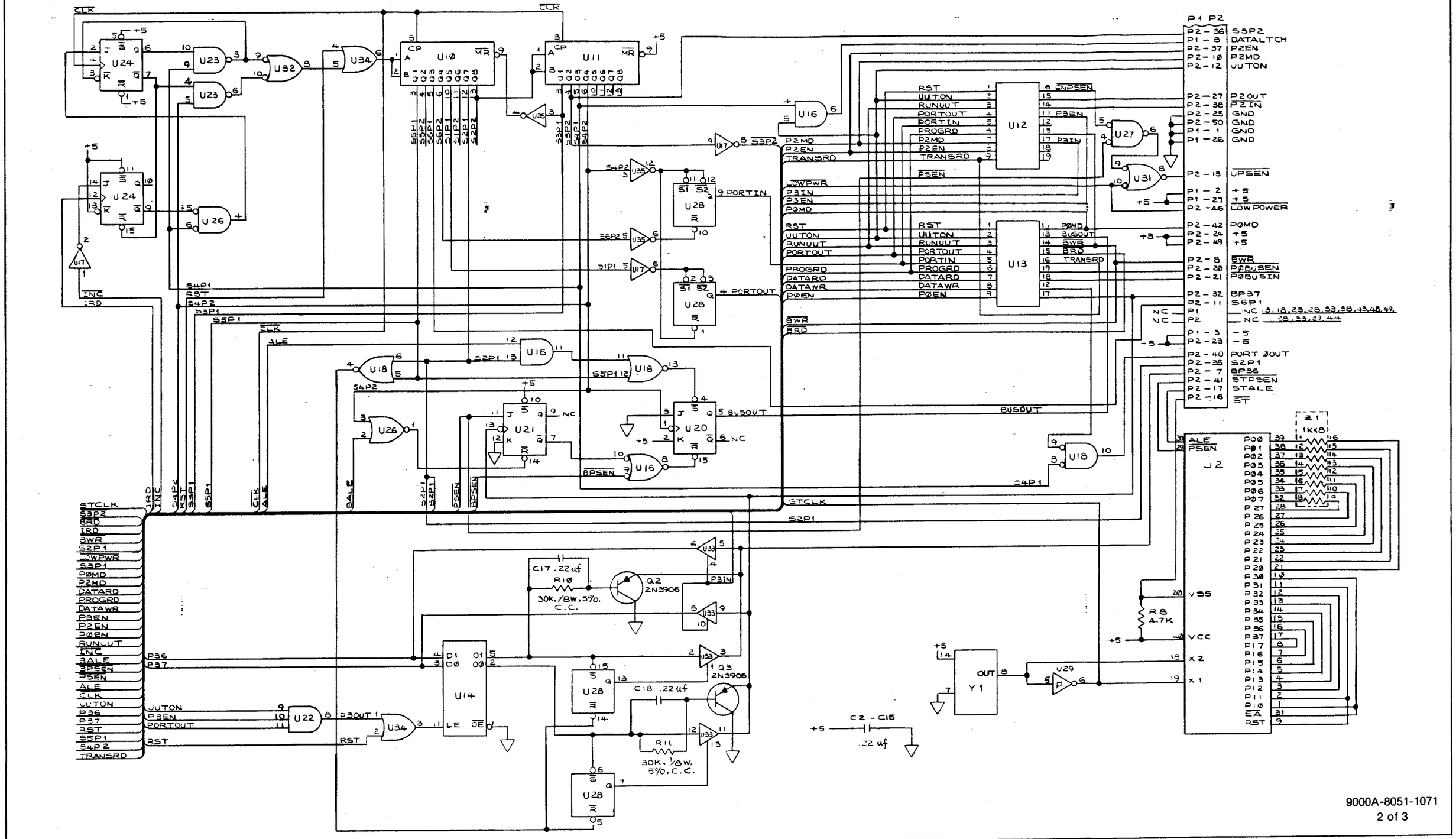
9000A-8051-1671

Figure 8-1. A38 Processor PCB Assembly



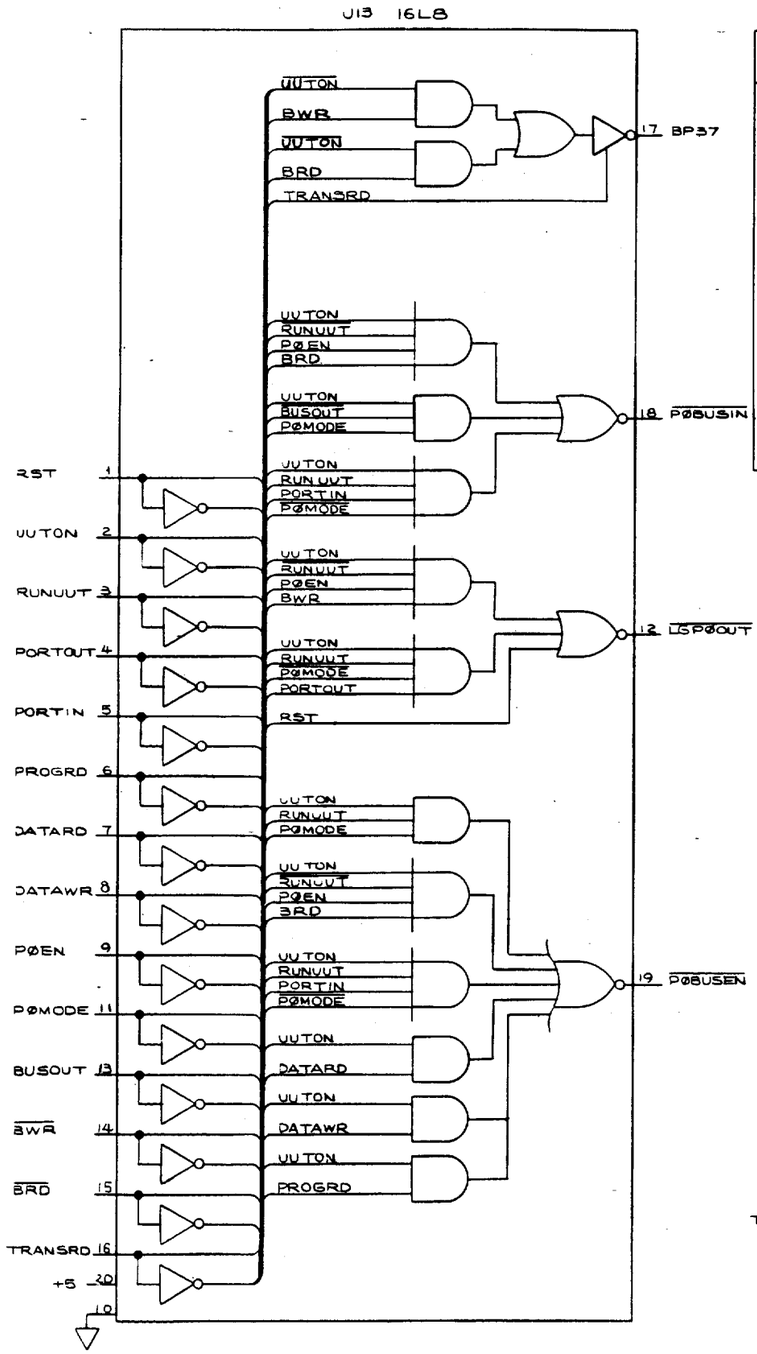
9000A-8051-1071
1 of 3

Figure 8-1. A38 Processor PCB Assembly (cont)



9000A-8051-1071
2 of 3

Figure 8-1. A38 Processor PCB Assembly (cont)



