NOTE

This manual documents the Model 9000A-6809 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-6809
Interface Pod

Instruction Manual

P/N 649400
OCTOBER 1982

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

Rev. 6/81

Table of Contents

(continued on page ii)

TABLE OF CONTENTS, *continued*

List of Tables

List of Illustrations

Section 1
Introduction

NOTE

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

1-1. PURPOSE OF INTERFACE POD

The purpose of the 9000A-6809 Interface Pod (hereafter referred to as the pod) is to interface any 9000 Series Micro System Troubleshooter (hereafter referred to as the troubleshooter) with equipment employing a 6809 or 6809E microprocessor.

The troubleshooter is designed to service printed circuit boards, instruments, and systems employing microprocessors. The architecture of the troubleshooter is general in nature, and accommodates devices with up to 32 address lines and 32 data lines. The pod adapts the general purpose architecture of the troubleshooter to the specific requirements of the 6809 and 6809E microprocessors. The pod adapts such microprocessor-specific functions as pin layout, status/control functions, interrupt handling, timing, and memory and I/O addressing.

1-2. DESCRIPTION OF POD

Figure 1-1 shows the communication between the pod, the troubleshooter, and the unit-under-test (hereafter 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 UUT microprocessor socket. The UUT microprocessor is removed from the UUT and replaced by the pod ribbon cable plug which gives the troubleshooter access to all system components which normally communicate with the microprocessor.

The external features of the pod are shown in Figure 1-2. In order for the pod to operate, the operator must install a 6809 (for 6809-based UUTs) or a 6809E (for 6809E-based UUTs) microprocessor in the pod. The UUT microprocessor may be installed in the pod if it is functioning properly. The microprocessor is installed in a screw-actuated zero-insertion-force socket that is exposed by sliding back the door on the top of the pod. A decal beside the socket provides instructions about how to set the DIP switches to select either the 6809 or 6809E operating mode.

The pod consists of a pair of printed circuit board assemblies mounted within a break-resistant case. The pod contains the supporting hardware and control software that is required to do the following:

1. Perform handshaking with the troubleshooter.

2. Receive and execute commands from the troubleshooter.

3. Report UUT status to the troubleshooter.

4. Allow the pod microprocessor to execute program code stored in the UUT.

The troubleshooter supplies operating power (+5V) for the pod. The UUT provides either the external crystal (for the 6809) or the external clock (either the 6809 or the 6809E) required for pod operation. Using the UUT crystal or 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. These circuits allow detection of bus shorts, stuck-high or stuck-low conditions, and any bus drive conflict (two or more drivers attempting to drive the same bus line).

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

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

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

The over-voltage protection circuits guard against voltages of +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 (+5V). If UUT power rises above or drops below a voltage level which could cause the UUT to operate improperly, the pod notifies the troubleshooter of the UUT power supply problem.

A self test socket provided 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 must be connected to the self test socket during self test operation. When the pod is not in use, the ribbon cable plug should also be inserted into this socket to provide protection for the plug.

1-3. SPECIFICATIONS

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

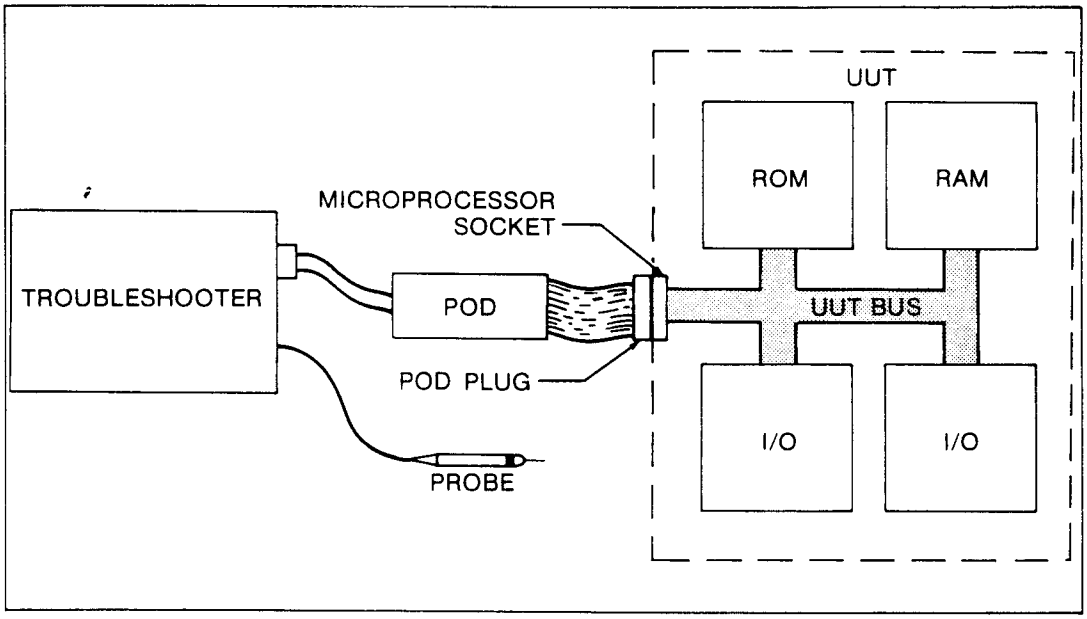


Figure 1-1. Communication Between the Troubleshooter, the Pod, and the UUT

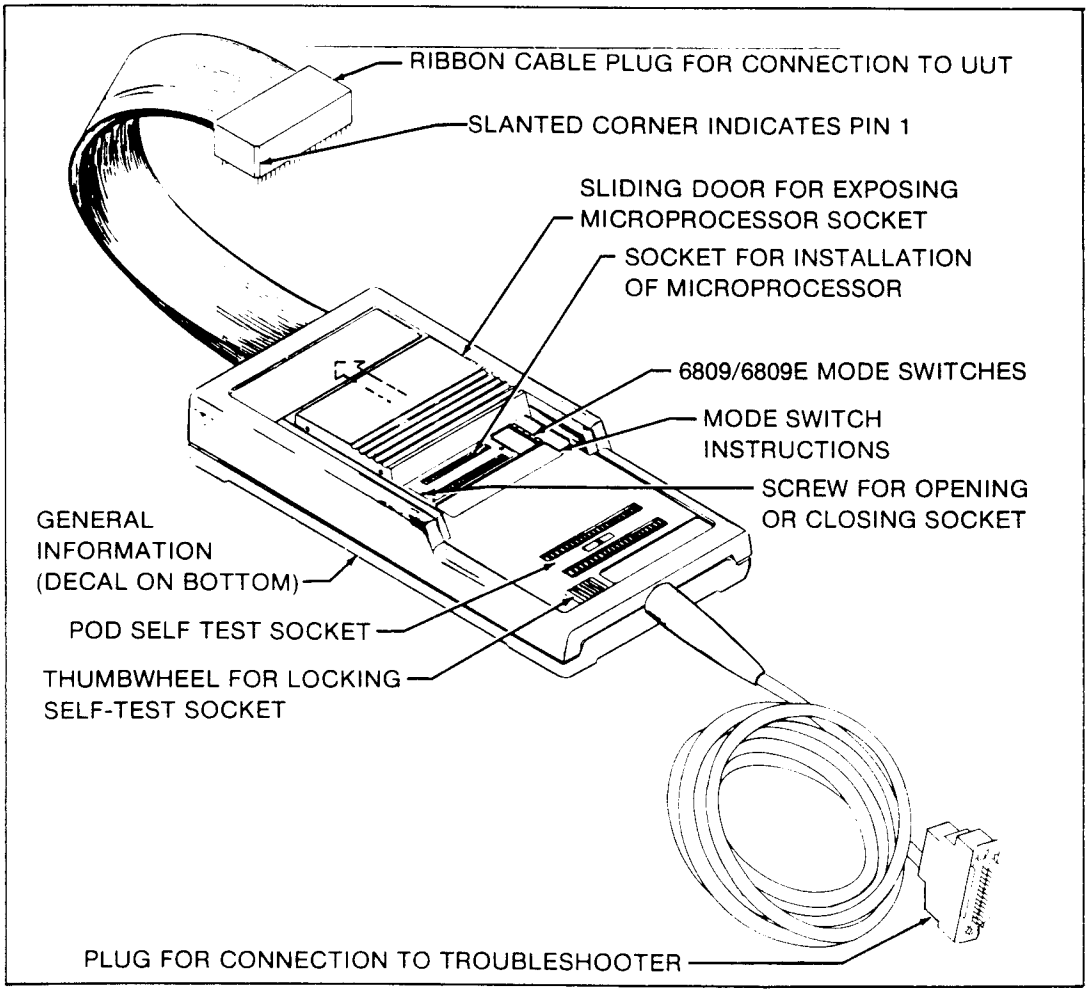


Figure 1-2. External Features of the 6809/6809E Interface Pod

Table 1-1. 6809 Pod Specifications

ELECTRICAL PERFORMANCE	
Power Dissipation	3.0 watts max.
Maximum External Voltage	-7 to +12 volts may be applied between ground and any ribbon cable plug pin continuously.
MICROPROCESSOR SIGNALS*	
Input Low Voltage (All Inputs)	0.8V max.
Input High Voltage	
ALL INPUTS EXCEPT $\overline{\text{RESET}}$	2.0V max.
$\overline{\text{RESET}}$	4.0V min.
Output Low Voltage	
ALL OUTPUTS EXCEPT D0-D7	0.4V max., Iol = 4 mA
D0-D7	0.5V max., Iol = 15 mA
Output High Voltage	
ALL OUTPUTS EXCEPT D0-D7	2.4V min., Ioh = 400 uA
D0-D7	2.4V min., 1 oh = 3 mA
Tristate Output Leakage Current	
Vin HIGH OR LOW, A0-A15, R/$\overline{\text{W}}$	20 uA max.
Vin HIGH, D0-D7	40 uA max.
Vin LOW, D0-D7	200 uA max.
TIMING CHARACTERISTICS	
Max. External Clock Frequency	8.0 MHz
Added Delays to 6809 Signals	
LOW-TO-HIGH TRANSITIONS	20 ns
HIGH-TO-LOW TRANSITIONS	24 ns
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	1.4 kg (3.1 lbs)
Environment	
STORAGE	-40°C to +70°C, RH <95%
OPERATING	0°C to +25°C, RH <95% +25°C to +40°C, RH <75% +40°C to +50°C, RH <45%
Protection Class 3	Relates solely to insulation or grounding properties defined in IEC 348.

*Signals are specified as they appear at ribbon cable plug pins.
**Pod outputs tristate and/or drives lines to low logic level.

Section 2
Installation and Self Test

2-1. INTRODUCTION
The following paragraphs describe how to install the microprocessor in the pod, connect the pod to the troubleshooter, perform the pod self test, and connect the pod to the UUT.

2-2. INSTALLING THE MICROPROCESSOR IN THE POD
The microprocessor that is installed in the pod must be the same type as the UUT microprocessor. In general, it is preferable to install the UUT microprocessor in the pod if it is functioning properly. (The basic functioning of the microprocessor that is installed in the pod is tested during the pod self test.)

Both the 6809 and the 6809E microprocessors are available in three different models with three different operating speeds (1.0 MHz, 1.5 MHz, and 2.0 MHz). The microprocessor that is installed in the pod should have an operating speed that is equal to or greater than the operating speed of the UUT microprocessor. A 68B09 microprocessor (2.0 MHz operating speed) is included with each pod that is purchased.

To install the microprocessor in the pod, perform the following steps:

1. Remove power from the UUT and the troubleshooter.

2. If you are going to install the UUT microprocessor in the pod, disassemble the UUT and remove the UUT microprocessor from its socket. If you are not going to install the UUT microprocessor in the pod, select a microprocessor that matches the UUT microprocessor type (6809 or 6809E) and has adequate operating speed.

3. Install the microprocessor in the pod microprocessor socket that is exposed under the sliding door as shown in Figure 2-1. The pod socket is opened or closed by using a screwdriver to turn the screw at the end of the socket. Turn the screw counterclockwise to open the socket, and clockwise to close the socket.

4. Place the 6809/6809E mode switches indicated in Figure 2-1 to the correct position as indicated on the decal beside the switches. (Notice that switch 4 is in the same position for both modes; switch 4 has a special use during pod troubleshooting as explained in the subsection in Section 6 titled Preparation for Troubleshooting a Defective Pod.)

2-3. CONNECTING THE POD TO THE TROUBLESHOOTER
Connect the pod to the troubleshooter by plugging the round shielded cable into the troubleshooter socket shown in Figure 2-2. Secure the connector using the sliding collar.

2-4. PERFORMING THE POD SELF TEST

To perform the pod self test, perform the following steps:

1. Be sure that the proper microprocessor is installed in the pod microprocessor socket and that the mode switches are set in accordance with the instructions on the pod decal.

2. Open the pins of the pod self test socket by operating the adjacent thumbwheel. Insert the ribbon cable plug into the socket and close the socket using the thumbwheel.

3. Be sure that the pod is properly connected to the troubleshooter and that the troubleshooter power is on. Press the BUS TEST key to initiate the pod self test.

If the troubleshooter displays the message *POD SELF TEST 6809 OK* then the pod is operating properly. If the troubleshooter displays any message other than *POD SELF TEST 6809 OK*, check the microprocessor position in the pod socket, check the cable connections, and check the position of the mode switches. Try the self test again. If the resulting message is not *POD SELF TEST 6809 OK*, install another microprocessor in the pod and repeat the self test. If the resulting message is not *POD SELF TEST 6809 OK*, the pod may not be operating properly.

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

2-5. CONNECTING THE POD TO THE UUT

WARNING

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

Connect the pod to the UUT as follows:

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

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

3. If you have not already done so, disassemble the UUT to gain access to the UUT microprocessor socket and remove the UUT microprocessor.

4. Turn the pod self test socket thumbwheel to release the pod ribbon cable plug, and remove the plug from the self test socket.

5. Insert the pod ribbon cable plug into the UUT microprocessor socket. Make sure the slanted corner of the pod plug is aligned with pin 1 of the UUT microprocessor socket.

6. Reassemble the UUT using extender boards if necessary.

CAUTION

To prevent damage to the pod, ensure that troubleshooter power is on before turning UUT power on in order to activate protection circuits within the pod.

7. Apply power to the UUT.

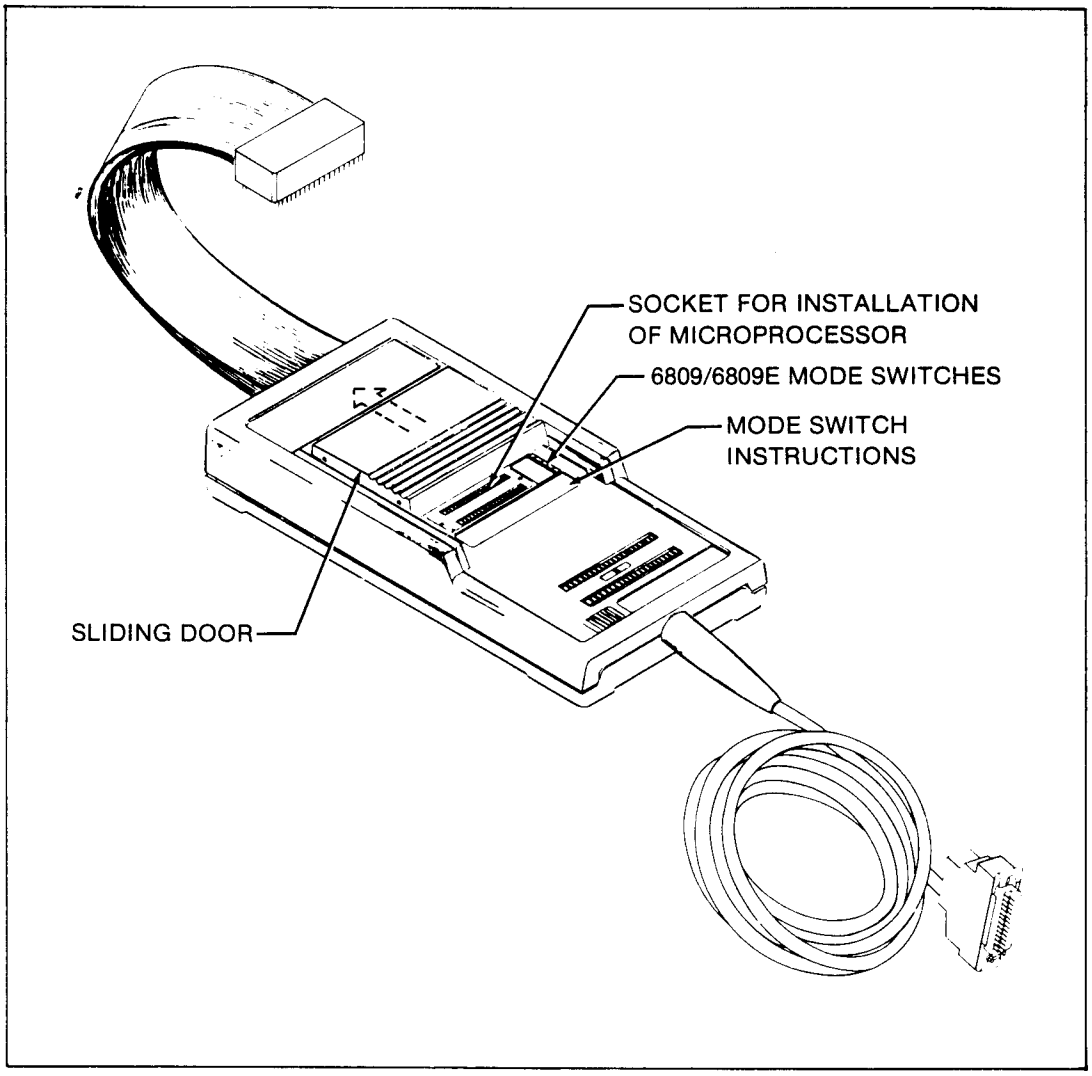


Figure 2-1. Location of Microprocessor Socket and 6809/6809E Mode Switches

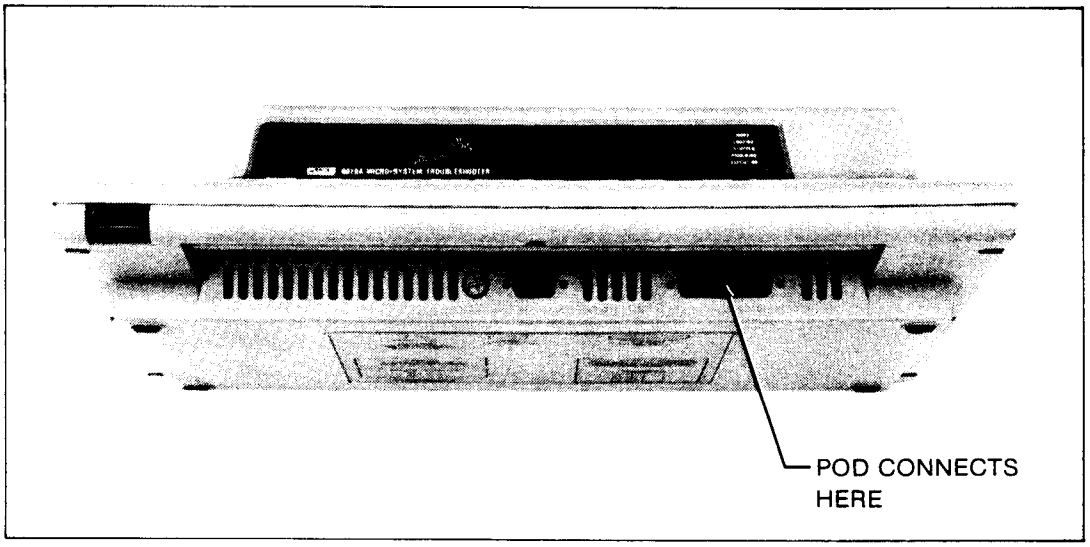


Figure 2-2. Connection of Interface Pod to Troubleshooter

Section 3
Microprocessor Data

3-1. INTRODUCTION

This section contains microprocessor data which may be useful during operation of the troubleshooter. This information includes descriptions of the 6809 and 6809E signals and pin assignments.

3-2. MICROPROCESSOR SIGNALS

Table 3-1 lists all of the 6809 microprocessor signals and provides a brief description of each signal. Table 3-2 provides similar information for the 6809E microprocessor.

Figure 3-1 shows the pin assignments for both the 6809 and the 6809E microprocessors. All of the signals are the same for the two microprocessors except for 6 of the 40 signals. Two of the signals, Q and E, have the same name for both microprocessors. Notice, however, that for the 6809, the Q and E signals are clock signal outputs, and for the 6809E, the Q and E signals are clock signal inputs.

Table 3-1. 6809 Signal Descriptions

SIGNAL NAME	DESCRIPTION
A0-A15	The 16 address lines are designated A0 through A15. When the uP does not require the address bus for a data transfer, it outputs all address lines high (and also makes R/$\overline{W}$ high and BS low). When the BA line is high, all address lines are in a high-impedance state.
D0-D7	The eight data lines are designated D0 through D7.
$\overline{HALT}$	The $\overline{Halt}$ line is an input which, when made low, causes the uP to stop for an indefinite period upon completion of the current instruction. In the Halt mode: • The BA and BS lines are high. • The uP does not respond to $\overline{FIRQ}$ or $\overline{IRQ}$. • The uP responds to a low at the DMA input. • Inputs to $\overline{NMI}$ or $\overline{RESET}$ are latched for later response. • The Q and E outputs operate normally.

Section 3
Microprocessor Data

3-1. INTRODUCTION

This section contains microprocessor data which may be useful during operation of the troubleshooter. This information includes descriptions of the 6809 and 6809E signals and pin assignments.

3-2. MICROPROCESSOR SIGNALS

Table 3-1 lists all of the 6809 microprocessor signals and provides a brief description of each signal. Table 3-2 provides similar information for the 6809E microprocessor.

Figure 3-1 shows the pin assignments for both the 6809 and the 6809E microprocessors. All of the signals are the same for the two microprocessors except for 6 of the 40 signals. Two of the signals, Q and E, have the same name for both microprocessors. Notice, however, that for the 6809, the Q and E signals are clock signal outputs, and for the 6809E, the Q and E signals are clock signal inputs.

Table 3-1. 6809 Signal Descriptions

SIGNAL NAME	DESCRIPTION
A0-A15	The 16 address lines are designated A0 through A15. When the uP does not require the address bus for a data transfer, it outputs all address lines high (and also makes R/$\overline{W}$ high and BS low). When the BA line is high, all address lines are in a high-impedance state.
D0-D7	The eight data lines are designated D0 through D7.
$\overline{HALT}$	The $\overline{Halt}$ line is an input which, when made low, causes the uP to stop for an indefinite period upon completion of the current instruction. In the Halt mode: • The BA and BS lines are high. • The uP does not respond to $\overline{FIRQ}$ or $\overline{IRQ}$. • The uP responds to a low at the DMA input. • Inputs to $\overline{NMI}$ or $\overline{RESET}$ are latched for later response. • The Q and E outputs operate normally.

Table 3-1. 6809 Signal Descriptions (cont)

<table>
<tr><th>SIGNAL NAME</th><th>DESCRIPTION</th></tr>
<tr><td>BA, BS</td><td>The BA (bus available) line is made high by the uP when halted, by an active DMA or Halt signal, or a Wait instruction, to indicate that the buses are in a high-impedance condition.

The BS (bus status) line is an output which can be decoded with the BA line, according to the following table, to indicate the state of the uP.

<table>
<tr><th>BA</th><th>BS</th><th>uP STATE</th></tr>
<tr><td>0</td><td>0</td><td>Normal (running)</td></tr>
<tr><td>0</td><td>1</td><td>Interrupt or reset acknowledge</td></tr>
<tr><td>1</td><td>0</td><td>Sync acknowledge</td></tr>
<tr><td>1</td><td>1</td><td>Halt or bus grant acknowledge</td></tr>
</table>
</td></tr>
<tr><td>R/$\overline{W}$</td><td>The Read/$\overline{\text{Write}}$ line is made high by the uP when in a read mode, and low when in a write mode to indicate the direction of data transfer on the data bus. When the BA line is made high, the state of the Read/$\overline{\text{Write}}$ line is high-impedance.</td></tr>
<tr><td>$\overline{\text{RESET}}$</td><td>The $\overline{\text{Reset}}$ line is an input which requires a low level for longer than one bus cycle. Reset vectors are fetched from location FFFE and FFFF.</td></tr>
<tr><td>$\overline{\text{FIRQ}}$</td><td>The fast interrupt request line is an input which, when made low, initiates a fast interrupt sequence, provided its mask bit in the condition code is clear. This sequence has priority over the standard interrupt ($\overline{\text{IRQ}}$) and is fast because it stacks only the contents of the condition code register and the program counter.</td></tr>
<tr><td>$\overline{\text{IRQ}}$</td><td>The interrupt request line is an input which, when made low, requests an interrupt to the uP, provided its mask bit in the condition code is clear.</td></tr>
<tr><td>$\overline{\text{NMI}}$</td><td>A low-going signal on the non-maskable interrupt line initiates a non-maskable interrupt routine within the uP upon completion of the current instruction. $\overline{\text{NMI}}$ has priority over $\overline{\text{FIRQ}}$ and $\overline{\text{IRQ}}$.</td></tr>
<tr><td>EXTAL</td><td>The external clock/crystal input connects to an external clock source when an external clock is employed, or to one side of a crystal when the internal clock circuit of the uP is employed.</td></tr>
</table>

Table 3-1. 6809 Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
XTAL	The crystal input connects to one side of a crystal when the internal clock circuit of the uP is employed. When using an external clock, the crystal input is tied low.
E	The enable output supplies a single-phase clock signal from the internal clock to the remainder of the system.
Q	The quadrature output supplies a single-phase clock signal which leads the enable (E) output by 90 degrees.
MR	The memory ready input provides for controlling the stretching of the enable (E) and quadrature (Q) outputs to accommodate slow memory or peripheral devices. This input is usually tied high when unused.
$\overline{DMA}$	The direct memory access input provides a method of suspending program execution and acquiring the uP bus for such purposes as DMA and dynamic memory refresh. A low signal activates this input, and the uP acknowledges the input by making BA and BS high.

Table 3-2. 6809E Signal Descriptions

SIGNAL NAME	DESCRIPTION
A0-A15	The 16 address lines are designated A0 through A15. When the uP does not require the address bus for a data transfer, it outputs all address lines high (and also makes R/$\overline{W}$ high and BS low). All address lines are placed in a high-impedance state when the TSC input is asserted.
D0-D7	The eight data lines are designated D0 through D7. All data lines are placed in a high-impedance state when the TSC input is asserted.
$\overline{HALT}$	The $\overline{Halt}$ line is an input which, when made low, causes the uP to stop for an indefinite period upon completion of the current instruction. In the $\overline{Halt}$ mode: • The BA and BS lines are high. • The uP does not respond to $\overline{FIRQ}$ or $\overline{IRQ}$. • Inputs to $\overline{NMI}$ or $\overline{RESET}$ are latched for later response. • The Q and E outputs operate normally.

Table 3-2. 6809E Signal Descriptions (cont)

<table>
<tr><th>SIGNAL NAME</th><th>DESCRIPTION</th></tr>
<tr><td>BA, BS</td><td>The BA (bus available) line is made high by the uP when halted, or waiting for external synchronization on an interrupt line, to indicate that the buses are in a high-impedance condition. The BA line is not asserted when the TSC input is active.

The BS (bus status) line is an output which can be decoded with the BA line, according to the following table, to indicate the state of the uP.

<table>
<tr><th>BA</th><th>BS</th><th>uP STATE</th></tr>
<tr><td>0</td><td>0</td><td>Normal (running)</td></tr>
<tr><td>0</td><td>1</td><td>Interrupt or reset acknowledge</td></tr>
<tr><td>1</td><td>0</td><td>Sync acknowledge</td></tr>
<tr><td>1</td><td>1</td><td>Halt or bus grant acknowledge</td></tr>
</table>
</td></tr>
<tr><td>R/$\overline{W}$</td><td>The Read/$\overline{Write}$ line is made high by the uP when in a read mode, and low when in a write mode to indicate the direction of data transfer on the data bus. The state of the Read/$\overline{Write}$ line is high-impedance when the TSC input is asserted.</td></tr>
<tr><td>$\overline{RESET}$</td><td>The $\overline{Reset}$ line is an input which requires a low level for longer than one bus cycle. Reset vectors are fetched from location FFFE and FFFF.</td></tr>
<tr><td>$\overline{FIRQ}$</td><td>The fast interrupt request line is an input which, when made low, initiates a fast interrupt sequence, provided its mask bit in the condition code is clear. This sequence has priority over the standard interrupt ($\overline{IRQ}$) and is fast because it stacks only the contents of the condition code register and the program counter.</td></tr>
<tr><td>$\overline{IRQ}$</td><td>The interrupt request line is an input which, when made low, requests an interrupt to the uP, provided its mask bit in the condition code is clear.</td></tr>
<tr><td>$\overline{NMI}$</td><td>A low-going signal on the non-maskable interrupt line initiates a non-maskable interrupt routine within the uP upon completion of the current instruction. $\overline{NMI}$ has priority over $\overline{FIRQ}$ and $\overline{IRQ}$.</td></tr>
<tr><td>E, Q</td><td>The E and Q inputs accept the clock signal required by the uP. The clock signal at Q leads the signal at E by less than 90 degrees.</td></tr>
</table>

Table 3-2. 6809E Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
BUSY	The busy output is made high by the uP during the read and modify cycles of a read-modify-write instruction and during the access of the first byte of a double-byte operation. The busy output is also high during the first byte of any indirect or other vector fetch. In a multiprocessor system, the busy signal indicates the need to defer rearbritration of the next bus cycle to insure the integrity of the above operations.
AVMA	The advance valid memory address output is made high to indicate that the uP will use the bus in the next bus cycle. In a multiprocessor system, the AVMA signal allows efficient shared-bus operation.
LIC	The last instruction cycle output is made high during the last cycle of every instruction.
TSC	The three-state control input commands the data, address, and R/W buses to the high-impedance state to allow shared-bus operations.

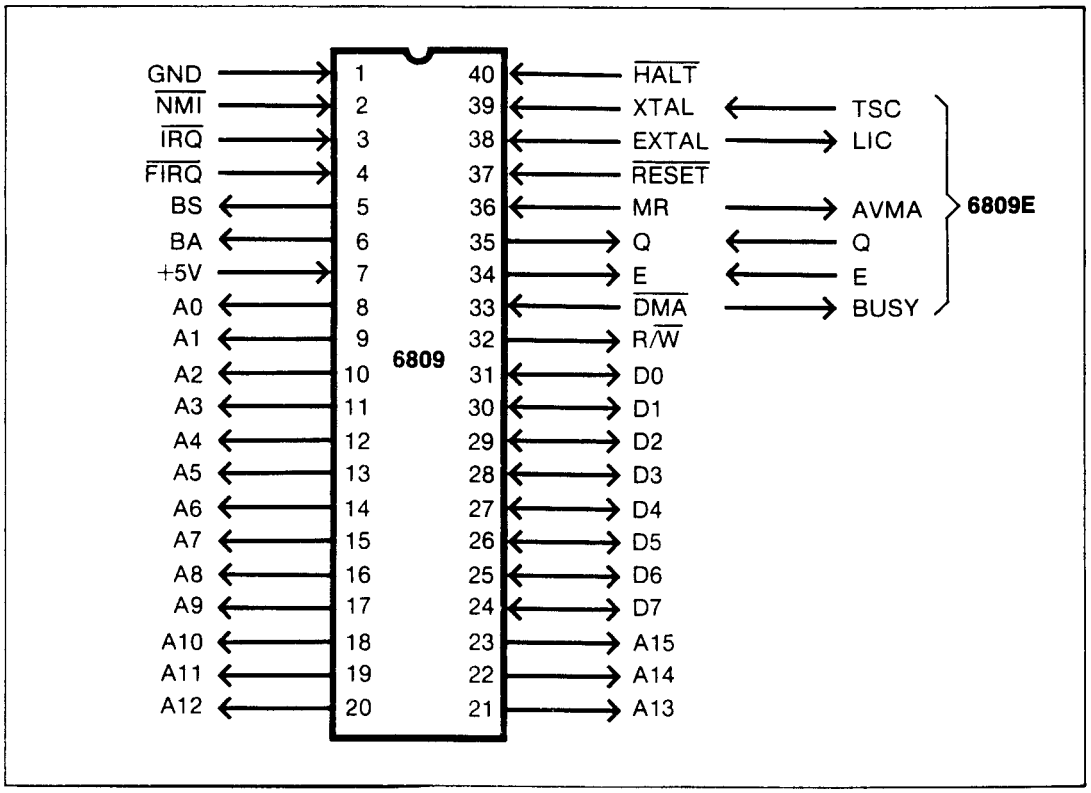


Figure 3-1. 6809/6809E Pin Assignments

Section 4
Operating Information

4-1. INTRODUCTION

This section contains information which pertains to operating the troubleshooter with 6809-based systems. This additional information complements the information in the troubleshooter operator and programming manuals, and covers such items as the following:

- Address space assignment
- Special address functions
- Definition and bit assignment of status lines
- Definition of forcing and interrupt lines
- Bit assignment of control lines
- Definition and characteristics of user-writable control lines
- Marginal UUT problems

4-2. HELPFUL HINTS ABOUT GETTING STARTED

After the pod is connected to the troubleshooter and installed in the UUT, it is very possible to encounter error messages that prevent you from performing any operations involving the pod and the UUT. These initial problems are very often easily solved by checking switches or selecting the proper troubleshooter Setup message. The following paragraphs present some possible error messages and some suggestions about how to eliminate the errors.

If the message *POD TIMEOUT - ATTEMPTING RESET* appears, try the following:

- Make sure the mode switches are in the correct position, and that the correct microprocessor is installed in the socket under the sliding door. Remember that to change from one mode (6809 or 6809E) to another you must reset the switches and change the microprocessor. The pod preparation and mode selection is described in Section 2.

- When the *POD TIMEOUT* message appears at power on, it is often caused by faulty status lines (input lines to the pod microprocessor) called pod enable lines. You can disable the enable lines with the troubleshooter Setup function by setting the Setup messages *SET - ENABLE xxxx? to NO.*

- If you attempt an operation and the message is now *ACTIVE FORCE LINE - LOOP?,* the problem is probably caused by the same faulty status line. You can disable the reporting of this error and continue operation by selecting the troubleshooter Setup function again and setting the Setup message *SET - TRAP ACTIVE FORCE LINE?* to *NO.* For more information about enable lines refer to a later section titled User-Enableable Status Lines. For more information about forcing lines, refer to a later section titled Forcing Lines.

- If the *POD TIMEOUT* message remains even after you have disabled the enable lines, the problem may be caused by a faulty UUT clock. If this problem exists, the pod will not operate until it is corrected. Information about troubleshooting the pod is provided in Section 6.

If the message *ACTIVE FORCE LINE @ addr-LOOP?* appears in response to a periodic reset, try the following:

- Many UUTs have a watchdog timer that generates periodic resets unless inhibited by UUT software. There are two ways to deal with this situation. You can select the troubleshooter Setup message *SET - TRAP ACTIVE FORCE LINE? NO.* This will not prevent the periodic resets, but it will disable the reporting of the resets and allow you to proceed with testing. However, some UUTs may not be reliably tested if the periodic reset continues. In this case, the watchdog timer should be disabled on the UUT.

- A looping Read Status operation is useful for discovering a watchdog timer. The pod Reset status line, bit 4, will be toggling high and low. Note that some UUTs have a very narrow reset pulse, and may be only occasionally detected by the troubleshooter.

If the message *UUT POWER FAIL - ATTEMPTING RESET* appears, the pod may have timed out because the UUT power supply has been detected to be below 4.5V. Check the UUT power supply.

If you encounter problems when using the Run UUT mode, try the following:

- Check the speed of the microprocessor. In most cases, the pod will operate in Run UUT if the microprocessor that is installed in the pod is the same speed as the UUT microprocessor. Remember that the pod adds a delay of approximately 20 nS to most of the pod signals. If the UUT timing is marginal, the extra delay may be responsible for the problems. Install a microprocessor with a higher operating speed and see if this solves the problems.

- Check the position of the switches to make sure they are set as indicated on the pod decal.

4-3. ADDRESS SPACE ASSIGNMENT

The address space assignments are listed in Table 4-1. The default execution addresses for the Learn operation, Run UUT, and Bus Test are also listed in Table 4-1.

In addition to the regular address spaces listed in Table 4-1, the pod has some special addresses that are used to access information in the pod or to cause the pod to perform some special functions. Two of these special functions, the Quick Looping read and the Quick Looping write, are described in the next section titled Quick Looping Functions. Other special addresses are used in troubleshooting and are described with the troubleshooting procedures in Section 6.

Table 4-1. Address Space Assignment and Quick Looping Functions

FUNCTION	ADDRESS OR OPERATION	DESCRIPTION
Memory Addresses I/O Addresses	0000 to FFFF	The pod is capable of addressing up to 65,536 (FFFF hex) memory locations. The I/O addresses are mapped into this memory space also. The pod does not distinguish between memory access operations and I/O access operations.
Learn Operation Default Address Range	0000 to FFFF	When the Learn operation is specified without a specified address range, the pod performs the Learn operation over the entire microprocessor range, 0000 to FFFF.
Run UUT Default Address	FFFFFFFE	When operation in the Run UUT mode is selected, but no execution address is specified, the pod begins execution at the vectored address that is read from locations FFFE (most significant word) and FFFF (least significant word). This address vectoring occurs in the same way as a standard 6809 microprocessor reset.
Bus Test Default Address	0000	When the Bus Test is performed, the troubleshooter tests the data lines at a particular address. This address is not entered when the Bus Test is selected, but is stored at power on as the troubleshooter Setup message *SET-BUS TEST @ 0000-CHANGE?* You may change the address by changing the Setup message.
Quick Looping Read or Write	READ @ 1XXXX WRITE @ 1XXXX	A read or write at address 1XXXX will cause the pod to perform a quick looping read or write at address XXXX. UUT system errors are reported only during the first execution of the read or write operation, and not during the succeeding execution loops.

4-4. QUICK LOOPING FUNCTIONS

There are two troubleshooting functions that are unique to the 6809/6809E pod: the Quick Looping read and the Quick Looping write. Unlike the ordinary Looping function, the software routines that control these Quick Looping functions reside in the pod and not in the troubleshooter. The operator selects these functions by writing to the special addresses listed in Table 4-1.

As the name implies, the advantage of the Quick Looping functions is that they are executed more quickly than the corresponding ordinary functions. It should be noted, however, that the diagnostics performed by the pod during the execution of the Quick Looping functions are less rigorous than the diagnostics performed during the execution of the ordinary functions. When a read or write operation is specified at the address 1XXXX, the pod first performs a read or write operation at address XXXX in the normal manner, reporting to the troubleshooter any UUT system errors detected (such as *ACTIVE FORCE LINE*, or *CTL ERR*, etc.). Then the pod enters the Quick Looping mode where the read or write operation is performed much faster than the ordinary

Looping function specified by pressing the LOOP key on the troubleshooter keyboard. During the Quick Loop, the pod does not check for any UUT system errors that may occur. Any pod timeout errors that occur are reported as usual. The Quick Loop continues until the operator selects another operation.

For example, if the operator specifies the operation *READ @ 1F000,* the pod will perform a looping read operation at the address F000. If the operator specifies the operation *WRITE @ 1B007 = 2F,* the pod will perform a looping write operation at address B007, writing the data 2F.

The Quick Looping function can be particularly useful when viewing signals on an oscilloscope that is synchronized to the TRIGGER OUTPUT pulse (available on the troubleshooter rear panel). If a signal trace on the oscilloscope screen is dim due to a low repetition rate, the Quick Looping function can increase the repetition rate to make the signal trace much more visible.

If both error reporting and the Quick Looping functions are desired, you may apply the ordinary troubleshooting Looping function to the Quick Looping read or write, such as *READ@12000 LOOP.* The troubleshooter will command read operations at address 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.

The Quick Looping function may not be used with any of the other troubleshooting functions except the read and write functions.

4-5. STATUS/CONTROL LINES

4-6. Introduction

As part of the task of interfacing the general troubleshooter architecture with the UUT architecture, the pod makes specific assignments between the microprocessor lines and the troubleshooter. These assignments include the following:

- Bit number assignment of status lines
- User-writable control lines
- Bit number assignment of control lines

These assignments, which are summarized on a decal on the bottom of the pod, are listed in Table 4-2 and described in the following paragraphs:

NOTE

The terms status and control as used in this manual differ in meaning from the same terms used in the microprocessor manufacturer's literature. The troubleshooter considers input lines to the microprocessor to be status lines, and output lines to be control lines.

4-7. Status Line Bit Assignments

When a Read Status operation is performed, the troubleshooter displays the result in binary form, where "1" indicates a logic high status line and a "0" indicates a logic low status line. To determine which characters of the display correspond to specific status lines, refer to Table 4-2.

Table 4-2. Status and Control Line Bit Assignments

STATUS LINES			CONTROL LINES		
BIT NUMBER	SIGNAL		BIT NUMBER	SIGNAL	
	6809	6809E		6809	6809E
7	PWR FAIL	PWR FAIL	7	—	LIC
6	**$\overline{\text{DMA}}$	**TSC	6	—	BUSY
5	**MR	—	5	—	AVMA
4	**$\overline{\text{RESET}}$	**$\overline{\text{RESET}}$	4	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$
3	**$\overline{\text{HALT}}$	**$\overline{\text{HALT}}$	3	*BA	*BA
2	$\overline{\text{FIRQ}}$	$\overline{\text{FIRQ}}$	2	*BS	*BS
1	$\overline{\text{IRQ}}$	$\overline{\text{IRQ}}$	1	Q(CLK)	—
0	$\overline{\text{NMI}}$	$\overline{\text{NMI}}$	0	E(CLK)	—
**FORCING LINES			*USER WRITEABLE		

For example, if a Read Status operation is performed when the pod is in the 6809E mode and the $\overline{\text{NMI}}$ (Bit 0) and $\overline{\text{IRQ}}$ (Bit 1) lines are low with all other status lines high, the troubleshooter displays the message *READ @ STS = 1101 1100 OK.* Bit numbers 0 ($\overline{\text{NMI}}$) and 1 ($\overline{\text{IRQ}}$) are zero to indicate a logic low, while other meaningful bits are 1 to indicate a logic high. Bit 5, which has no meaning as a 6809E status line, is always represented by a zero in the troubleshooter display message.

4-8. User-Enableable Status Lines

The pod has several status line inputs which the operator can individually enable or disable by selecting the proper Setup message using the Setup function of the troubleshooter. For the 6809 mode, the enableable status lines are $\overline{\text{DMA}}$, MR, and $\overline{\text{Halt}}$. For the 6809E mode, the enableable status lines are TSC and $\overline{\text{Halt}}$.

When these enable lines are disabled, UUT-generated signals appearing at these inputs are prevented from affecting troubleshooter and pod operation. For example, a stuck-low $\overline{\text{Halt}}$ line would cause the microprocessor within the pod to complete the current instruction and then stop for an indefinite period of time, thus preventing normal troubleshooter/pod operation. When the Setup function of the troubleshooter is used to disable this input to the microprocessor (by means of hardware in the pod), the $\overline{\text{Halt}}$ signal is prevented from reaching the microprocessor within the pod, thus allowing the troubleshooter/pod interactions to take place normally.

The relevant Setup message that is used to disable or enable these lines is *SET-ENABLE xxxxxx?* where *xxxxxx* is the name of the status line. (There is one Setup message for each enable line.) Pressing the YES key on the troubleshooter enables the status line; pressing the NO key disables the status line.

NOTE

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

4-9. Forcing Lines

One of the troubleshooter error messages that may be displayed is the message *ACTIVE FORCE LINE (@ aaaa)-LOOP?* Forcing lines are a special category of status lines which, when active, can force the microprocessor into some specific state or action which causes a pod timeout. The *ACTIVE FORCE LINE* error message helps isolate status lines which are not functioning properly.

The forcing lines consist of $\overline{\text{DMA}}$, MR, $\overline{\text{Reset}}$, and $\overline{\text{Halt}}$ in the 6809 mode, and TSC, $\overline{\text{Reset}}$, and $\overline{\text{Halt}}$ in the 6809E mode. Notice all of these lines except Reset are user-enableable lines. If these user-enableable lines are disabled (using the troubleshooter Setup function), their inputs to the pod microprocessor are disabled, but the pod continues to monitor their condition; if they are active, the pod reports to the troubleshooter that a forcing line is active. Note that if these user-enableable lines are enabled (using the troubleshooter Setup function), they are not considered forcing lines even when they are active.

A typical way the *ACTIVE FORCE LINE* message would be encountered is as follows. Assume the pod and troubleshooter are connected to a UUT and the operator first turns on the power to the troubleshooter and the UUT (at power on the user-enableable lines are all enabled). When attempting a Read operation with the troubleshooter, the error message *POD TIMEOUT - ATTEMPTING RESET* appears. The operator enters the troubleshooter Setup function, disables the $\overline{\text{Halt}}$ enableable line, and again attempts a Read operation. This time the troubleshooter displays the message *ACTIVE FORCE LINE @ aaaa-LOOP?* (aaaa is the address specified for the Read operation).

To find out which line or lines are the active forcing lines, the operator presses the MORE key and the message *STS BTS 0000 1000* is displayed. According to Table 4-2, the $\overline{\text{Halt}}$ line is Bit 3, so the display message indicates that the $\overline{\text{Halt}}$ line is the active forcing line that is not functioning properly. While the $\overline{\text{Halt}}$ line was enabled, its improper function caused the pod timeout to occur. After the $\overline{\text{Halt}}$ line was disabled, its improper function caused the *ACTIVE FORCE LINE* error message to appear.