MEMORY MAP

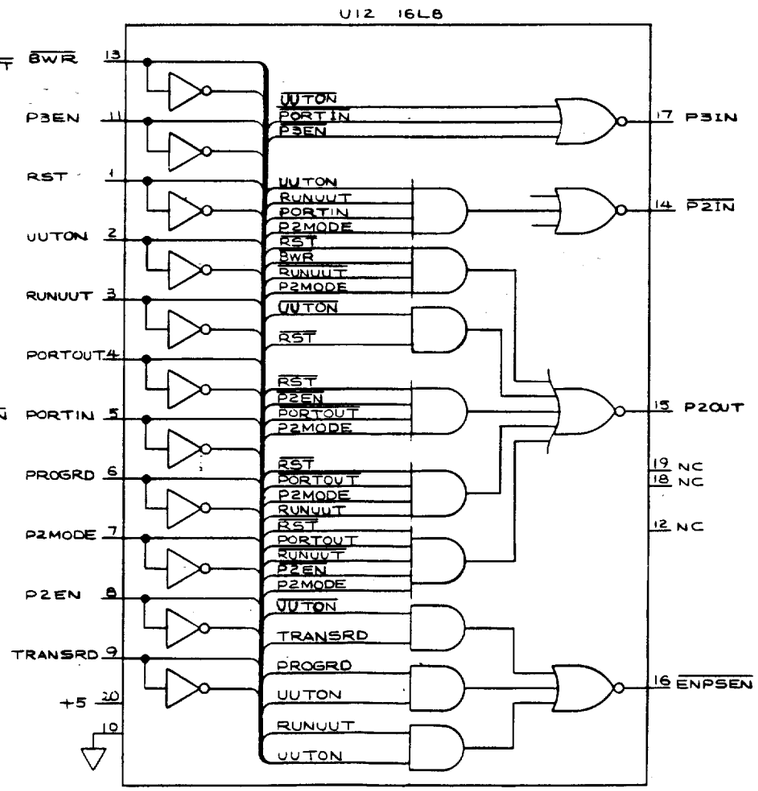
CONTROL	LOWER ADDRESS	UPPER ADDRESS	DEVICE	DESCRIPTION
PSEN	0000	1FFF	U2(2764A)EPROM	PROGRAM MEMORY
RD / WR	8000	80FF	U1(8155)	256 BYTE RAM
	8100			CONTROL/STATUS REGISTER
	8101			PORT A
	8102			PORT B
	8103			PORT C
	8104			TIMER(LOW BYTE)
	8105			TIMER(HIGH 6 BITS + MODE)
	8800		U6 + U7	PORT D.
RD	9000		* U2(74SC374)	PORT0 DATA LATCH
	9001		* U5	PORT0 ADDRESS LATCH
	9002		* U7	PORT 1 LATCH
	9003		* U1	PORT 2 LATCH
	9004		* U4	PORT 3 LATCH
	9005		* U3	STATUS LATCH
	9006		* U6	CONTROL LATCH
	9800		U 24	STATE COUNTER INCREMENT

* DEVICES ON INTERFACE BOARD *9000A-8051-1072

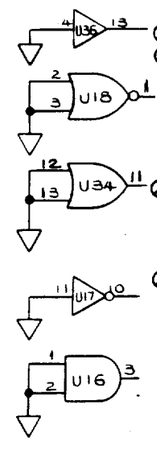
REFERENCE DESIGNATIONS	
LAST USED	NOT USED
U38	
R16	R2
C19	
W3	
Q3	
S1	
J3	
P2	
Z3	
Y1	
TP5	TP3
CR1	

NOTES: UNLESS OTHERWISE SPECIFIED.