NOTE

It is possible to disable the reporting of active forcing lines by selecting the troubleshooter Setup function message SET-TRAP ACTIVE FORCE LINE? NO. The pod will still monitor the lines, but the troubleshooter will not interrupt its operation to display the ACTIVE FORCE LINE error message on the display. Sometimes it is useful to disable the reporting of active forcing lines, particularly if the information is not needed by the operator. Reporting of the error message is enabled at power on.

4-10. Interrupt Lines

Interrupt lines for the pod (in either mode) consist of $\overline{\text{NMI}}$, $\overline{\text{IRQ}}$, and $\overline{\text{FIRQ}}$. The $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ inputs are software disabled except during operation in the Run UUT mode. The $\overline{\text{NMI}}$ input is hardware disabled by the pod except during operation in the Run UUT mode. The default condition at power on is to disable the reporting of active interrupts. The reporting of active interrupts may be reenabled by selecting the troubleshooter Setup message *TRAP ACTIVE INTERRUPTS? YES.*

4-11. User-Writable Control Lines

The pod has two control lines (in either mode), BA and BS, which the troubleshooter can set high or low with the Write Control function. The Write Control function is described in the following paragraphs as it pertains to the pod. Note that the Write Control function only sets a line high or low for approximately 20 microseconds, just long enough to verify that it can be driven.

4-12. Control Line Bit Assignments

There are two troubleshooting functions which require the entry of binary digits to identify user-writable control lines. These functions are Write Control and Data Toggle Control.

When performing or programming either of these two functions, the operator is prompted for a binary number to identify the control line(s) to be written. Table 4-2 shows that these lines are assigned to bit numbers 0 through 4 when the pod is in the 6809 mode, and bit numbers 2 through 7 when the pod is in the 6809E mode.

For example, to perform a Write Control operation which writes both the writable control lines high in the 6809 mode, the operator enters the string 1100 when the troubleshooter prompts with the message *WRITE @ CTL =*. To write either of the two writable lines to the low state, the operator enters a 0 in place of 1 at the bit position which corresponds to the particular control line. Any unused lines are set to 0 by the troubleshooter.

When performing Bus Test or various other troubleshooter functions, the troubleshooter may detect that one or more control lines are not drivable. For example, suppose that during a Bus Test the troubleshooter detects that the BS line (bit 2) is not drivable. The troubleshooter displays the message *CTL ERR 0000 0100-LOOP?* The zeros and ones correspond to the bit numbers assigned to the control lines as listed in Table 4-2. Bit 2 is set to 1 because the BS line was detected as not drivable. The other error messages that pertain to non-drivable control lines use the same bit number assignments as listed in Table 4-2.

4-13. PROBE AND SCOPE SYNCHRONIZATION MODES

The operator may use the troubleshooter Synchronization function (selected with the SYNC key) to synchronize probe operation or the rear panel TRIGGER OUTPUT pulse to events on the pod microprocessor bus. There are three synchronization modes available:

A = Address Sync

D = Data Sync

F = Free-Run

If address sync or data sync is selected, both the probe and scope trigger output on the troubleshooter are synchronized to the UUT access. The scope trigger will pulse low shortly before the UUT access begins, and pulse high at the end of the cycle. If the probe stimulus mode is selected, the probe will pulse at the selected level (high or low) for the time between the two scope trigger pulses described above. For probe response, the probe will latch on the signal level present at its tip at the time of the second or high-going scope trigger pulse. For the pod (in either the 6809 or 6809E mode), there is no difference between address and data sync; they may be used interchangeably.

If free-run is selected, the probe stimulus pulses are generated at a frequency of approximately 1 kHz with a 1% duty cycle.

One recommended method for using the scope synchronization is to synchronize on the negative edge of the scope trigger using the address sync or data sync mode. This will trigger the scope at a time slightly before the UUT access, allowing you to see the entire UUT access. If you cannot verify that the scope is synchronized to the correct edge, use the pulse stimulus mode of the probe and display the signal on the scope. Since the probe stimulus pulses only occur during the UUT access (when the address sync or data sync

mode is selected), you can easily check whether the scope is synchronized to the correct edge. After verifying the correct synchronization, use the scope to look at the UUTON pulse to verify when the signal of interest should be valid. The relationship of these signals and the UUT access is described and illustrated in Section 5.

If the signal image on the scope is dim because of a low repetition rate, use the Quick Looping function described in the previous section titled Quick Looping Functions. The Quick Looping function will increase the repetition rate considerably, which makes the signal much easier to see on the scope.

Note that the scope trigger output pulses are always synchronized to the UUT accesses, even if free-run is selected. At power on the probe is in free-run, but the scope trigger output pulses are still synchronized to UUT accesses. For the pod (in either the 6809 or 6809E mode), the troubleshooter Synchronization function affects only the probe, not the scope.

4-14. PROBLEMS DUE TO A MARGINAL UUT

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

4-15. UUT Operating Speed and Memory Access

Some UUTs operate at speeds which approach the time limits for memory access. The pod contributes a slight time delay which causes memory access problems to become apparent.

4-16. UUT Noise Levels

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

4-17. Bus Loading

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

4-18. Clock Loading

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

The crystal oscillator in the pod (activated in the 6809 mode only) is designed to reject large amounts of shunt capacitance. It is far more resistant to shunt capacitance than the oscillator in the 6809 microprocessor.

4-19. POD DRIVE CAPABILITY

As a driving source on the UUT bus, the pod provides equal to or better than normal 6809 or 6809E current drive capability. All pod inputs and outputs (except the crystal on the 6809) are TTL-compatible.

4-20. LOW UUT POWER DETECTION

The pod has a UUT power detection circuit which constantly monitors the UUT power supply. If the UUT power supply drops below 4.5V or rises above 5.5V, this circuit produces an output to the troubleshooter which causes the troubleshooter to display a UUT power fail error message.

Also, anytime the UUT power supply drops below approximately 3.5V, all active pod outputs are disabled or written to their low logic level. This feature has been incorporated to protect UUT circuits from possibly being damaged by pod outputs when the UUT power supply drops below safe operating limits. The troubleshooter will display a *UUT POWER FAIL* message. When the proper operating power supplies have been restored to the UUT, the outputs of the pod will return to normal and the troubleshooter will be ready for additional testing.

Section 5
Theory of Operation

5-1. INTRODUCTION

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

5-2. GENERAL POD OPERATION

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

- Processor Section
- UUT Interface Section
- Timing Section

Each of these three sections and a UUT Power Sensing circuit are described in the following paragraphs:

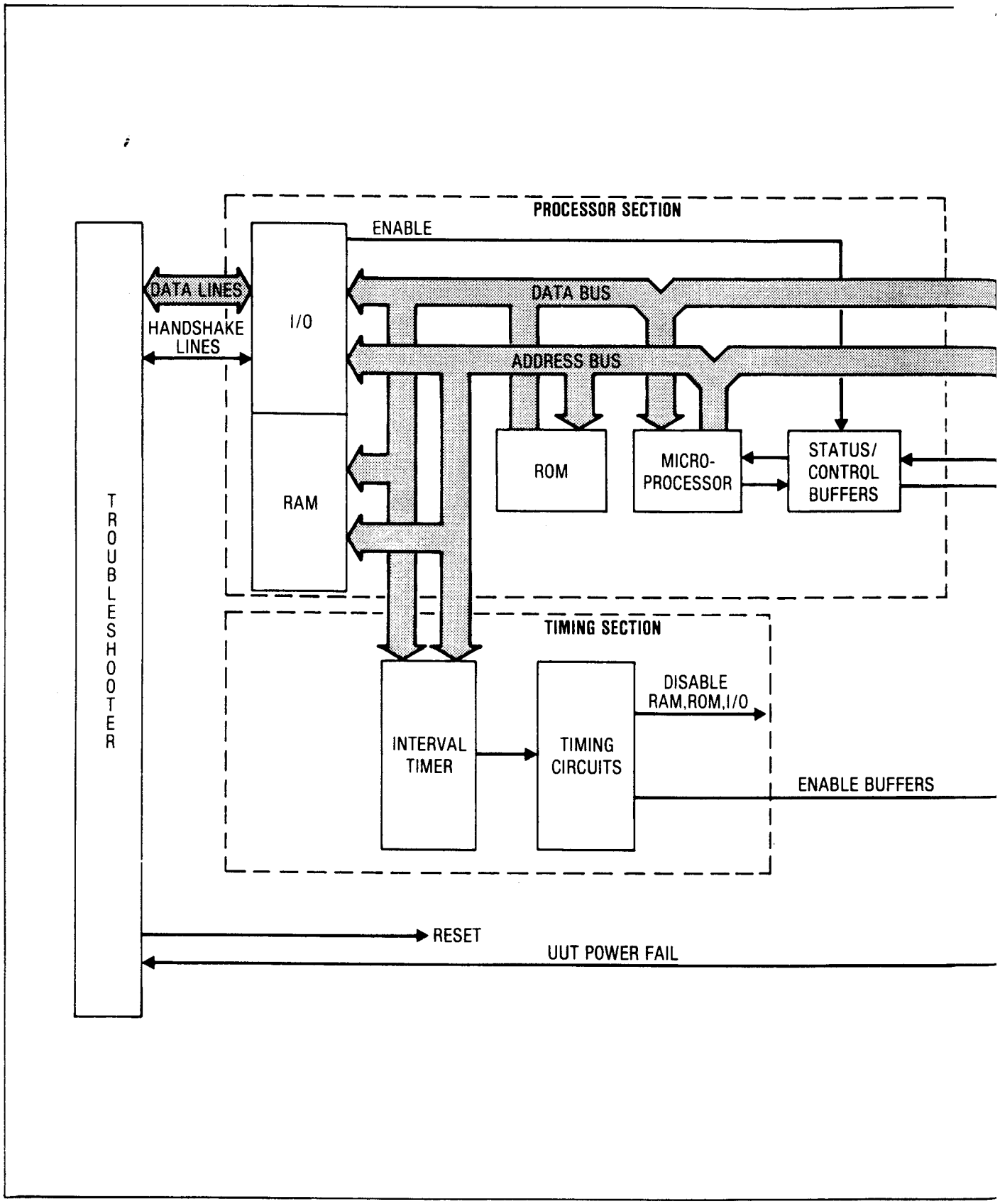
5-3. Processor Section

The Processor Section, shown in Figure 5-1, is made up of a microprocessor, RAM, ROM, and an I/O interface to the troubleshooter. These elements comprise a small computer system which receives troubleshooter commands and directs all pod operations during execution. All disrupting inputs such as reset or non-maskable interrupts are either automatically disabled by hardware in the pod, or may be disabled by the operator using the troubleshooter Setup function. Disabling these inputs helps prevent UUT faults from disabling the pod microprocessor.

The Processor Section has the capability of operating with the troubleshooter, or with the UUT, but not with both concurrently. The microprocessor spends most of its time monitoring the troubleshooter I/O interface for commands. During this time, called standby, the data and address buses of the Processor Section are isolated from the UUT Interface Section (although the pod sends signals to the UUT so that continuous read operations appear to be taking place at the reset addresses FFFE and FFFF in order to refresh any dynamic RAM).

5-4. UUT Interface Section

The UUT Interface Section, shown in Figure 5-1, includes the following elements:


PROCESSOR SECTION
ENABLE
DATA LINES
HANDSHAKE LINES
I/O
DATA BUS
ADDRESS BUS
RAM
ROM
MICRO-PROCESSOR
STATUS/ CONTROL BUFFERS
TROUBLESHOOTER
TIMING SECTION
INTERVAL TIMER
TIMING CIRCUITS
DISABLE RAM,ROM,I/O
ENABLE BUFFERS
RESET
UUT POWER FAIL

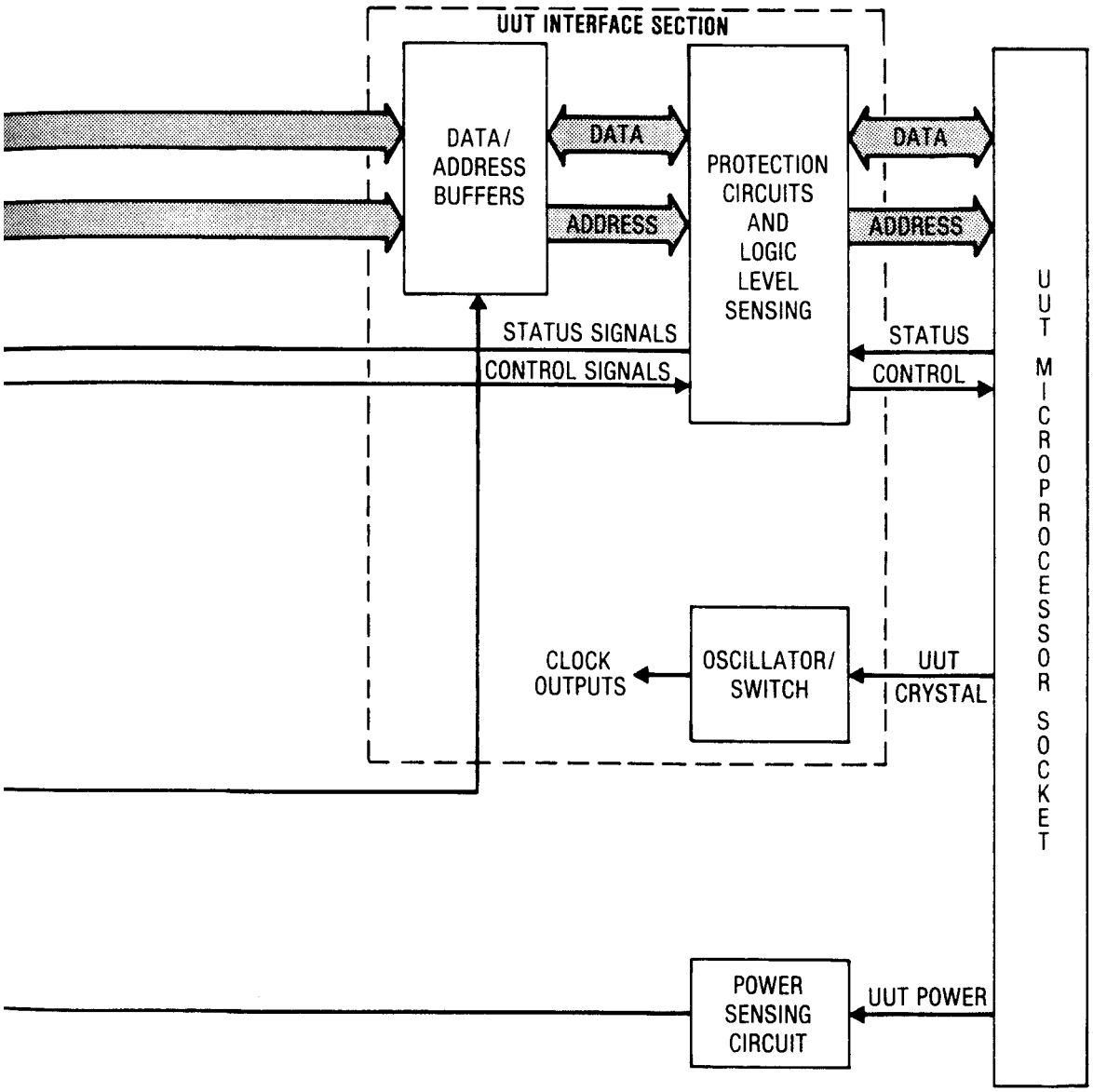


Figure 5-1. 6809/6809E Pod General Block Diagram

- Data and address buffers
- Protection circuits for signal lines
- Logic level detection circuits for data, address, status and control lines
- Crystal oscillator and analog switches to support either the 6809 or 6809E microprocessors

The data and address buffers are enabled to connect the microprocessor to the UUT, or disabled to isolate the microprocessor from the UUT. When the buffers are enabled, the microprocessor can address the UUT as commanded by the troubleshooter. When disabled, the Interface Section performs a continuous standby read at addresses FFFE and FFFF. Control of the buffers is maintained by the Timing Section.

Each line to the UUT contains a detection circuit. A detection circuit consists of an edge-triggered latch connected to the UUT side of the 100-ohm protection resistor. The latch stores the level at the UUT side of the protection circuit at the conclusion of each UUT operation. Each latch is then individually addressed and read by the Processor Section. The contents of each latch are then compared with the desired results as a means of detecting UUT bus drivability faults.

The crystal oscillator connects to the crystal in the UUT through an analog switch. The oscillator generates the clock signals needed when the pod is in the 6809 mode. If the pod is in the 6809E mode, the oscillator is bypassed by the analog switch. The functional differences between the 6809 and 6809E microprocessors are described in Section 3.

5-5. Timing Section

The primary function of the timing section is to cause the microprocessor to communicate with either the Processor Section or the UUT Interface Section at a time predetermined by the microprocessor itself. During communication with the Processor Section, the microprocessor receives commands from the troubleshooter; during communication with the UUT Interface Section, the microprocessor implements the commands received.

The Timing Section of the pod, shown in Figure 5-1, consists of an interval timer and an arrangement of timing circuits. The interval timer, preset by the microprocessor, determines the time at which the microprocessor switches from addressing the Processor Section (RAM, ROM and I/O) to addressing the UUT Interface Section (and the UUT). The timing is critical, since any attempt by the microprocessor to address the Processor Section with addresses meant for the UUT, or vice versa, would result in improper operation.

In their reset or standby state, the timing circuits cause the microprocessor to operate as part of the Processor Section, which includes an I/O port to the troubleshooter. When the troubleshooter issues a command which specifies a UUT read or UUT write operation, the microprocessor sets the interval timer for the length of time needed by the Processor Section to prepare to execute the command. The timing circuits modify this length of time if forcing lines from the UUT interrupt the command execution. When the specified time interval is completed, the UUTON cycle begins. (The timing of the UUTON cycle is shown in Figure 5-2.) The timing circuits disable the RAM, ROM, and I/O of the pod and enable the buffers in the UUT Interface Section. This action allows the microprocessor to address the UUT instead of the pod. At the same time, the microprocessor, having completed preparation for command execution, places a UUT address on the address bus and UUT data on the data bus (if the command being executed requires data).

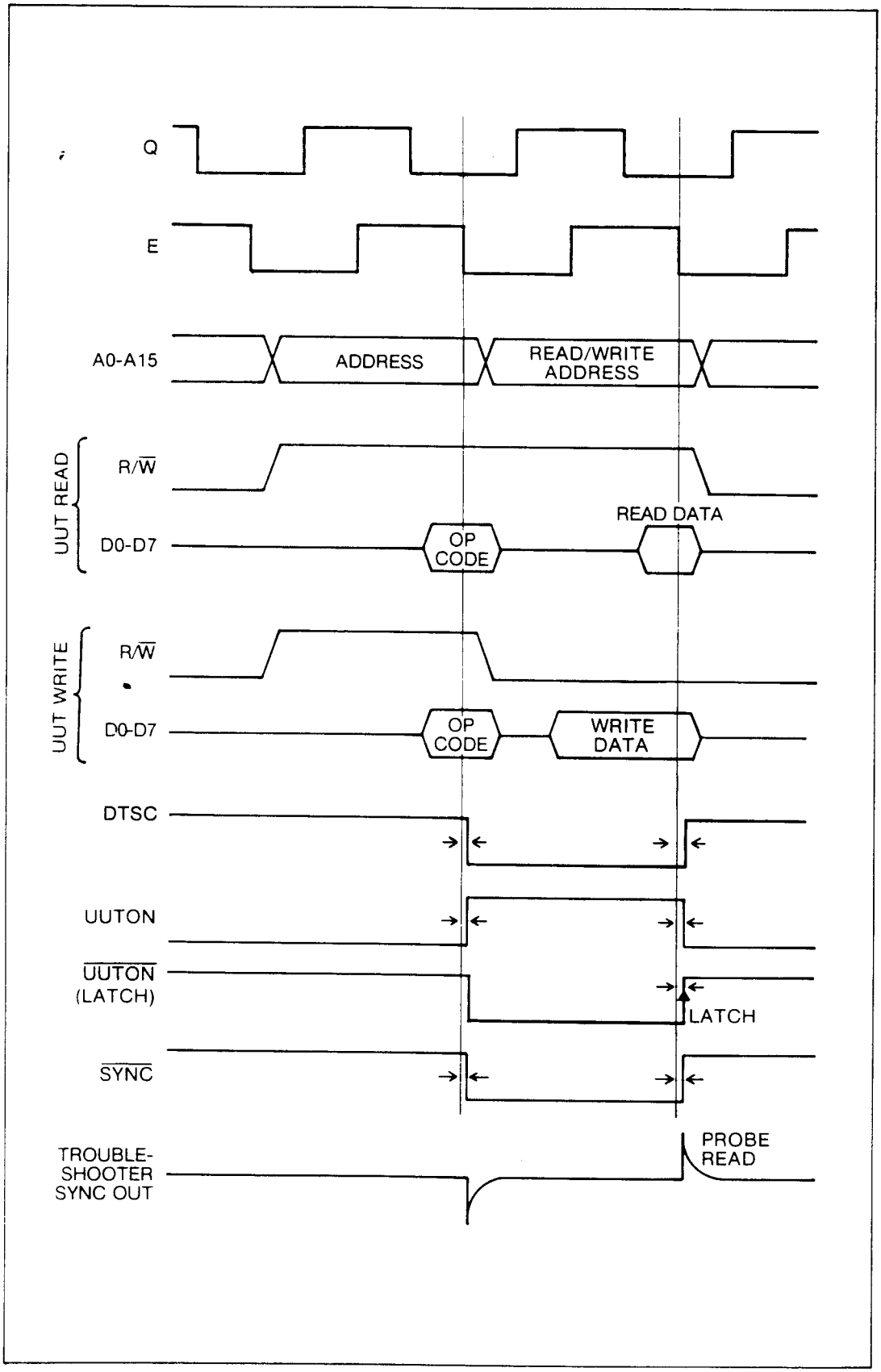


Figure 5-2. UUTON and Latch Signal Timing

At the conclusion of the UUTON cycle (which equals one period of the E clock), the timing circuits terminate the addressing of the UUT and the microprocessor again controls the RAM, ROM, and I/O of the Processor Section. The timing circuits also sample the latches within the logic level detection circuits to store the state of the UUT bus during the UUT bus transaction.

All data transactions (reads and writes) occur on the falling edge of the E signal, the trailing edge of the UUTON cycle. A $\overline{\text{SYNC}}$ pulse that coincides with the UUTON cycle is sent to the troubleshooter. The $\overline{\text{SYNC}}$ pulse allows use of the synchronized probe. The troubleshooter also differentiates the $\overline{\text{SYNC}}$ pulse to produce the rear panel trigger output pulse which allows the operator to synchronize an oscilloscope with the UUT access.

The Run UUT mode allows the pod to emulate the microprocessor. To enter the Run UUT mode, the Timing Section causes the pod to begin a UUTON cycle and to address the UUT through the UUT Interface Section. Unlike a normal UUTON cycle, however, the timing circuits do not allow the microprocessor to return to addressing the Processor Section of the pod. In addition, the Reset line and all forcing lines and interrupts are enabled. In this configuration, the pod appears to the UUT to be no different from the UUT microprocessor. To terminate the Run UUT mode, the troubleshooter resets the pod which returns control to the Processor Section.

5-6. UUT Power Sensing

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 approximately 4.5V or rises above approximately 5.5V.

Also, anytime the UUT power supply drops below approximately 3.5V, all active pod outputs are disabled or written to their low logic level. This feature has been incorporated to protect 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 supplies have been restored to the UUT, the outputs of the pod will return to normal, and the troubleshooter will be ready for additional testing.

5-7. DETAILED BLOCK DIAGRAM DESCRIPTION

A detailed block diagram of each major pod section is presented in Figure 5-3. Each major section is described in the following paragraphs along with a description of the Self Test circuit.

5-8. Detailed Description of the Processor Section

The Processor Section of the pod consists of the following components:

- Microprocessor, U26
- ROM, U2
- RAM (128 bytes X 8 bytes), U1
- I/O ports A and B, U1
- Address decoder, U11 (on Interface PCB schematic)
- Status line buffers, U4, U10, U11, U14, U19

- Control line buffers, U10, U15, U19
- $\overline{\text{SYNC}}$ driver circuit, U17
- Processor mode switch (6809/6809E), S1

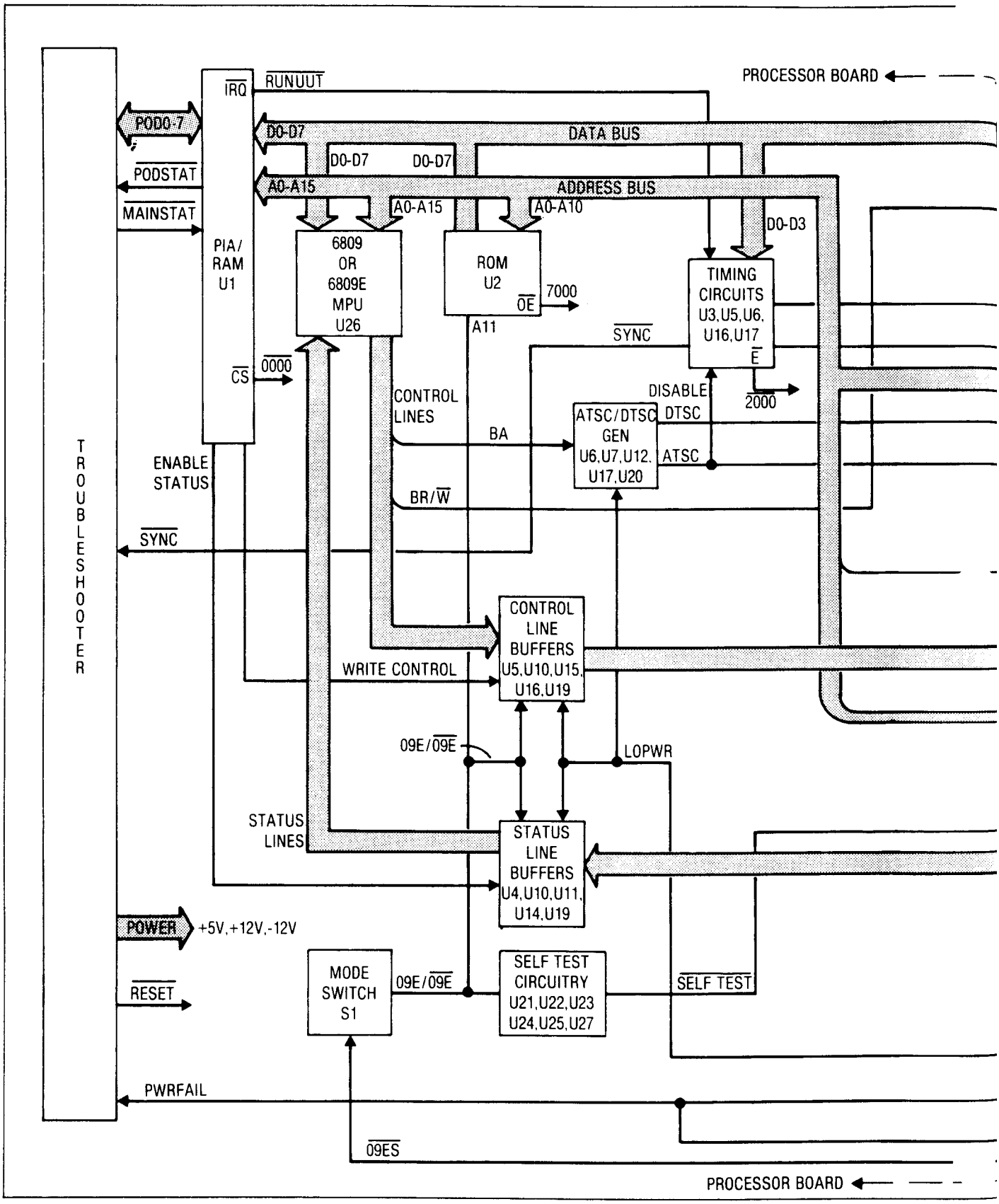
The pod can operate in either the 6809 mode or the 6809E mode. To operate in a desired mode, a corresponding microprocessor is inserted into the U26 socket, and the mode switch, S1 (which consists of four switches, S1-1 through S1-4), is set as required. Switch S1-1 is used to generate the two mode control signals, signal 09E and the complementary signal $\overline{09E}$. Signal 09E is true (high) when the 6809E mode is selected. Signals 09E and $\overline{09E}$ also switch address bit 11 on the ROM, U2, to allow different software to be used with each microprocessor. The remaining three switches, S1-2 through S1-4, reconfigure several buffers and analog switches as required for proper operation.

The E clock is buffered with bidirectional driver U10 between the UUT and the microprocessor. Switches S1-2 and S1-3 are used to select the correct timing for the E clock used in the pod; the timing signal is PE (Pod E). Signal PE is derived from the microprocessor side of the buffer in the 6809 mode (when the microprocessor is generating E), and is derived from the UUT side of the buffer in the 6809E mode (when the UUT is generating E). The use of switch S1-4 is described in Section 6. The remainder of the Processor Section description is applicable to either the 6809 or 6809E configuration.

The pod communicates with the troubleshooter via Port A of the PIA (I/O port A), U1. This communication is controlled via two handshake lines, $\overline{\text{MAINSTAT}}$ and $\overline{\text{PODSTAT}}$. These handshake lines allow asynchronous communication between the pod and troubleshooter. The troubleshooter operates at a fixed clock frequency, and the pod operates at a frequency determined by the UUT. The microprocessor monitors the handshake line, $\overline{\text{MAINSTAT}}$, at I/O port B, when waiting for troubleshooter commands. The microprocessor addresses I/O port B by means of address lines A0 through A7 and the address decoder, U11. The address decoder decodes address lines A12 through A14 to produce the signal $\overline{0000}$ which selects the PIA, U1. This PIA also contains 128 (7F hex) bytes of RAM, which is the only RAM present in the pod.

The troubleshooter places a low on the $\overline{\text{MAINSTAT}}$ line when a command is placed on lines POD0 through POD7. The microprocessor responds by addressing I/O port A of U4 and reading each byte of the troubleshooter command. As each byte is received, the handshaking lines operate as shown in the upper portion of Figure 5-4 to insure that no data is lost. For the pod to send data to the troubleshooter, the handshaking sequence is identical; the troubleshooter again places a low on $\overline{\text{MAINSTAT}}$ indicating that it expects to receive some data. The microprocessor in the pod writes $\overline{\text{PODSTAT}}$ low when the byte of data is available at I/O port B. Whenever the troubleshooter sends a command and receives an unexpected response from the pod, the troubleshooter resets the pod and resubmits the command.

Each troubleshooter command causes the microprocessor to execute a corresponding software routine contained in ROM, U2. When the routine is executed, it allows the microprocessor to address the UUT. In order to perform a UUT access, the interval timer is first preset by writing to address 2000 with the desired count on the data bus. The processor then performs all necessary internal operations in preparation for addressing the UUT. These necessary operations are timed precisely so that when the timer reaches the end of the time interval and switches the address and data buffers in the Interface Section over to the UUT, the microprocessor is performing the required read or write operation. The timer operation is described in more detail in the section titled Detailed Description of the Timing Section.


PROCESSOR BOARD
IRQ
RUNUUT
POD0-7
D0-D7
DATA BUS
D0-D7
D0-D7
PODSTAT
A0-A15
ADDRESS BUS
MAINSTAT
A0-A15
A0-A10
D0-D3
PIA/
RAM
U1
6809
OR
6809E
MPU
U26
ROM
U2
OE
7000
A11
TIMING
CIRCUITS
U3,U5,U6,
U16,U17
E
SYNC
CS
0000
2000
DISABLE
DTSC
CONTROL
LINES
BA
ATSC/DTSC
GEN
U6,U7,U12,
U17,U20
ATSC
TROUBLESHOOTER
ENABLE
STATUS
BR/W
SYNC
CONTROL
LINE
BUFFERS
U5,U10,U15,
U16,U19
WRITE CONTROL
09E/09E
LOPWR
STATUS
LINES
STATUS
LINE
BUFFERS
U4,U10,U11,
U14,U19
POWER
+5V,+12V,-12V
MODE
SWITCH
S1
09E/09E
SELF TEST
CIRCUITRY
U21,U22,U23
U24,U25,U27
SELF TEST
RESET
PWRFAIL
09ES
PROCESSOR BOARD

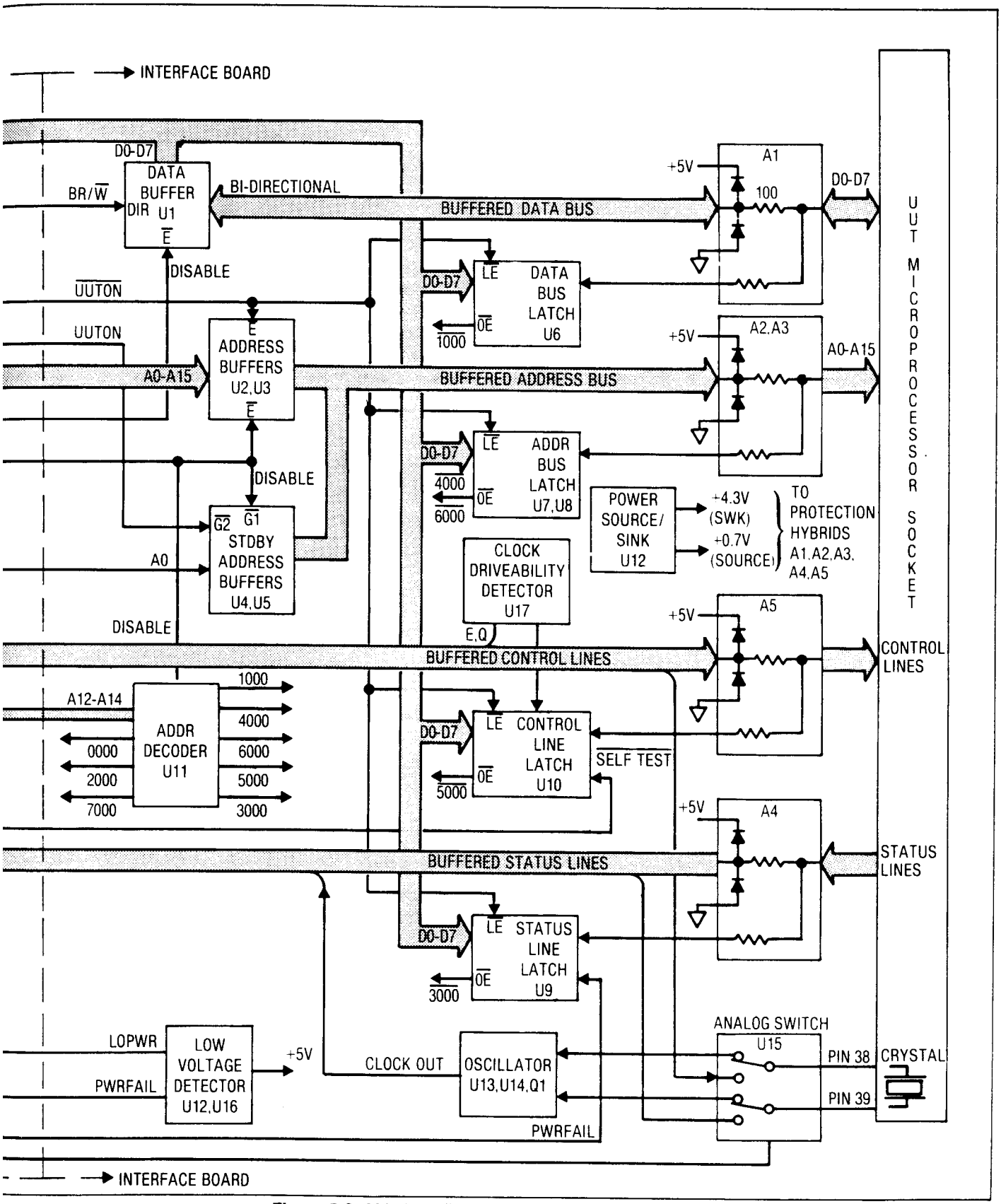


Figure 5-3. 6809/6809E Pod Detailed Block Diagram

The software routines that are invoked by the commands from the troubleshooter also direct the actual write and read functions of the UUT. As part of each routine, the pod transmits any response data back to the troubleshooter and produces a fault byte which reflects the current condition of the pod and the UUT. During the transmission of data and the fault byte back to the troubleshooter, the handshake lines operate as shown in the lower portion of Figure 5-4. The handshake insures that no data is lost during the transmission process.

The microprocessor has the capability of writing control lines BA and BS either high or low as a means of verifying that they can be driven. Normally, these two lines are not modified before they are sent out to the UUT. However, control line BS is modified in one case. Lines BS and BA are decoded together by the UUT to indicate a reset acknowledge (BA = 0 and BS = 1). Whenever the troubleshooter resets the pod, the microprocessor is also simultaneously reset. In response to this, a reset acknowledge is generated. To suppress the reset acknowledge, the circuitry of U15 and U5 is used to suppress BS from going high the first time a reset is received by the pod. In the Run UUT mode, reset acknowledges are allowed to pass through unimpeded.

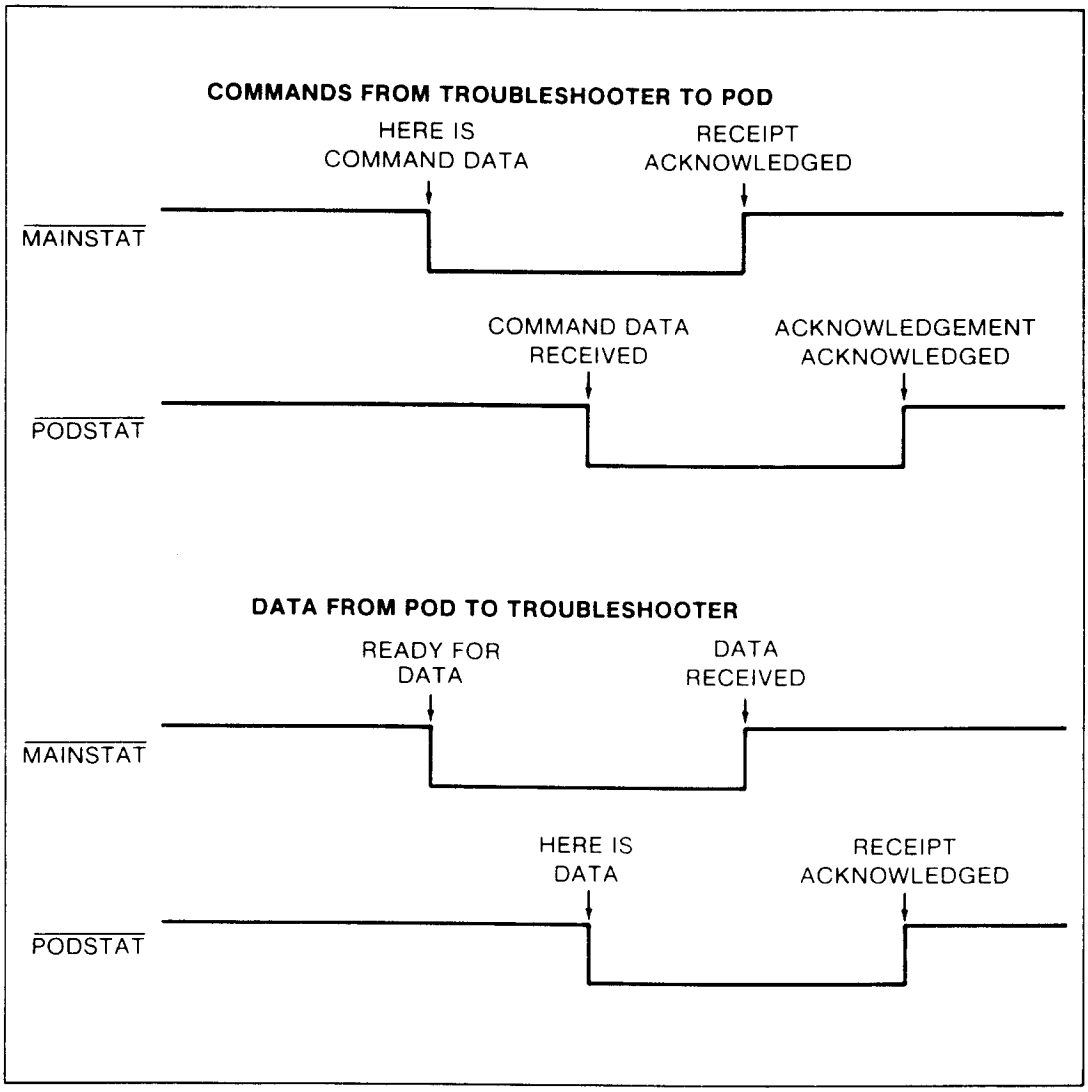


Figure 5-4. Handshake Signals

Analog switch U4 is used to prevent the Reset line from the UUT from reaching the microprocessor. In the Run UUT mode, switch S4 is closed allowing the UUT Reset line to reach the microprocessor. By using an analog switch, the pod is able to emulate precisely the behavior of the UUT microprocessor Reset line.

5-9. Detailed Description of the UUT Interface Section

The UUT Interface Section includes the following components:

- Bidirectional data buffer, U1
- Protection circuits, A1 through A5
- Address buffers, U2 and U3
- Standby address buffers, U4 and U5
- Sensing latches, U6, U7, U8, U9 and U10
- Clock drivability detector, U17
- UUT Power Supply Monitor, U12
- Low Voltage detector, U16
- Crystal Oscillator circuit, U13, U14, Q1
- Analog switch, U15

5-10. DATA LINES

The bidirectional data buffer, U1, is enabled by the signal DTSC (Data Tristate Control), an output from the Timing Section of the pod. The DTSC signal places the data buffer in a tristate condition while internal pod communication is taking place, and enables the data buffer if a UUT access is to be performed. The direction of the data driver is determined by the signal DRIVE, which is derived from the R/W line of the microprocessor. In the Run UUT mode, the data buffer is always enabled except when placed in a high impedance state by the signal DTSC in response to the high impedance state of the microprocessor.

All data passing between the pod and the UUT is fed through a series of protection circuits, with one circuit per line. Each protection circuit consists of a 100-ohm resistor in series with the line, and a pair of clipping diodes connected to ground and +5 volts.

The data lines are also equipped with logic level detection circuits, with one circuit per line. The detection circuits consist of a series of latches coupled to the UUT side of the respective protection circuits. A series resistor at the input of each latch provides overvoltage protection.

The data lines are coupled to the inputs of latches in U6 by lines LD0 through LD7. The input to each latch is logic high if the line is driven high, and logic low if the line is driven low. The Latch signal from the Timing Section latches the data line logic levels at the falling edge of the UUTON cycle (shown in Figure 5-2) to store the logic levels representing the state of each data line.

At the conclusion of a UUT write operation, latches U6 are addressed by the microprocessor. Address decoder U11 produces the signal $\overline{1000}$ which places the contents

of the latches on the data bus. The microprocessor compares the contents of the addressed latches with the intended write data. Any difference between the contents of the latches and the intended data is considered a data drivability error.

5-11. ADDRESS LINES

Address buffers U2 and U3 and Standby address buffers U4 and U5 are responsible for providing the address line output to the UUT. In the standby mode, when internal pod communication is taking place, the standby buffers present the reset addresses FFFE and FFFF to the UUT in a randomly alternating manner. When a UUT access is being performed, the address buffers present the address commanded by the microprocessor to the UUT. This arrangement prevents the troubleshooter from addressing sensitive UUT addresses except when commanded by the troubleshooter (and thus the operator).

The address and standby buffers are controlled by the signal ATSC (Address Tristate Control) and the signal UUTON; both signals are outputs from the Timing Section. The address buffers are enabled and the standby address buffers are disabled when the pod is performing a UUT access. The address buffers are disabled and the standby buffers enabled when internal pod communication is taking place. In the Run UUT mode, the address buffers are always enabled and the standby buffers are always disabled, except that the address buffers can be placed in a tristate condition by ATSC if the microprocessor itself places the address bus in a tristate condition. As described for the data lines, the address lines are equipped with logic level detection circuits, with one circuit per line. The detection circuits consist of a series of latches coupled to the UUT side of the respective protection circuits. A series resistor at the input of each latch provides overvoltage protection.

In a manner similar to that described for the data lines, all UUT addresses are fed through a series of protection circuits. The protection circuits consist of a 100-ohm resistor in series with the line, and a pair of clipping diodes connected to ground and +5 volts.

The address lines are coupled to the inputs of latches U7 and U8 by lines LA0 through LA15. The input to each latch is logic high if the line is driven high, and logic low if the line is driven low. The Latch signal from the Timing Section latches the address line logic levels (at the time shown in Figure 5-2) to store the logic levels representing the state of each data line.

At the conclusion of a UUT operation, the latches are separately addressed by the microprocessor. Address decoder U11 produces the signals $\overline{4000}$ and $\overline{6000}$ which places the contents of the latches on the data bus, one byte at a time. The microprocessor compares the contents of the addressed latches with the actual address. Any difference between the contents of the latches and actual address is considered an address drivability error.

5-12. CLOCK DRIVABILITY DETECTOR

The quad comparator, U17, generates clock drivability information. U17 is configured as a dual window comparator that compares the integrated clock waveforms (both E and Q) with a 1V to 3V window. Thus, if either clock signal is stuck high or low, the window comparator goes low; the information can be read at latch U10. If the clock signals are operating properly, the input to the window comparator will be one half of the peak value (approximately 2V). Since this value is within the window, the comparator outputs remain high when the clock signals are operating properly.