1. ALL RESISTORS ARE IN OHMS, ¼W, 5%. ALL CAPACITORS ARE IN MICROFARADS.



SPARE GATES

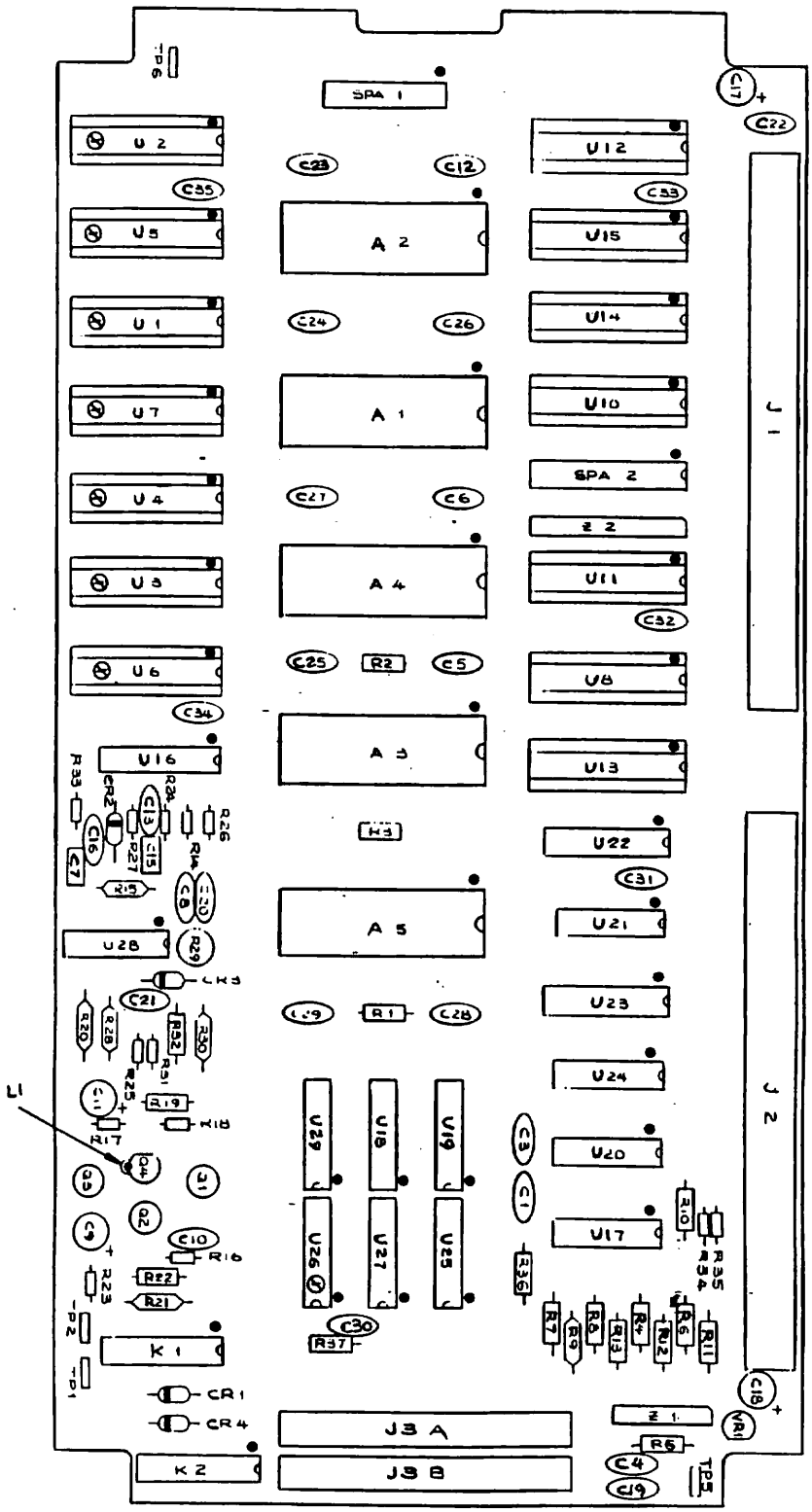


DES	DEVICE	+5	GND	PINS	QTY
Z1	DIP. RES.NET. 1K	—	—	16	1
Z2	SIP. RES.NET. 10K	1	—	10	1
Z3	SIP. RES.NET. 2K	1	—	10	1
U1	8155-2	40	20	40	1
U2	27128	28	14	28	1
U3	74LS373	20	1,10	20	1
U4	74ALS245	20	10	20	1
U5	74LS138	6	8	16	1
U6	74LS541	20	10	20	1
U7	74LS273	20	10	20	1
U8	40208	16	8	16	1
U9	74LS93	5	10	14	1
U10,11	74LS164	14	7	14	2
U19,24	74LS109	16	8	16	2
U12,13	PAL16L8	20	10	20	2
U14	74F373	20	10	20	1
U15,27,34	74LS32	14	7	14	3
U16	74LS08	14	7	14	1
U17,35	74LS04	14	7	14	2
U18,26	74LS02	14	7	14	2
U21,20,30	74LS112	16	8	16	3
U22	7411	14	7	14	1
U23,32	74LS00	4	7	14	2
U25	7412	14	7	14	1
U28	74LS279	16	8	16	1
U31	74F08	14	7	14	1
U33	74LS126	14	7	14	1
U36	74LS367	16	1,8,15	16	1
U29	74LS14	14	7	14	1
U37	LM361	8,13,14	10	14	1
U38	8031AH	40	20	40	1