5-13. UUT POWER SUPPLY MONITOR

U12 is an IC designed by Fluke that contains a window comparator used to monitor UUT power supply voltages. Voltages outside of the window of approximately 4.75V to 5.25V

cause the PWRFAIL signal to go high. The pod will still operate in this condition, but the operator is informed of the condition with the message *BAD POWER SUPPLY.*

5-14. LOW VOLTAGE DETECTOR

In addition to the window comparator in U12, U16 is another comparator designed to detect a low voltage condition. If the UUT power supply drops below approximately 3.5V, the output of U16, LOPWR, goes high. Signal LOPWR is used to force all UUT connections to a low current state so that they cannot deliver more than 1 mA into any one line of the UUT. The LOPWR signal effectively places most of the lines in a tristate condition, such as the address and data lines, and forces the other lines low. The reason for this is to protect a UUT without power from being damaged by active pod inputs. In addition, the LOPWR signal is logically ORed with the PODSTAT handshake line so that the pod is unable to communicate with the troubleshooter. When this UUT low power condition occurs, the troubleshooter displays the message *UUT POWER FAIL.*

5-15. CRYSTAL OSCILLATOR AND ANALOG SWITCH

The pod contains a crystal oscillator that is designed to operate from the crystal on the UUT when the pod is in the 6809 mode. Unlike the oscillator in the microprocessor itself, this discrete oscillator is able to reject the capacitance of the UUT cable. In addition, the crystal lines may be driven with a TTL clock source in the same way as the 6809 microprocessor. Transistor array U14 comprises the oscillator along with Q1 and U13 which function as a buffer to drive the signal from the Interface PCB to the Processor PCB.

When the pod is in the 6809E mode, however, the pod does not use an oscillator, and uses the microprocessor pins 38 and 39 for entirely different functions. To allow the use of the different functions, an analog switch is used to switch these two signals between the oscillator (used in the 6809 pod) and the status and control line buffers (used in the 6809E pod).

5-16. STATUS AND CONTROL LINES

The status and control lines are provided with protection circuits, logic level detection circuits, and latches. These circuits and latches operate in a manner similar to those provided for the data and address lines. Refer to the previous sections titled Data Lines or Address Lines for a functional description.

5-17. Detailed Description of the Timing Section

The Timing Section consists of the interval timer U3, flip-flops U5 and U6, and the ATSC and DTSC generators consisting of U6, U7, U12, U17, and U20.

5-18. INTERVAL TIMER

As mentioned in the description of the Processor Section, the microprocessor executes troubleshooter commands by first setting the interval timer and then performing all necessary internal operations in preparation for a UUT access. The interval timer is set to a time equal to the amount of time required by the microprocessor to perform all necessary internal operations.

A timing diagram of the interval timer is shown in Figure 5-5. For the example shown, a UUT read has been commanded by the troubleshooter. The microprocessor is reading and executing the instruction LDA ,Y where the Y register contains the desired UUT address. In this case, when the microprocessor is to read at the indicated address, the interval timer will have timed out causing the Interface Section to address the UUT, and not the pod. The timing for a UUT write is identical except that the operation performed is STB ,Y where the data to be written is contained in the B accumulator. The operation of the interval timer is described as follows.

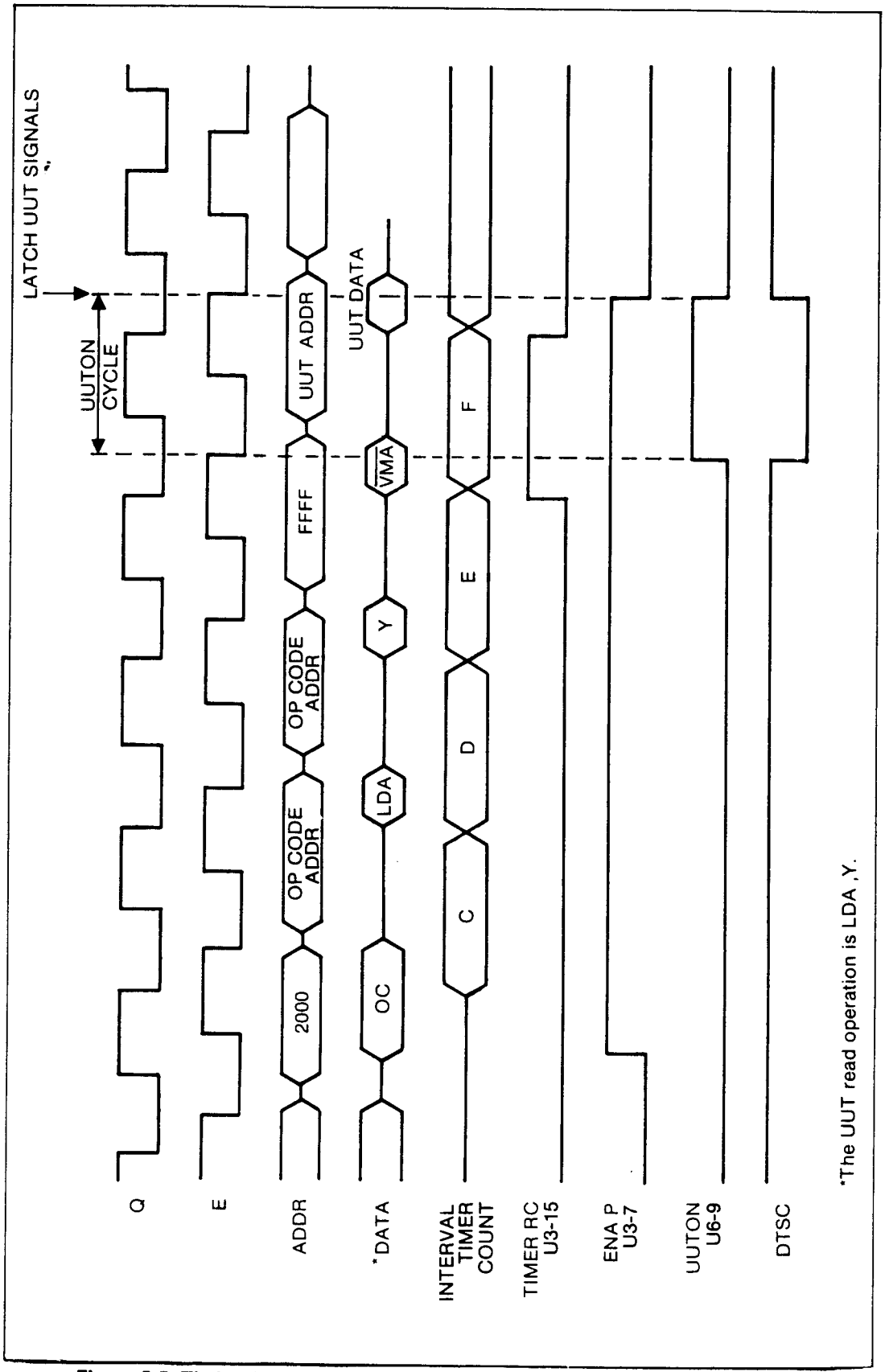


Figure 5-5. Timing Diagram of the Interval Timer During a UUT Read Operation

The interval timer is preset by writing to address 2000 with the desired count on the data bus. (The count that is written to address 2000 in Figure 5-5 is 0C.) Address decoder U11 decodes the address 2000 and provides an output which, when logically ORed with signals Q and R/W, is used to preset the count in the interval timer. The timer counts cycles of the Q clock and produces an RC (Ripple Counter) output when the count reaches 15 (0F hex). The RC output sets the input to the D flip-flop U6 high. U6 is clocked by the main E clock. Thus, on the next falling edge of E, the Q output of U6-9 (the UUTON signal) goes high. The counter continues counting, and on the next falling edge of Q, the RC output again goes low. This is clocked through U6 by the E clock. Thus, the width of the UUTON pulse is one clock period of the E clock. On the trailing edge of UUTON, a low signal is clocked through U5-6 which forces a low at U3-7 (the Enable P input) through U16. Thus, at the end of the UUTON cycle, the counter U3 is disabled from counting any further. The counter may be reenabled only by another write to address 2000.

The operator may set the interval timer count by writing a different count to the special address E0000. Figure 5-6 shows how different timer counts affect the time when the logic levels of the UUT lines are latched for the 6809E mode. In this example the UUT operation being performed is a read at FFFF. With the normal interval timer count the AVMA and LIC control lines are always high when latched, and the BUSY control line is always low when latched. As shown at the top of Figure 5-6, when the operator sets the interval timer count to 0E or to 0D, the latch occurs earlier in the cycle. By using different timer counts the operator can cause the logic levels of the control lines to be latched at times other than the last cycle of an instruction. This process is useful for verifying that these control lines a drivable both high and low. (A Bus Test performs this drivability test automatically.)

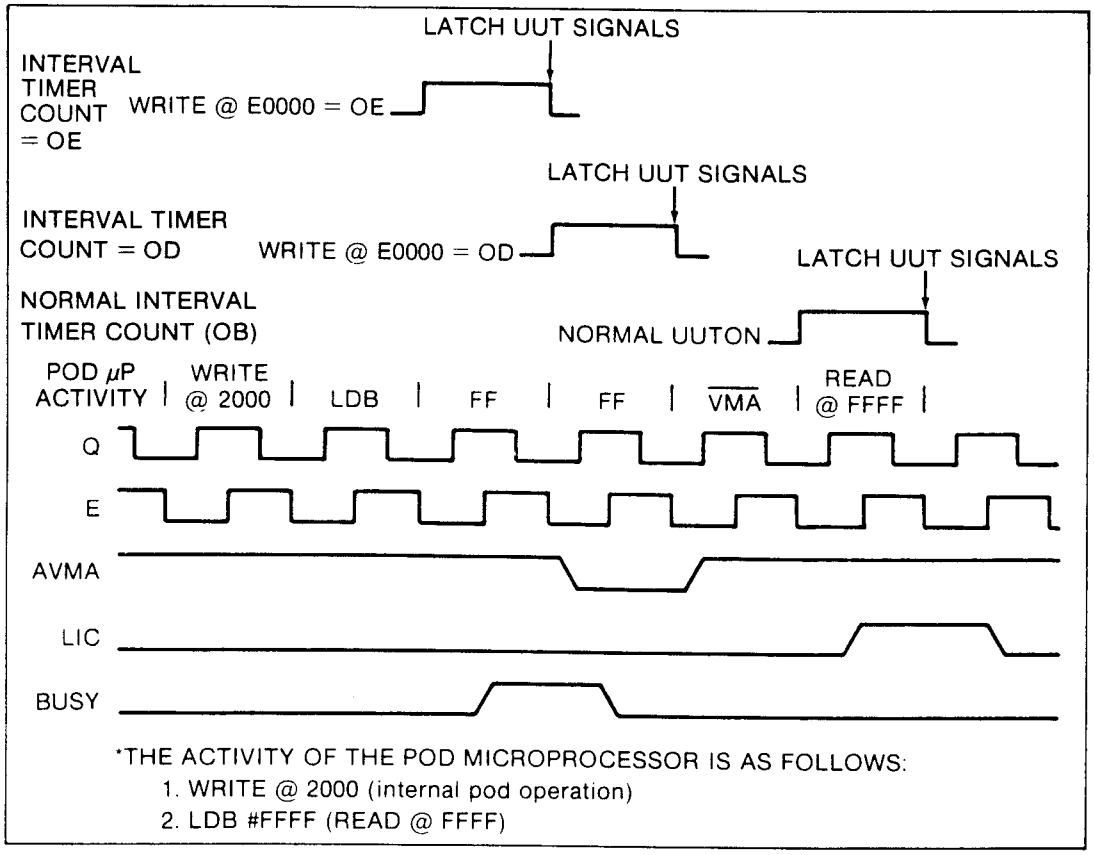


Figure 5-6. Timing of Latch in Relation to Different Internal Timer Counts (6809E Mode Only)

5-19. ATSC AND DTSC GENERATORS

The ATSC and DTSC generators are used to place the address and data buffers in a high impedance state when required to emulate the microprocessor. The generator is also used to extend or delay the UUTON pulse in response to a forcing line. For example, consider the case where a UUT access is about to be performed; assume that after the interval timer has been set and is counting cycles of the Q clock, the DMA line is made active by the UUT in order to acquire the bus and refresh some dynamic RAM. If the pod released the bus as requested, the interval timer would continue to count and would initiate the UUT cycle at the wrong time. As another example, consider the case where the UUT access has begun when a DMA bus request is granted. In this case, the UUT access would again occur at the wrong time.

To protect against either of these cases, the ATSC generator is used. In the first case, when ATSC goes high (indicating that the address bus in the microprocessor is in a high impedance state), the counting ability of the interval timer is disabled for the duration of the high impedance state. Thus, the count on the interval timer is effectively extended as required to provide a correct UUT access. In the second case, the interval timer is again disabled, but after the UUTON cycle has begun. Thus, the UUTON pulse is not allowed to go low as long as the high impedance state exists. This extends the UUTON cycle as required. Since the actual data transfer and latch enables occur on the trailing edge of the UUTON cycle, extending the UUTON cycle does not interfere with the data transfer.

The ATSC and DTSC generators decode the high impedance state of the microprocessor in several ways. The primary method is to monitor the BA (Bus Available) line. The BA line is usually made true when the microprocessor places the address and data buses in the high impedance state. In addition, there are delays between the time the BA signal goes low and the time that the microprocessor actually regains control of the bus. These delays are simulated by the generator. If the $\overline{\text{DMA}}$ or $\overline{\text{HALT}}$ lines are pulled low, the 6809 microprocessor address and data buses are placed in a high impedance state. For the 6809E microprocessor, the TSC line also places the address and data buses in a high impedance state. The TSC line does not activate the BA line; therefore, the ATSC generator decodes the TSC line directly.

The ATSC generator also responds to the LOPWR line, the line that senses when the UUT power drops below approximately 3.5V. In response to an active LOPWR line, the ATSC line forces all the address lines to the high impedance state in an effort to prevent damage from occurring to the UUT. DTSC is functionally equivalent to the ATSC signal logically ORed with the $\overline{\text{UUTON}}$ signal. DTSC is always true (which means the data buses are always in a tristate condition) unless the UUTON cycle is in progress. In the Run UUT mode, both ATSC and DTSC work together to emulate the high impedance state of the microprocessor.

5-20. RUN UUT MODE TIMING

The Run UUT mode is commanded by the $\overline{\text{IRQ}}$ output of U1. This output is caused by writing the data value 1 to address 0084. The resulting signal, $\overline{\text{RUNUUT}}$, is logically ORed with $\overline{\text{UUTON}}$ to form the signal $\overline{\text{LRUUT}}$, (Latched Run UUT). Signal $\overline{\text{LRUUT}}$ is used to set the flip-flop output U6-9 high so that the UUTON pulse is extended indefinitely. The $\overline{\text{LRUUT}}$ signal also enables all of the disabled status lines ($\overline{\text{RESET}}$ and $\overline{\text{NMI}}$), and all other enableable lines ($\overline{\text{DMA}}$, $\overline{\text{HALT}}$, and MR for the 6809, and TSC and $\overline{\text{HALT}}$ for the 6809E). The only way to clear signal $\overline{\text{LRUUT}}$ is for the troubleshooter to send the signal $\overline{\text{9010RESET}}$ which clears the PIA, U1, the counter U3, and the flip-flops U5 and U6.

5-21. Detailed Description of the Self Test Circuit

During the pod self test, the pod outputs are connected to the pod inputs, causing the pod

to appear to the troubleshooter to be a small, known UUT. Because the UUT is a known UUT, the troubleshooter can perform a series of operations and compare the expected behavior of the UUT with the actual behavior to determine whether or not the pod is operating properly.

When the pod ribbon cable plug is inserted into the self test socket, ground is applied through the ribbon cable to pin 1 and switch S1-4 to generate the signal $\overline{\text{SELFTEST}}$. This line is sensed by latch U10 and is used to notify the troubleshooter that the pod is in the self test configuration.

The address lines are connected back to the data lines through buffers U24 and U25. Control line BS is used to select which of these buffers is active, and thus determine whether the data bus will be connected to the high order address lines or to the low order lines. Performing a series of reads on this UUT, with BS written high and then low, allows a check of all address and data lines.

Two mode control signals, $\overline{\text{09E}}$ and 09E, are used to configure the socket for the two different modes, 6809 and 6809E. Signal 09E is active in the 6809E mode, and signal $\overline{\text{09E}}$ is active in the 6809 mode. These lines are controlled by switch S1-1.

A 4 MHz clock signal is generated by U23. In the 6809 mode, the 4 MHz clock signal is applied via U22 to pin 38 of the self test socket. In the 6809E mode, the 4 MHz clock signal is divided by U21 into two quadrature outputs which are applied to the E and Q clock inputs. The E input is tied to the $\overline{\text{Reset}}$ line, and the Q input is tied to the $\overline{\text{NMI}}$ line. Because the state of the E and Q clocks is always known (when the latches are enabled), the state of the $\overline{\text{Reset}}$ and $\overline{\text{NMI}}$ lines is also always known.

In the 6809 mode, control line BA is used to write $\overline{\text{DMA}}$, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, MR, and $\overline{\text{HALT}}$. In the 6809E mode, BUSY is used to write $\overline{\text{IRQ}}$, AVMA is used to write $\overline{\text{FIRQ}}$, and LIC is used to write TSC.

All forcing lines and interrupts can be set to the active state which allows testing of enabling hardware. A +5V dc voltage is applied to pin 8 to simulate UUT power and check the UUT Power Sensing circuit.

Section 6
Troubleshooting

WARNING

THESE INSTRUCTIONS ARE FOR USE BY QUALIFIED PERSONNEL ONLY. TO AVOID ELECTRIC SHOCK, DO NOT PERFORM ANY INSTRUCTIONS UNLESS YOU ARE QUALIFIED TO DO SO.

6-1. INTRODUCTION

This section provides troubleshooting information for the pod, and includes repair precautions and disassembly procedures.

The troubleshooting guidelines presented in this section are intended to assist in the isolation of faults within the pod. If you do not want to service the pod yourself, or if attempted troubleshooting fails to reveal the pod fault, you may ship the pod to the nearest Fluke Technical Service Center for repair. If requested, a free cost estimate will be provided before any repair work is performed. Refer to the troubleshooter operator manual or service manual for a list of Fluke Technical Service Centers.

If pod shipment is necessary, the pod should be shipped in its original shipping container if it is available. 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.

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.

Table 6-1. Required Test Equipment for Pod Troubleshooting

EQUIPMENT TYPE	REQUIRED TYPE
Micro System Troubleshooter	Fluke 9000 Series
Interface Pod	Fluke 9000A-6809
Digital Multimeter	Fluke 8020
Oscilloscope	Tektronix 485 or equivalent

- 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, vinyl, and styrofoam from the work area.
- Use a grounded soldering iron with a rating of 25 watts or less to prevent overheating the pcb assembly.
- When shipping the pod, always place the pod in a static-free plastic bag.

6-2. DETERMINING WHETHER THE POD IS DEFECTIVE OR INOPERATIVE

The first task of troubleshooting the pod is to determine whether the pod 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.

Depending on the results of the pod self test and the pod behavior when connected to a known good UUT, the pod may be categorized in 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 the section titled Troubleshooting a Defective Pod.
- Inoperative Pod: The pod is unable to complete the pod self test and the troubleshooter displays an *ATTEMPTING RESET* message. Refer to the section titled Troubleshooting an Inoperative Pod.
- Suspected Defective Pod: The pod passes the pod self test but exhibits abnormal behavior when connected to a known good UUT. Refer to the section titled Extended Troubleshooting Procedures.

6-3. TROUBLESHOOTING A DEFECTIVE POD

6-4. Introduction

This section tells what to do if the troubleshooter displays the following message when the pod self test is performed: *POD SELF-TEST 6809 FAIL xx* (*xx* represents a self test failure code). If instead the troubleshooter displays an *ATTEMPTING RESET* message, refer to the section titled Troubleshooting an Inoperative Pod.

The procedures for troubleshooting a defective pod are based on the information reported by the self test failure codes. In addition to the test routines that are performed on the pod by the troubleshooter in the normal pod self test routine, the pod has an

internal enhanced self test which performs some more thorough test routines. Together, the standard self test and the enhanced self test provide information that can enable the operator to locate the problem or problems that are causing the pod failure.

The recommended method for troubleshooting a defective pod is to recreate the self test routine that detected the pod failure and to use the routine as the starting point for tracing the problem. The following paragraphs describe how to prepare for troubleshooting, how to determine which self test routine failed, and how to recreate the self test routine.

6-5. Interpreting the Results of the Pod Self Test

You are now ready to explore the possible causes of pod failure that are indicated by the pod self test failure codes.

Note that the very 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 since they are essential for accepting the self test commands and communicating the results to the troubleshooter.

Whenever the pod self test is performed and the troubleshooter displays the message *POD SELF TEST 6809 FAIL xx* where *xx* equals 01, 02, or 03, the pod failed the standard self test. Refer to the section titled Recreating the Standard Self Test Routines.

Whenever the pod self test is performed and the troubleshooter displays the message *POD SELF TEST 6809 FAIL 00,* the pod may have failed either the standard self test or the enhanced self test. In most cases the enhanced self test, which is more thorough, will detect the failure. (This enhanced self test is transparent to the operator and to the troubleshooter; the troubleshooter does not know it is being performed.)

To find out if the pod failed the enhanced self test or the standard self test, perform a read operation at F0000. (F0000 is not an address within the normal address space of the 6809, but is a special address within the pod that contains information pertaining to the result of the enhanced self test). If the resulting message is *READ @ F0000 = FF OK,* then the pod passed the enhanced self test, which implies that any reported pod self test failures were caused by the failure of the standard self test; refer to the section titled Recreating the Standard Self Test Routines.

If the message resulting from a read operation at F0000 is anything other than *READ @ F0000 = FF OK,* then the pod failed the enhanced self test. Proceed to the next section titled Enhanced Self Test for appropriate troubleshooting procedures.

6-6. The Enhanced Self Test

The enhanced self test consists of 10 self test routines that are performed on the pod circuitry. The test routines are listed and described in Table 6-2.

Whenever the pod fails the enhanced self test, the pod causes the troubleshooter to display the message *POD SELF TEST 6809 FAIL 00.* To confirm that the enhanced self test failed, perform a read operation at address F0000. If the troubleshooter displays any message other than *READ @ F0000 = FF OK,* then the pod failed the enhanced self test. (In many cases an error message will be displayed.) Press the MORE key to find out more information about the error that was detected. Then press the CONT key to find out which enhanced self test routine failed.

NOTE

When investigating enhanced self test errors, be sure to set the active force line, control error, address error, and data error setup messages to YES.

For example, assume the pod self test is performed and the troubleshooter displays the message *POD SELF TEST 6809 FAIL 00.* The following sequence of keystrokes provides information about the pod failure detected:

PRESS	DISPLAY	COMMENT
READ F0000 ENTER	*DATA ERR @ F0000-LOOP?*	Enhanced self test detected data error.
MORE	*DATA BITS 10-LOOP?*	Data bit 4 is incorrect. 10 corresponds to bit 4).
CONT	*READ @ F0000 = 08 FAIL*	Failure occurred during test routine 8.

NOTE

If you press the LOOP key or the YES key in response to the LOOP? prompts which accompany the preceding messages, the troubleshooter loops on the READ @ F0000; the troubleshooter does not loop on the test routine in which the failure occurred.

The previous sequence of messages indicates that data bit 4 was found to be in error while the pod was performing test routine 8 (as described in Table 6-2). Note that although a data error usually implies a drivability error, when encountered with the enhanced self test a data error indicates only that incorrect data was received.

Here is another example of error information that may be obtained from failure of the enhanced self test:

PRESS	DISPLAY
READ F0000 ENTER	*ACTIVE FORCE LINE @ F0000-LOOP?*
MORE	*STS BITS 0000 0001-LOOP?*
CONT	*READ @ F0000 = 03 FAIL*

The previous sequence of messages indicates that status bit 0 ($\overline{\text{NMI}}$) was found to be in the incorrect state while the pod was performing test routine 3. Note that the *ACTIVE FORCE LINE* message, when encountered with the enhanced self test, may apply to any of the status lines, not just forcing the forcing lines.

6-7. Preparation for Troubleshooting a Defective Pod

Prepare to troubleshoot your defective pod as follows:

1. Refer to the later section titled Disassembly and disassemble the pod. 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 6-2. Enhanced Pod Self Test Failure Codes

<table>
<tr><th>FAILURE CODE/
TEST ROUTINE</th><th>OPERATIONS
PERFORMED BY POD</th><th>ACTIONS OPERATOR MAY TAKE
TO RECREATE TEST ROUTINE</th></tr>
<tr><td>01</td><td>Check internal pod RAM for read/writeability.</td><td>Operator cannot duplicate test.</td></tr>
<tr><td>02</td><td>Check internal pod ROM.</td><td>Operator cannot duplicate test.</td></tr>
<tr><td>03</td><td>READ @ A55A (with lines BA and BS low). Expected data is A5. Status and control lines are checked according to the list at right.</td><td>Perform looping READ @ A55A*. The data returned should be A5. Use probe in address sync to check latches U9 and U10 for these values:
<table>
<tr><th></th><th colspan="2">6809 MODE</th><th colspan="2">6809E MODE</th></tr>
<tr><th>PIN #</th><th>NAME</th><th>LEVEL</th><th>NAME</th><th>LEVEL</th></tr>
<tr><td>U9-3</td><td>$\overline{\text{NMI}}$</td><td>0</td><td>$\overline{\text{NMI}}$</td><td>0</td></tr>
<tr><td>U9-4</td><td>$\overline{\text{IRQ}}$</td><td>0</td><td>$\overline{\text{IRQ}}$</td><td>0</td></tr>
<tr><td>U9-7</td><td>$\overline{\text{FIRQ}}$</td><td>0</td><td>$\overline{\text{FIRQ}}$</td><td>1</td></tr>
<tr><td>U9-8</td><td>$\overline{\text{HALT}}$</td><td>0</td><td>$\overline{\text{HALT}}$</td><td>0</td></tr>
<tr><td>U9-13</td><td>$\overline{\text{RES}}$</td><td>1</td><td>$\overline{\text{RES}}$</td><td>1</td></tr>
<tr><td>U9-14</td><td>MR</td><td>0</td><td>AVMA</td><td>1</td></tr>
<tr><td>U9-17</td><td>$\overline{\text{DMA}}$</td><td>0</td><td>BUSY</td><td>0</td></tr>
<tr><td>U10-3</td><td>E</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U10-4</td><td>Q</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U10-7</td><td>BS</td><td>0</td><td>BS</td><td>0</td></tr>
<tr><td>U10-8</td><td>BA</td><td>0</td><td>BA</td><td>0</td></tr>
<tr><td>U10-13</td><td>R/$\overline{\text{W}}$</td><td>1</td><td>R/$\overline{\text{W}}$</td><td>1</td></tr>
<tr><td>U10-17</td><td>—</td><td>—</td><td>TSC</td><td>1</td></tr>
<tr><td>U10-18</td><td>—</td><td>—</td><td>LIC</td><td>1</td></tr>
</table></td></tr>
<tr><td>04</td><td>READ @ 5AA5 (with lines BA and BS low). Expected data is 5A. No status or control lines are checked during this test.</td><td>Perform a looping READ @ 5AA5*. The data returned should be 5A.</td></tr>
<tr><td>05</td><td>READ @ FFFF (with lines BA and BS low). This test is used in 6809E mode only; it tests several status and control lines.</td><td>Only applies to 6809E mode. Perform a looping WRITE @ E0000 = 0D*. Writing to this special address instructs the pod to perform a read at FFFF, but with a modified internal timer count which causes the pod to latch control lines at times other than at the end of an instruction cycle. Examine the following four lines with the probe in address sync:
<table>
<tr><th>PIN #</th><th>NAME</th><th>LEVEL</th></tr>
<tr><td>U9-7</td><td>$\overline{\text{FIRQ}}$</td><td>0</td></tr>
<tr><td>U9-14</td><td>AVMA</td><td>0</td></tr>
<tr><td>U10-17</td><td>TSC</td><td>0</td></tr>
<tr><td>U10-18</td><td>LIC</td><td>0</td></tr>
</table></td></tr>
</table>

Table 6-2. Enhanced Pod Self Test Failure Codes (cont)

<table>
<tr><th>FAILURE CODE/
TEST ROUTINE</th><th>OPERATIONS
PERFORMED BY POD</th><th>ACTIONS OPERATOR MAY TAKE
TO RECREATE TEST ROUTINE</th></tr>
<tr><td>06</td><td>READ @ FFFF (with lines BA and BS low). This test is used in 6809E mode only; it tests lines $\overline{IRQ}$ and $\overline{BUSY}$.</td><td>Only applies to 6809E mode. Perform a looping WRITE @ E0000 = 03.* This instructs the pod to perform a read at FFFF with a modified internal timer count (see preceding description of code 05). Examine the following two lines with the probe in address sync:
<table>
<tr><th>PIN #</th><th>NAME</th><th>LEVEL</th></tr>
<tr><td>U9-4
U9-17</td><td>$\overline{IRQ}$
$\overline{BUSY}$</td><td>0
0</td></tr>
</table></td></tr>
<tr><td>07</td><td>READ @ A55A (with line BA low and line BS high). Expected data is 5A. Lines BA and BS are checked.</td><td>Disable reporting of control line errors by setting SET-TRAP CTL ERROR? to NO (troubleshooter Setup function).

Perform a WRITE @ E0001 = 04. Writing to this special address sets line BS high so that the self test socket is configured to connect low order address lines to the data lines.

Next, perform a looping READ @ A55A*. The expected data is 5A. You can check the BA and BS lines on latches U9 and U10 (with the probe in address sync):
<table>
<tr><th></th><th colspan="2">6809 MODE</th><th colspan="2">6809E MODE</th></tr>
<tr><th>PIN #</th><th>NAME</th><th>LEVEL</th><th>NAME</th><th>LEVEL</th></tr>
<tr><td>U10-7
U10-8</td><td>BS
BA</td><td>1
0</td><td>BS
BA</td><td>1
0</td></tr>
</table>
After testing, set the lines low by performing a WRITE = E0001 = 0 or performing a pod reset. Remember to select SET-TRAP CTL ERROR? YES.</td></tr>
<tr><td>08</td><td>READ @ 5AA5 (with line BA low and line BS high). Expected data is A5.</td><td>Perform the same steps as listed for code 07 above, except perform a looping READ @ 5AA5*. The expected data is A5. After testing, set the lines low by performing a WRITE @ E0001 = 0 or by performing a pod reset. Remember to select SET-TRAP CTL ERROR? YES.</td></tr>
</table>

Table 6-2. Enhanced Pod Self Test Failure Codes (cont)

<table>
<tr><th>FAILURE CODE/
TEST ROUTINE</th><th>OPERATIONS
PERFORMED BY POD</th><th>ACTIONS OPERATOR MAY TAKE
TO RECREATE TEST ROUTINE</th></tr>
<tr><td>09</td><td>WRITE @ F00F = A5 (with lines BA and BS high). Status and control lines are checked as listed at right.</td><td>Perform the same steps as listed for code 07, except perform a single WRITE @ E0001 = 0C. Then perform a looping WRITE @ F00F = A5*. With the probe in address sync, examine latches U9 and U10 for the following data:

<table>
<tr><th></th><th colspan="2">6809 MODE</th><th colspan="2">6809E MODE</th></tr>
<tr><th>PIN #</th><th>NAME</th><th>LEVEL</th><th>NAME</th><th>LEVEL</th></tr>
<tr><td>U9-3</td><td>$\overline{\text{NMI}}$</td><td>0</td><td>—</td><td>—</td></tr>
<tr><td>U9-4</td><td>$\overline{\text{IRQ}}$</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U9-7</td><td>$\overline{\text{FIRQ}}$</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U9-8</td><td>$\overline{\text{HALT}}$</td><td>1</td><td>$\overline{\text{HALT}}$</td><td>1</td></tr>
<tr><td>U9-13</td><td>$\overline{\text{RESET}}$</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U9-14</td><td>$\overline{\text{MR}}$</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U9-17</td><td>$\overline{\text{DMA}}$</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>U10-7</td><td>BS</td><td>1</td><td>BS</td><td>1</td></tr>
<tr><td>U10-8</td><td>BA</td><td>1</td><td>BA</td><td>1</td></tr>
<tr><td>U10-13</td><td>R/$\overline{\text{W}}$</td><td>0</td><td>R/$\overline{\text{W}}$</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>10</td><td>WRITE @ F00F = 5A (with lines BA and BS high). The $\overline{\text{HALT}}$ line is checked.</td><td>Only applies to the 6809E mode. Perform the same steps as listed for code 07, except perform a looping WRITE @ F00F = 5A*.

With the probe in address sync, check U9-8 ($\overline{\text{HALT}}$) for a 0 level.

After testing, set the lines low by performing a WRITE @ E0001 = 0 or performing a pod reset. Remember to select SET-TRAP CTL ERROR? YES.</td></tr>
<tr><td>FF</td><td>No problem was found with any of the enhanced self test routines.</td><td>No action required.</td></tr>
<tr><td colspan="3">*When viewing a looping read or write operation with a synchronized scope, use the Quick Looping read or write feature described in Section 4 to obtain a brighter signal trace on the scope.</td></tr>
</table>

3. Connect the pod to the troubleshooter, and insert the ribbon cable plug into the self test socket as shown in Figure 6-1. Lock the self test socket by turning the lock screw at the end of the socket clockwise.

4. Set the pod switches beside the self test socket to the positions shown in Figure 6-1. Note that the position of switch 4 that is shown in Figure 6-1 is opposite to what is shown on the pod decal beside the switches.

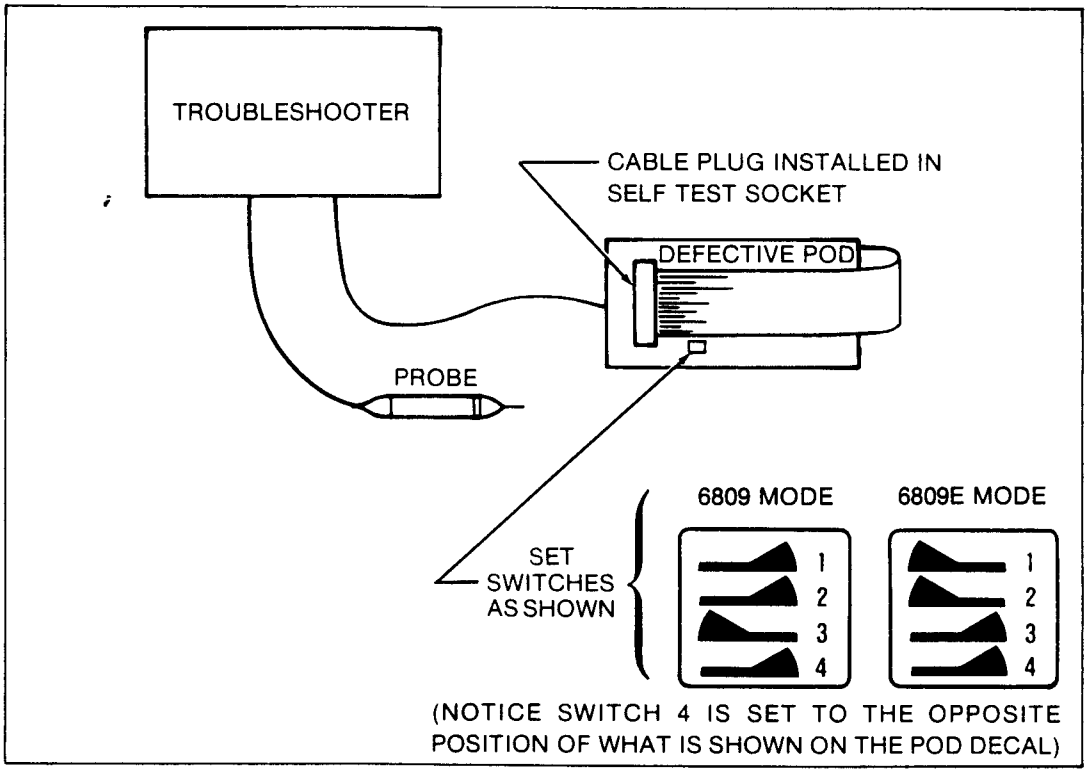


Figure 6-1. Troubleshooting a Defective Pod

5. Press the SETUP key on the troubleshooter and set the following conditions:

SET-TRAP BAD POWER SUPPLY? YES

SET-TRAP ILLEGAL ADDRESS? NO

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

In addition, set the following conditions for the 6809 mode:

SET-ENABLE HALT? NO

SET-ENABLE DMA? NO

SET-ENABLE MR? NO

Set the following conditions for the 6809E mode:

SET-ENABLE HALT? NO

SET-ENABLE TSC? NO

When the pod and the troubleshooter are connected in this configuration (with switch 4 in the position indicated in Figure 6-1), 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. Remember that in order to perform the pod self test, you must set switch 4 to the position indicated on the pod decal.

6-8. Recreating the Enhanced Self Test Routines

Notice that the information provided in Table 6-2 not only describes the operations the pod performs for each test routine, but the information also describes actions the operator may take to recreate the test routines. By recreating the test routine in which the pod failure was detected, you may then use a synchronized probe or an oscilloscope to trace the cause of the failure.

Consider the first example in the preceding section where the pod detected that data bit 4 had the incorrect value during the execution of test routine 8. To troubleshoot this pod, first make sure you have followed the steps listed in the previous section titled Preparation for Troubleshooting a Defective Pod (which includes disabling the pod enable lines and setting switch 4 in the correct position for troubleshooting as shown in Figure 6-1). Then press the following sequence of keystrokes to recreate test routine 8 (as described in Table 6-2):

PRESS	DISPLAY	COMMENT
WRITE E0001 ENTER 4 ENTER	*WRITE @ E0001 = 4 OK*	Sets line BS high.
READ 5AA5 ENTER LOOP	*READ @ 5AA5 = B5 OK*	Wrong answer, expected A5. Data bit 4 is high in error.

Since you have confirmed that data bit 4 is in error, set the probe to the address synchronization mode. While the troubleshooter continues to loop on the read at address 5AA5, refer to the schematic diagram in Section 8 and trace the circuit logic to find where the data is in error. In this example, for instance, an open line (A4) on the ribbon cable causes data bit 4 to appear to be stuck at the high logic level.

By following the procedures listed in Table 6-2, other test routines may be recreated in the same way.

6-9. Recreating the Standard Self Test Routines

Most of the problems that occur with pod circuitry will be detected by the enhanced pod self test. However, there are a few problems that may be detected only by the standard pod self test. If the pod fails the standard pod self test, the recommended procedure is the same as the recommended procedure for failure of the enhanced self test. You may recreate the standard self test routine that detected the failure and use that routine as the starting point for subsequent troubleshooting.

The failure codes for the standard self test are listed in Table 6-3 along with a description of the actions that you may take to recreate the self test routine. Use similar troubleshooting techniques as previously described for the recreation of the enhanced self test routines.

6-10. TROUBLESHOOTING AN INOPERATIVE POD

6-11. Introduction

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

Table 6-3. Standard Pod Self Test Failure Codes

FAILURE CODE/ TEST ROUTINE	OPERATIONS PERFORMED BY TROUBLESHOOTER	ACTIONS OPERATOR MAY TAKE TO RECREATE TEST ROUTINE
00*	Reset pod and READ @0FF0 (data expected by troubleshooter is 0F).	Perform READ @ 0FF0 and check for data 0F. If a power fail error message occurs, check the power detection circuits. If a control or address drivability error occurs, press the MORE key to examine the second line of the error message and see which line is not drivable. Use a synchronized probe or scope to trace the problem. For a brighter scope display, use the Quick Looping Read function described in Section 4.
01	WRITE @ 0FF0 = 0F	Perform WRITE @ 0FF0 = 0F. Follow the same procedure as described for test routine 00 to locate the problem.
02	Test control line driveability high or low.	Perform a Bus Test and observe the error message. If the error involves a writeable control line, perform a looping WRITE @ CTL to help locate the problem. If the error involves a clock line, use a scope to help locate the problem. If the problem involves one of the 6809E control lines LIC, AVMA, or BUSY, try replacing the microprocessor since these lines are connected directly to the UUT from the microprocessor.
03	Send command to enable all pod enable lines and verify that a pod timeout occurs. (This pod timeout is transparent to the operator.)	Enable each pod enable line singly and verify that the active logic level present at the self test socket is also present at the corresponding pin of the microprocessor.

*If the self test message is *POD SELF TEST 6809 FAIL 00,* the enhanced self test may have failed. To check, perform a read operation at F0000:

If the resulting message is *READ@F0000 = FF OK,* then the pod passed the enhanced self test, but failed the standard self test as described in this table.

If the resulting message is anything other than *READ@F0000 = FF OK,* the pod failed the enhanced self test; refer to the previous section titled The Enhanced Self Test.

If you correct a problem while using the procedures provided in this section, try the pod self test again. (Before performing the pod self test, remember to set switch 4 in the position shown on the pod decal.) 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 6809 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. (The Processor Section and the Timing Section are described in the theory of operation in Section 5.)

6-12. Preparation for Troubleshooting an Inoperative Pod

Before beginning to troubleshoot an inoperative pod, perform the following steps:

1. Recheck the position of the switches beside the pod microprocessor socket and make sure they are in the position shown on the pod decal.

2. Ensure that the proper microprocessor has been installed in the pod microprocessor socket, and that pin 1 is in the correct location.

3. If the message *POD TIMEOUT - ATTEMPTING RESET* occurred during the pod self test, ensure that switch 4 is set in the position shown in the pod decal and try the pod self test again. If the same message is again displayed, check for shorted connections to the crystal in the 6809 mode, or shorted connections to the clock in the 6809E mode.

4. If the pod passes the self test, but the message *POD TIMEOUT - ATTEMPTING RESET* is displayed when the pod is connected to a UUT and an operation is attempted, disable all the pod enable lines and the forcing line reporting. Then try the operation with the UUT again.

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. If the pod is still inoperative after performing the preceeding steps, then prepare to troubleshoot the pod by performing the following steps:

1. Refer to the later section titled Disassembly and disassemble the pod, 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.

3. Remove the pod microprocessor from its socket.

4. To provide a clock signal for the inoperative pod, insert the inoperative pod ribbon cable into the inoperative pod self test socket. (Make sure the clock is working properly.) An alternative source for a clock signal is a known good UUT).

5. 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 (2A) power supply and a -5V dc (200 mA) power supply to the inoperative pod as shown in Figure 6-2. An easy place to make the power connections is at the connector that

usually connects the cable to the troubleshooter. Connect pins 2 and 15 to +5V, pin 21 to -5V, and pin 25 to ground.

6. 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 in the microprocessor socket of the inoperative pod.

CAUTION

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

CAUTION

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

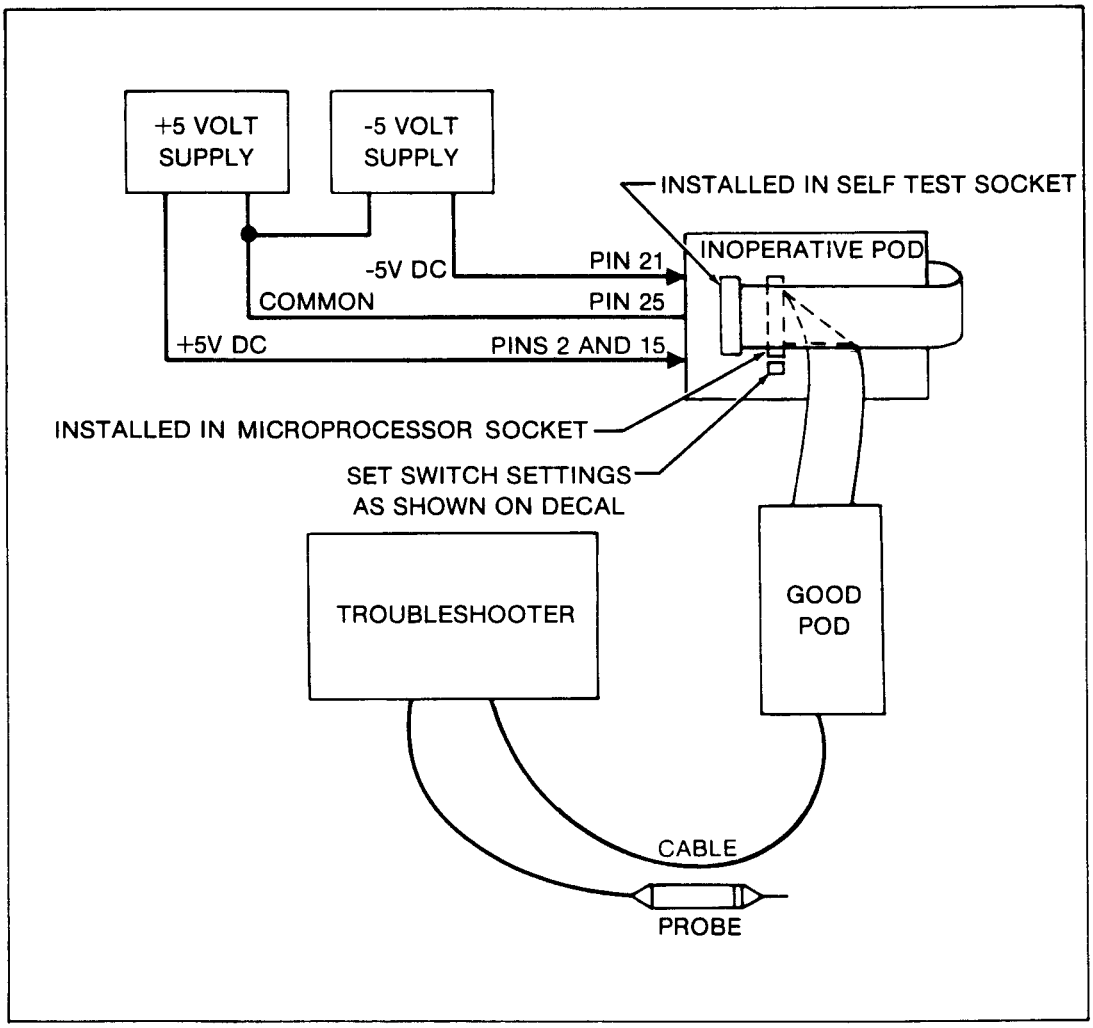


Figure 6-2. Troubleshooting an Inoperative Pod

6-13. Procedure for Troubleshooting an Inoperative Pod

Use the following steps as a guide for troubleshooting an inoperative 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

When performing looping read or write operations with a synchronized scope 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. Select the troubleshooter Setup function and disable all the pod enable lines for the good pod.