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

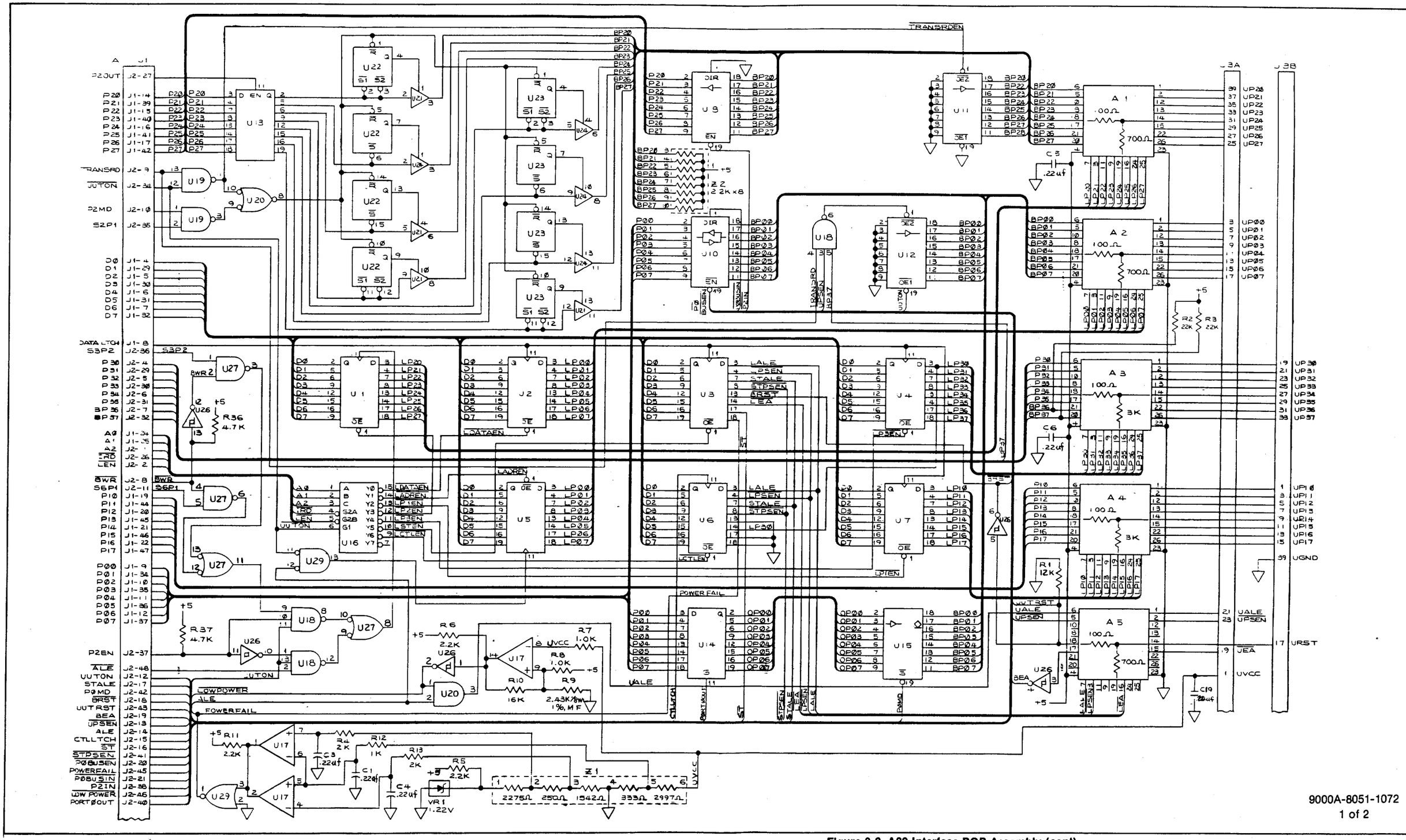
9000A-8051-1071
3 of 3

Figure 8-1. A38 Processor PCB Assembly (cont)