2. Check the pod mode control lines, 09E, and $\overline{09E}$. Line 09E is found at U18-6. Line 09E should be high for the 6809E mode, and low for a 6809 pod. Line $\overline{09E}$ is found at U18-5, and should be the complement of 09E.

3. Check the LOPWR signal at pin 58 of the pcb board-to-board connector. Assuming proper UUT voltages, this line should be low.

4. Verify the presence of clean square waves at the Q and E clocks at the microprocessor socket, pins 34 and 35. If the inoperative pod is in the 6809 mode, check the analog switch, U15, and the transistor array U14; both components are located on the processor pcb. The inputs to the oscillator, if a crystal is used, should be sine waves about 0.5 to 1V peak to peak at the frequency of the crystal. These signals can be measured with a high impedance scope probe at U15 pins 1 and 13.

5. Reset the pod by momentarily shorting pins 22 and 23 of the shielded cable connector located on the processor pcb.

6. Perform a Bus Test.

7. Perform a RAM Short and RAM Long Test. The RAM addresses are listed in Table 6-4.

8. Perform a ROM Test. The ROM addresses for each mode (6809 and 6809E) are listed in Table 6-4. For the 6809 mode, the expected signature for the ROM Test can be read at ROM locations F000 (most significant word) and F001 (least significant word); perform the ROM Test over the address space F002 to F7FF. For the 6809E mode, the expected signature for the ROM Test can be read at ROM locations F800 (most significant word) and F801 (least significant word); perform the ROM Test over the address space F802 to FFFF.

NOTE

Steps 9 and 10 are for checking the output operation of I/O port A. These steps are intended to be used only if the inoperative pod is supplied with power from separate power supplies. If the pod is supplied with power by a second troubleshooter, I/O port A may be overdriven by the outputs from the troubleshooter.

9. Check the output operation of I/O port A (contained in U1) as follows:

   a. Perform the operation *WRITE @ 81 = FF*. This operation writes to the port A direction register and sets all the lines of I/O port A (PA0 through PA7) as outputs.

Table 6-4. RAM, ROM, I/O, and Device Addresses

ADDRESSABLE DEVICE	ADDRESS (HEX)
RAM	0000 - 007F
ROM*	
6809 pod	F002 - F7FF
6809E pod	F802 - FFFF
I/O	
Port A Direction Register	0081
Port A Data Register	0080
Port B Direction Register	0083
Port B Data Register	0082
Interval Timer Load/Start	2xxx (write only)
Latch Addresses	
Data Latch (U6)	1xxx
Address Latch, low byte (U7)	4xxx
Address Latch, high byte (U8)	6xxx
Status Latch (U9)	3xxx
Control Latch (U10)	5xxx

x = any hex value

*For the 6809 mode, the ROM signature for ROM Test is stored in addresses F000 (most significant word) and F001 (least significant word).

For the 6809E mode, the ROM signature for ROM Test is stored in addresses F800 (most significant word) and F801 (least significant word).

b. Perform the operation *WRITE @ 80 = FF*. This operation writes to the I/O port A data register and sets all bits high.

c. Check the I/O port A lines with the probe or scope to confirm that all the levels are logic high.

d. Repeat step b with 00 as the data.

e. Repeat step c, checking for all logic low levels.

10. Check the input operation of I/O port A (contained in U1) as follows:

a. Perform the operation *WRITE @ 81 = 00*. This operation writes to the port A direction register and sets all the lines of I/O port A (PA0 through PA7) as inputs.

b. Perform a read operation at address 80, the address of the I/O port A data register, while sequentially applying the probe (with high stimulus pulses selected) to each of the I/O port A input pins (pins 8 through 15 of U1). Observe the troubleshooter display and check whether each input is high.

NOTE

The following steps may be performed on an inoperative pod that is supplied with power either by separate power supplies or by a second troubleshooter. If a second troubleshooter is used and the message POD TIMEOUT - ATTEMPTING RESET is displayed on the second troubleshooter, the inoperative pod is continually being reset. Press the STOP key on the second troubleshooter to prevent repetitive pod resets.

11. Check the output operation of I/O port B (PB0 through PB7) by repeating step 9. Use address 83 for the I/O port B direction register, and address 1082 for the I/O port B data register.

12. Check the input operation of I/O port B, line PB7 ($\overline{\text{MAINSTAT}}$), by repeating step 10. Use address 83 for the Port B direction register, and use data 00 to set line PB7 as an input. Perform a looping read at address 82 and apply +5V (dc) to U1, pin 16.

13. Check the operation of the interval timer by performing the operation *WRITE @ 2000 = 0B*. Verify that the UUTON output, U6-9, goes high in response to the write operation. Verify that both ATSC, U20-8, and DTSC, U12-8 go low during this same operation.

14. Perform a looping *WRITE @ 2000 = 0B* and verify that UUTON goes high four clock cycles after the write operation, and stays high for one clock period. Refer to the theory of operation in Section 5 for a timing diagram of the interval timer and UUTON generator. Use the trigger output on the rear panel of the troubleshooter to synchronize the scope for this operation.

15. Check for the presence of the $\overline{\text{SYNC}}$ signal at pin 10 of the cable connector to the troubleshooter.

16. Check the address decoder (U11 on the interface pcb) by performing read operations at addresses 0000, 1000, 2000, 3000, 4000, 5000, 6000, and 7000. Verify that the corresponding decoder output goes low when addressed.

17. If repairs have been made to the inoperative pod as a result of the preceeding steps, try the pod self test again. (Remember to place switch 4 in the position indicated on the pod decal.) If the message displayed is *POD SELF TEST 6809 FAIL xx*, refer to the previous section titled Troubleshooting a Defective Pod.

6-14. EXTENDED TROUBLESHOOTING PROCEDURES

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

6-15. Cable Lines

The enhanced self test checks every line in the cable to ensure that it may be driven both high and low except for pin 39 in the 6809 mode, which is the undriven crystal input. Remember that the self-test socket provides a driven clock input, rather than a crystal, so that this line is not checked. 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 outputs.

6-16. Multiple Faults

The pod self test routines that check the cable assume that there may be only a single fault. If there are multiple faults, the self test might not report the error. For example, during the self test of the 6809 pod, line $\overline{NMI}$ is driven by the Q clock output. During the subsequent tests, $\overline{NMI}$ is checked to confirm that it is always low, which it is during the falling edge of the signal E. The Q drivability detector is checked to confirm that it is always high, which indicates that Q is drivable both high and low. However, if two faults exist, such as the $\overline{NMI}$ line stuck low and the Q line in the cable open, then the self test would report no problem.

One good way to troubleshoot problems like this is to set the switches so that the pod is in the other mode (the 6809E mode for the preceeding example) and again perform the pod self test. Performing a self test in the 6809E mode would discover the error in the preceeding example.

6-17. Pod Enable Lines

The circuitry for enabling the various forcing lines is checked by the normal pod self test (corresponding to the failure code 03). During this test routine, all the forcing lines are simultaneously enabled. If the enableable line circuitry seems to be functioning improperly when the pod is connected to a UUT, try selectively enabling the lines with the pod cable inserted into the self test socket, but with the pod self test disabled. (The pod self test is disabled when switch 4 is placed in the position indicated in Figure 6-1.) If a line is found that does not cause the pod to timeout, then the logic controlling the enabling of that line should be examined. Remember that in all cases, the PIA, U1, generates the command to enable or disable these lines. Start with U1 and follow the circuitry through to the microprocessor.

6-18. Timing Problems

These problems are usually caused by components that are still functioning, but are not functioning within the allowable specifications. The best way to check this problem is to look at suspected signals using an oscilloscope synchronized to valid addresses. Look for slow rise or fall times or signals driven to marginal logic levels. If the part is too slow, it might fail in the UUT, but pass the pod self test because the pod clock rate is slightly slower. The clock rate at the self test socket is approximately 1 MHz.

6-19. Noise Problems

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

6-20. DISASSEMBLY

To gain access to the two pcb assemblies in the pod, perform the following steps:

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

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

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

NOTE

When the shield is removed, all the components are exposed. It may not be necessary to separate the two pcb assemblies while troubleshooting except to replace components. If the two pcb assemblies are not separated, be sure that they are securely screwed together to ensure proper electrical connection.

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

Section 7
List of Replaceable Parts

7-1. INTRODUCTION

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

Parts lists include the following information:

1. Reference Designation.

2. Description of Each Part.

3. Fluke Stock Number.

4. Federal Supply Code for Manufacturers (see the 9000 Series Troubleshooter Service Manual for Code-to-Name list).

5. Manufacturer's Part Number.

6. Total Quantity of Components Per Assembly.

7. Recommended quantity: This entry indicates the recommended number of spare parts necessary to support one to five instruments for a period of 2 years. This list presumes an availability of common electronic parts at the maintenance site. For maintenance for 1 year or more at an isolated site, it is recommended that at least one of each assembly in the instrument be stocked.

7-2. HOW TO OBTAIN PARTS

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

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

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

1. Quantity.

2. Fluke Stock Number.

3. Description.

4. Reference Designation.

5. Printed Circuit Board Part Number and Revision Letter.

6. Instrument Model and Serial Number.

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

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

CAUTION

Indicated devices are subject to damage by static discharge.

7-3. MANUAL CHANGE AND BACKDATING INFORMATION

Table 7-4 contains information necessary to backdate the manual to conform with earlier pcb configurations. To identify the configuration of the pcbs 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-6809 Final Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
	FINAL ASSEMBLY, 9000A-6809 POD FIGURE 7-1 (9000A-6809-5001)						
A23Ⓧ	PROCESSOR PCB ASSEMBLY	647875	89536	647875	1		
A24Ⓧ	INTERFACE PCB ASSEMBLY	647891	89536	647891	1		
H1	SCREW, SEMS, 4-40 X 1/4	185918	89536	185918	4		
H2	SCREW, PHP, 4-40 X 5/8	145813	89536	145813	1		
H3	SCREW, RHP, 4-40 X 3/4	115063	89536	115063	4		
H4	WASHER, #4 LOCK	110403	89536	110403	1		
MP1	ACTUATOR	582916	89536	582916	1		
MP2	COVER, SLIDE	653139	89536	653139	1		
MP3	DECAL, POD	680710	89536	680710	1		
MP4	DECAL, SPECIFICATION	680728	89536	680728	1		
MP5	DECAL, SWITCHING	680413	89536	680413	1		
MP6	DECAL, WARNING	659813	89536	659813	1		
MP7	LABEL, STATIC CAUTION	605808	89536	605808	1		
MP8	LABEL, UUT CAUTION	634030	89536	634030	1		
MP9	SHELL, BOTTOM	648881	89536	648881	1		
MP10	SHELL, TOP	653113	89536	653113	1		
MP11	SHIELD	659771	89536	659771	1		
MP12	SOCKET, DIP 40-PIN (SOCKET SAVER)	614297	89536	614297	2		
MP13	SPACER, HEX	187575	89536	187575	1		
MP14	ACCESSORY KIT (9000A-6809-POD) (NOT SHOWN)	607150	89536	607150	1		
U2	IC, EPROM (ON A23)	649111	89536	649111	1	1	
U26	IC, 8-BIT MICROPROCESSOR (ON A23)	647099	89536	647099	1	1	
W1	CABLE, POD (TO J1)	581819	89536	581819	1		
W2	CABLE, UUT (TO J2)	680595	89536	680595	1		

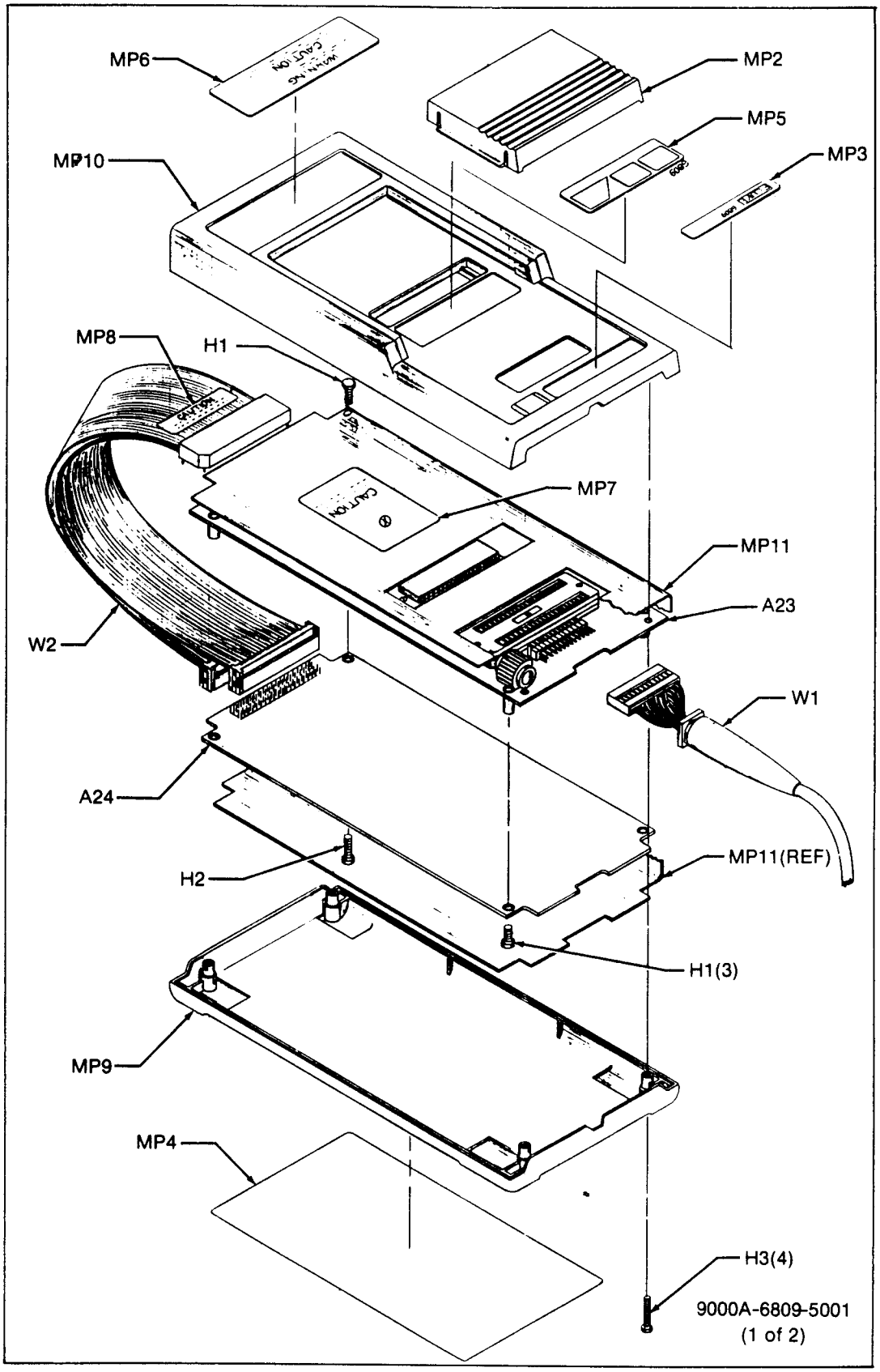
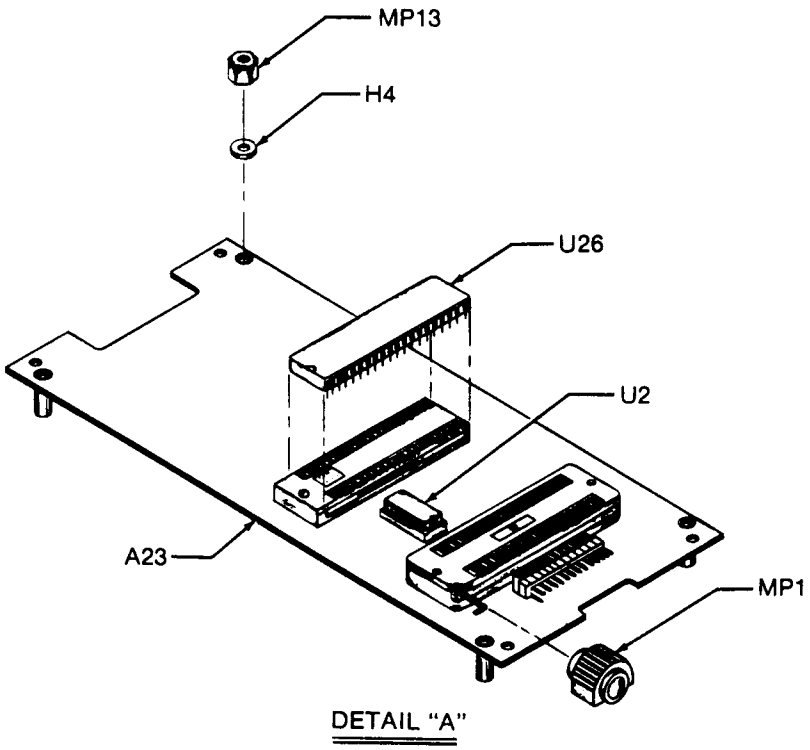


Figure 7-1. 9000A-6809 Final Assembly



9000A-6809-5001
(2 of 2)

Figure 7-1. 9000A-6809 Final Assembly (cont)

Table 7-2. A23 Processor PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A23Ⓧ	PROCESSOR PCB ASSEMBLY FIGURE 7-2 (9000A-6809-4071)	647875	89536	647875	REF		
C1	CAP, CER, 22 PF +/-5%, 100V	448449	80031	2222-638-10229	1		
C2-C15	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	14		
CR1	DIODE, SI, SMALL SIGNAL	313247	28484	HP5082-6264	1	1	
J1	CONNECTOR, RIGHT ANGLE	512590	89536	512590	1		
J2	CONNECTOR, 40-PIN	585133	89536	585133	1		
J3	SOCKET, MICROPROCESSOR U26	524124	19613	240-4846-00-1757	1		
MP1	STANDOFF, SWAGE	380329	89536	380329	5		
P1	CONNECTOR PINS	267500	00779	87022-1	60		
R1	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	4		
R2	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	2		
R3	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R4	RES, DEP. CAR, 240 +/-5%, 1/4W	376624	80031	CR251-4-5P240E	1		
R5	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R6	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	REF		
R7	RES, DEP. CAR, 100K +/-5%, 1/4W	348920	80031	CR251-4-5P100KT	1		
R8	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
S1	SWITCH, DIP, 4PST	408559	00779	435166-2	1		
TP1	CONNECTOR, TEST POINT	512889	02660	62395	2		
TP2	CONNECTOR, TEST POINT	512889	02660	62395	REF		
U1Ⓧ	IC, NMOS, RAM, I/O, TIMER	536417	55576	SYP6532A	1	1	
U2	See Final Assembly Table 7-1						
U3	IC, LSTTL, 4-BIT SYNCCRNS BIN CNTR	495598	01295	SN74LS163N	1	1	
U4	IC, TTL, TRI-2BIT-1BIT MX/DEMX	375808	02735	CD4053BE	1	1	
U5	IC, LSTTL, DUAL&EDG TRG FF W/PRST&CLR	393124	01295	SN74LS74N	2	1	
U6	IC, STTL, DUAL&EDG TRG D FF W/PRST&CLR	418269	01295	SN74S74N	1	1	
U7	IC, LSTTL, DUAL&EDG TRG FF W/PRST&CLR	393124	01295	SN74LS74N	REF		
U8	IC, LSTTL, QUAD 2-INPUT AND GATE	393066	01295	SN74LS08N	3	1	
U9	IC, LSTTL, QUAD 2-INPUT AND GATE	393066	01295	SN74LS08N	REF		
U10	IC, LSTTL, QUAD BFR GTS @ TRI-ST NOT	472746	01295	SN74LS125	3	1	
U11	IC, LSTTL, QUAD BFR GTS @ TRI-ST NOT	472746	01295	SN74LS125	REF		
U12	IC, LSTTL, QUAD 2-INPUT OR GATE	393108	01295	SN74LS32N	4	1	
U13	IC, LSTTL, QUAD 2-INPUT OR GATE	393108	01295	SN74LS32N	REF		
U14	IC, LSTTL, QUAD 2-INPUT OR GATE	393108	01295	SN74LS32N	REF		
U15	IC, LSTTL, HEX INVERTER	393058	01295	SN74LS04N	2	1	
U16	IC, LSTTL, QUAD 2-INPUT AND GATE	393066	01295	SN74LS08N	REF		
U17	IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	1	1	
U18	IC, LSTTL, HEX INVERTER	393058	01295	SN74LS04N	REF		
U19	IC, LSTTL, QUAD BFR GATES EA TRI-STATE	585273	01295	SN74LS126N	2	1	
U20	IC, STTL, 8-INPUT NAND GATE	407338	01295	SN74S30N	1	1	
U21	IC, LSTTL, DUAL EDG TRG JK FF W/PRST&CLR	414029	01295	SN74LS112N	1	1	
U22	IC, LSTTL, QUAD BFR GTS @ TRI-ST NOT	472746	01295	SN74LS125	REF		
U23	IC, STTL, CLOCK GENERATOR	586065	04713	MC6875L	1	1	
U24	IC, TTL, OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	2	1	
U25	IC, TTL, OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	REF		
U26	See Final Assembly Table 7-1	647099	89536	647099	1	1	
U27	IC, LSTTL, QUAD BFR GATES EA TRI-STATE	585273	01295	SN74LS126N	REF		
U28	IC, LSTTL, QUAD 2-INPUT OR GATE	393108	01295	SN74LS32N	REF		

Table 7-2. A23 Processor PCB Assembly (cont)

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
W2	CABLE, UUT	680611	89536	680611	1		
XU1	SOCKET, IC, 40-PIN	429282	09922	DILB40P-108	1		
XU2	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	1		
XU24	SOCKET, IC, 20-PIN	454421	01295	C932002	2		
XU25	SOCKET, IC, 20-PIN	454421	01295	C932002	REF		
XU28	SOCKET, IC, 14-PIN	276527	09922	DILB8P-108	1		
Z1	RESISTOR NETWORK, 4.7K	412916	89536	412916	1	1	
Z2	RESISTOR NETWORK, 4.7K	494690	89536	494690	1	1	