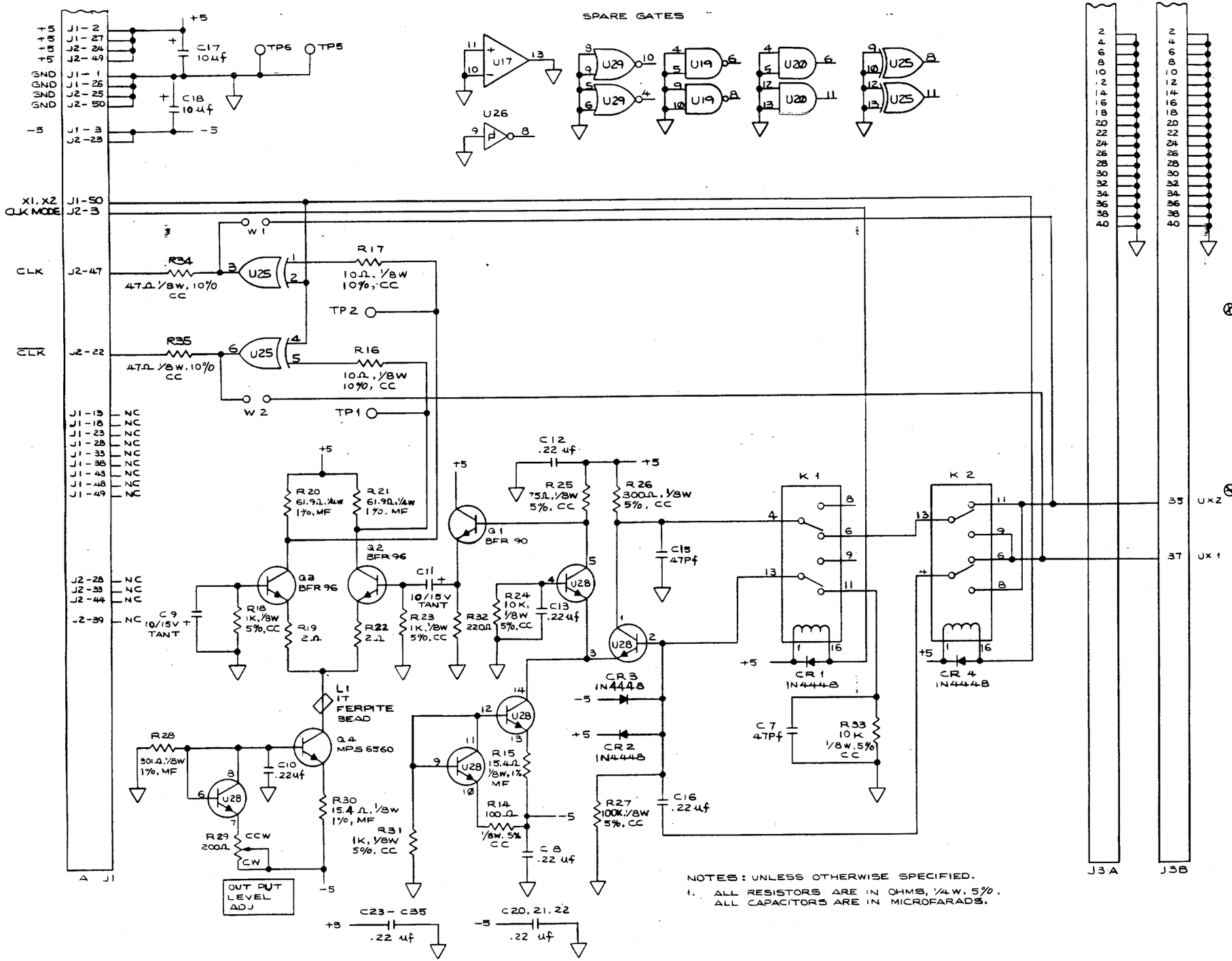
9000A-8051-1672

Figure 8-2. A39 Interface PCB Assembly



9000A-8051-1072
1 of 2

Figure 8-2. A39 Interface PCB Assembly (cont)



DES	DEVICE	+5	GND	PINS	QTY
A1 A2, A5	9000A -4H02T	4	23	26	3
A3, A4	9000A -4H01T	4	23	26	2
⊗ U1 – U7	74SC374	20	10	20	7
U8, 10	74ALS245	20	10	20	2
U11, 12	74LS 541	20	10	20	2
U15	74LS641	1.20	10	20	1
U16	74LS138	16	8	16	1
U17	LM 339	3	12,13	14	1
U18	74LS10	14	1.7	14	1
U19, 27	74LS00	14	7	14	2
U20	74LS08	14	7	14	1
U21, 24	74LS126	14	7	14	2
U25	74F86	14	7	14	1
U26	74HC14	14	7	14	1
U28	CA3046	—	—	14	1
U29	74LS02	14	7	14	1
U22, 23	74LS279	16	8	16	2
U13, 14	74F373	20	1,10	20	2
Z1	54-4087T		4		1
Z2	RES. NETWORK, 22KΩ	1		10	1

REFERENCE DESIGNATION	
LAST USED	NOT USED
A5	
C35	C14, C2
CR4	
J1	
L1	
Q4	
R37	
U29	J9
VR1	
Z2	
TP6	TP3, 4
K2	

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

9000A-8051-1072
2 of 2

Figure 8-2. A39 Interface PCB Assembly (cont)

INDEX

The number following each index entry indicates the page number.