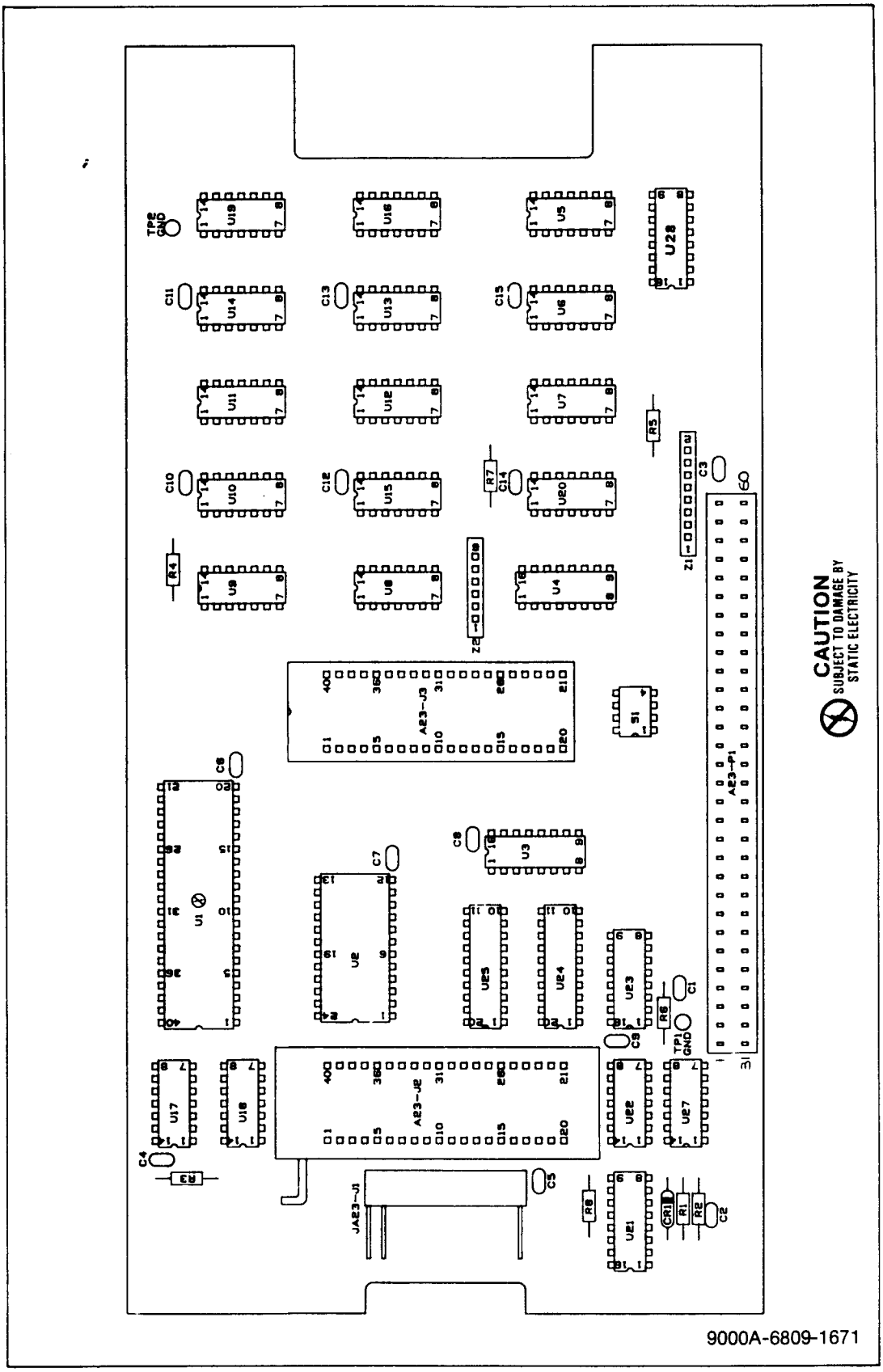


Figure 7-2. A23 Processor PCB Assembly

Table 7-3. A24 Interface PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A24Ⓐ	INTERFACE PCB ASSEMBLY FIGURE 7-3 (9000A-6809-4072)	647891	89536	647891	REF		
A1	HYBRID, PROTECTION, TESTED, 700	582189	89536	582189	1		
A2-A5	HYBRID, TESTED, 3K	582247	89536	582247	4		
C1-C16	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	27		
C17	CAP, CER, 47 PF +/-2%, 100V	512368	89536	512368	2		
C18-C21	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C22	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015KA1	4		
C23	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C24	CAP, CER, 0.01 UF +/-20%, 100V	407361	72982	8121-A100-W5R-103M	3		
C25	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015KA1	REF		
C26	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C27	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C28	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015KA1	REF		
C29	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015KA1	REF		
C30	CAP, CER, 0.01 UF +/-20%, 100V	407361	72982	8121-A100-W5R-103M	REF		
C31	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C32	CAP, CER, 0.01 UF +/-20%, 100V	407361	72982	8121-A100-W5R-103M	REF		
C33-C35	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C36	CAP, CER, 47 PF +/-2%, 100V	512368	89536	512368	REF		
CR1	DIODE, SI, HI-SPEED SWITCHING	203323	07910	1N4448	4	1	
CR2	DIODE, SI, SMALL SIGNAL	313247	28484	HP5082-6264	2	1	
CR3	DIODE, SI, HI-SPEED SWITCHING	203323	07910	1N4448	REF		
CR4	DIODE, SI, HI-SPEED SWITCHING	203323	07910	1N4448	REF		
CR5	DIODE, SI, HI-SPEED SWITCHING	203323	07910	1N4448	REF		
CR6	DIODE, SI, SMALL SIGNAL	313247	28484	HP5082-6264	REF		
J1	CONNECTOR, 60-PIN	602813	00779	86396-6	1		
J3	CONNECTOR, PIN	267500	00779	87022-1	40		
Q1	TRANSISTOR, PNP, HI SPEED SWITCHING	369629	07263	543576	1	1	
R1	RES, DEP. CAR, 200 +/-5%, 1/4W	441451	80031	CR251-4-5P200E	2		
R2	RES, DEP. CAR, 200 +/-5%, 1/4W	441451	80031	CR251-4-5P200E	REF		
R3	RES, DEP. CAR, 3K +/-5%, 1/4W	441527	80031	CR251-4-5P3K	2		
R4	RES, DEP. CAR, 820 +/-5%, 1/4W	442327	80031	CR251-4-5P820E	1		
R5	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	1		
R6	RES, MTL. FILM, 12.7K +/-1%, 1/8W	294918	91637	CMF551272F	1		
R7	RES, MTL. FILM, 2.61K +/-1%, 1/8W	289983	91637	CMF552611F	1		
R8	RES, DEP. CAR, 470K +/-5%, 1/4W	342634	80031	CR251-4-5P470K	1		
R9	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	4		
R10	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R11	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R12	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R13	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	5		
R14	RES, DEP. CAR, 3K +/-5%, 1/4W	441527	80031	CR251-4-5P3K	REF		
R15	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R16	RES, DEP. CAR, 220 +/-5%, 1/4W	342626	80031	CR251-4-5P220E	1		
R17	RES, DEP. CAR, 330 +/-5%, 1/4W	368720	80031	CR251-4-5P330E	1		
R18	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	2		
R19	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
R20	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		

Table 7-3. A24 Interface PCB Assembly (cont)

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
R21	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	REF		
R22	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	REF		
TP1	CONNECTOR, TEST POINT	512889	02660	62395	2		
TP2	CONNECTOR, TEST POINT	512889	02660	62395	REF		
U1	IC, ALSTTL, OCTAL BUS XCVR W/3-STATE	647214	01295	SN74ALS245N	1	1	
U2-U5	IC, TTL, OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	4	1	
U6-U10Ø	IC, CMOS, OCATAL LTCH HI SPEED EQ	585364	36665	74SC374A	5	1	
U11	IC, LSTTL, 3-8 BIT DECODER W/ENABLE	407585	01295	SN74LS138N	1	1	
U12	IC, PROTECTOR	585992	89536	585992	1	1	
U13	IC, LSTTL, HEX INVERTER	393058	01295	SN74LS04N	1	1	
U14	IC, LINEAR, 5-TRANSISTOR ARRAY	248906	12040	LM3046N	1	1	
U15	IC, TTL, TRI-2BIT-1BIT MX/DEMX	375808	02735	CD4053BE	1	1	
U16	IC, LINEAR, OP-AMP	472894	12040	LM311N	1	1	
U17	IC, LINEAR, QUAD COMPARATOR	387233	12040	LM339N	1	1	
VR1	IC, LINEAR, LO-VOLT REF (SELECT)	452771	89536	452771	1	1	
W1	WIRE, JUMPER, #22AWG	529701	89536	529701	2		
W2	WIRE, JUMPER, #22AWG	529701	89536	529701	REF		
XU1-XU10	SOCKET, IC, 20-PIN	454421	01295	C932002	10		
XU12	SOCKET, IC, 18-PIN	418228	91506	318-AG39D	1		
XVR1	INSULATOR	175125	89536	175125	1		
Z1	RESISTOR NETWORK	583476	89536	583476	1	1	

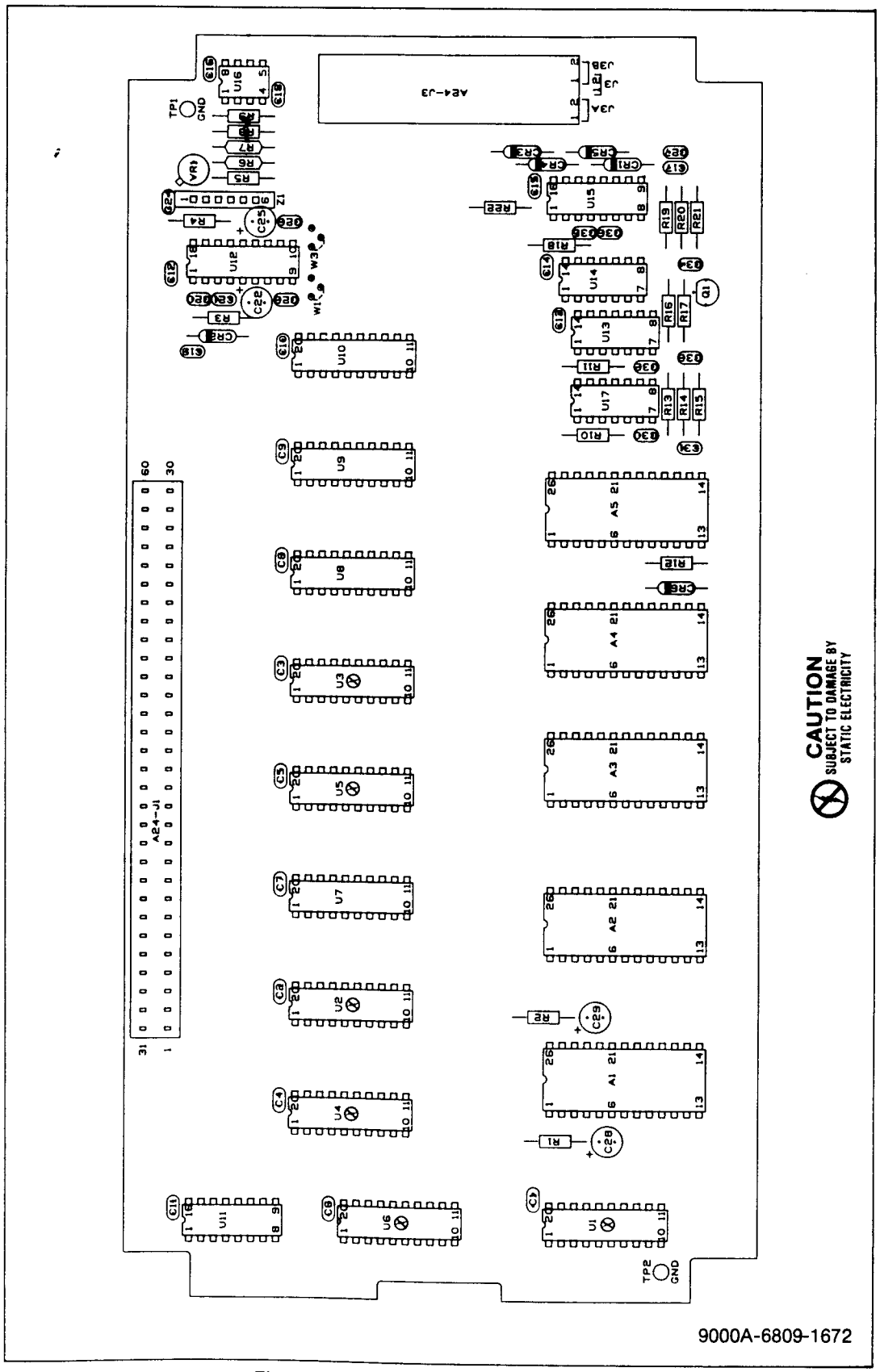


Figure 7-3. A24 Interface PCB Assembly

Table 7-4. Manual Status and Backdating Information

Ref Or Option No.	Assembly Name	Fluke Part No.	* To adapt manual to earlier rev configurations perform changes in desending order (by no.), ending with change under desired rev letter																					
			–	A	B	C	D	E	F	G	H	J	K	L	M	N	P							
A23	Processor PCB Assembly	647875	●	●	X																			
A24	Interface PCB Assembly	647891	●	●	X																			

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

Section 8
Schematic Diagrams

TABLE OF CONTENTS

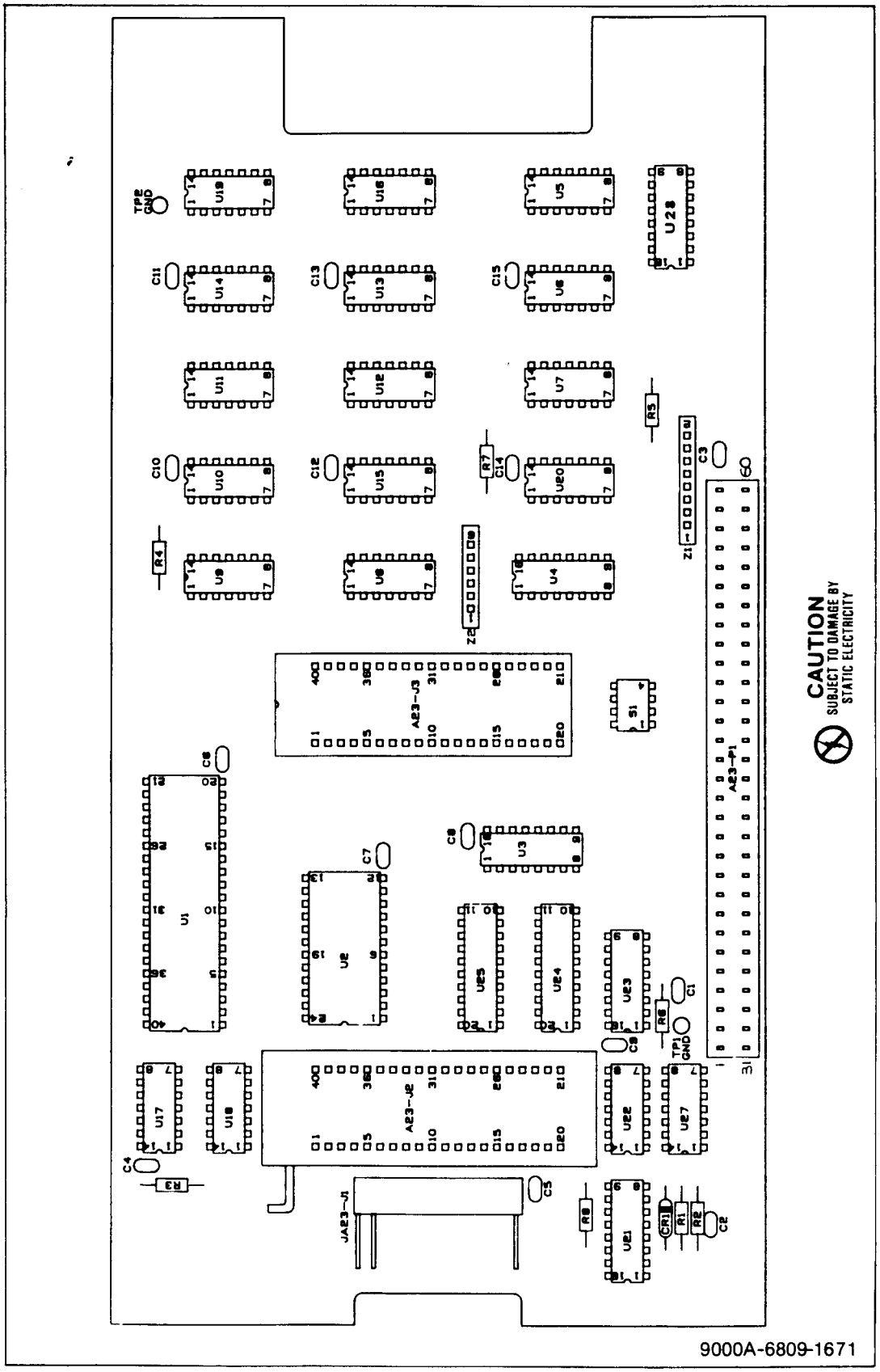
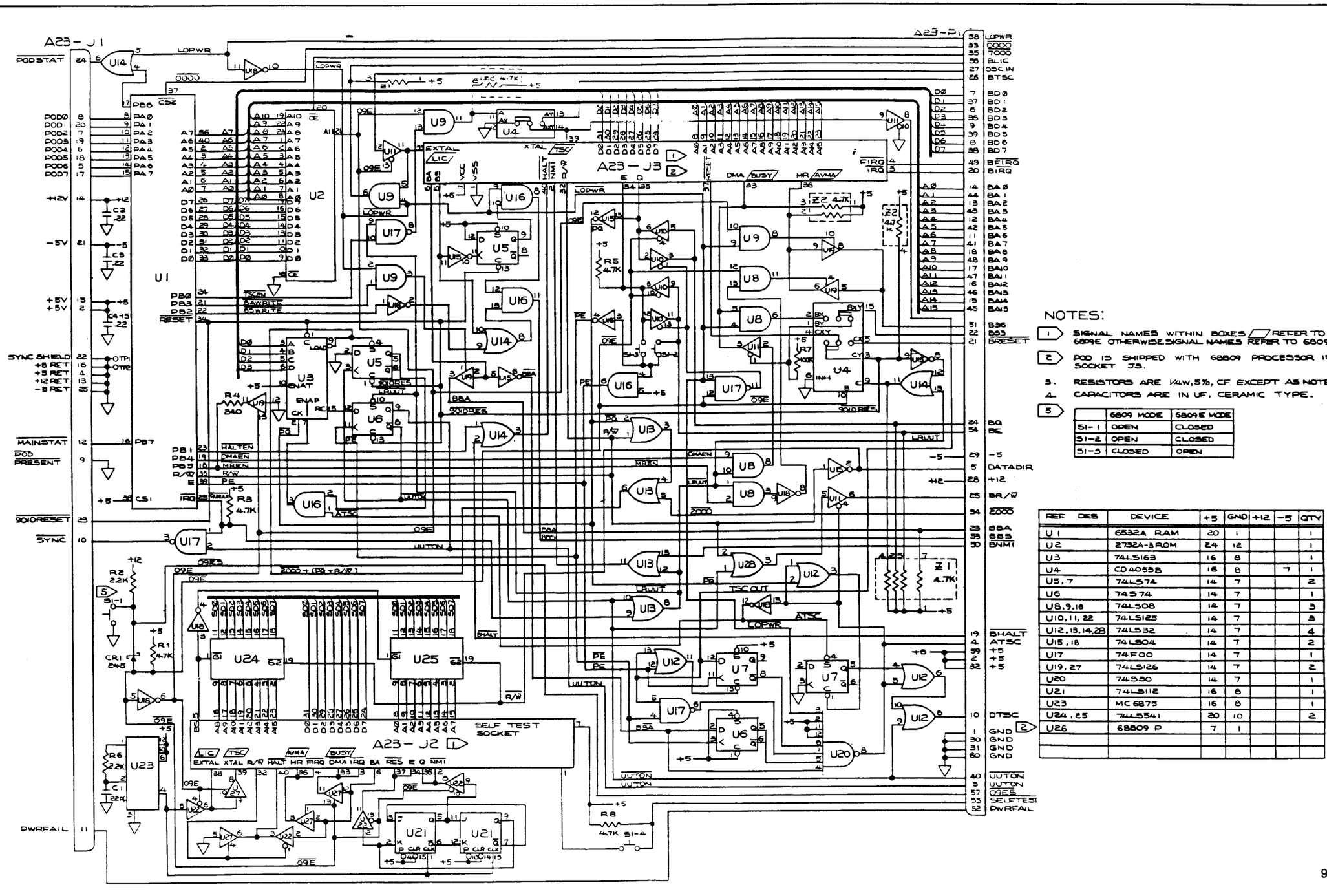
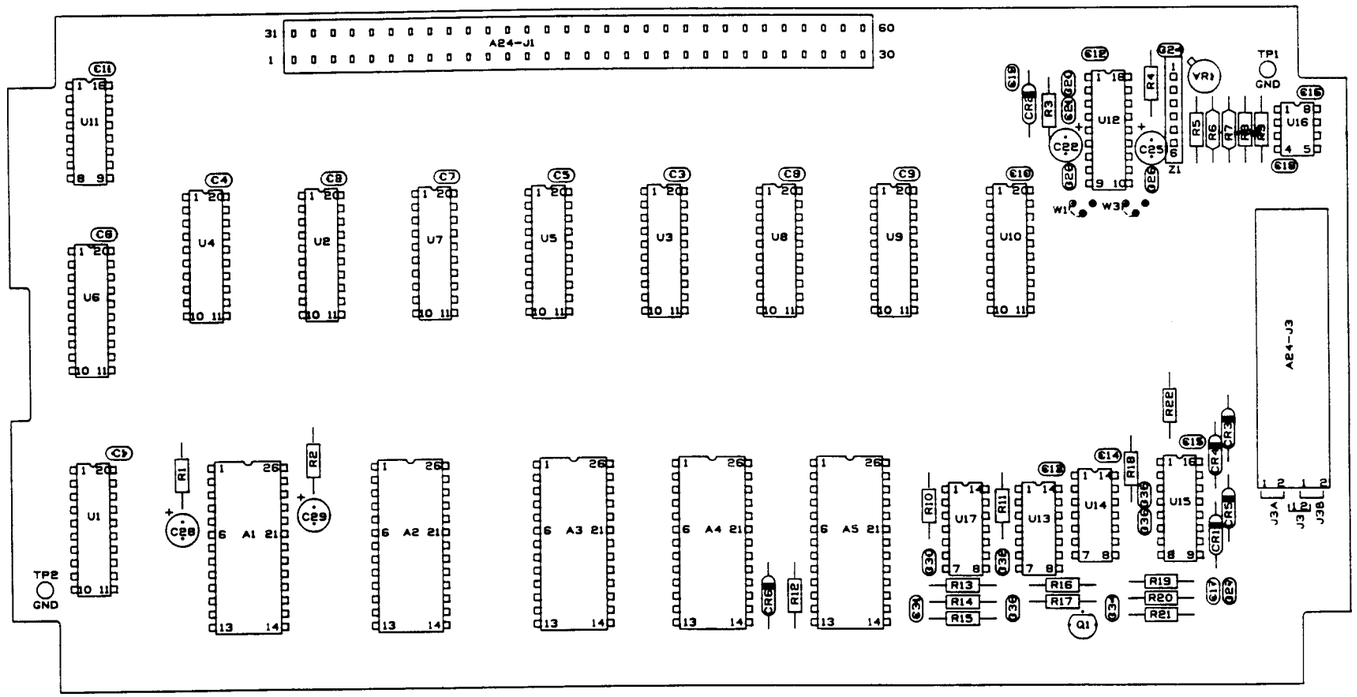


Figure 8-1. A23 Processor PCB Assembly



9000A-6809-1071

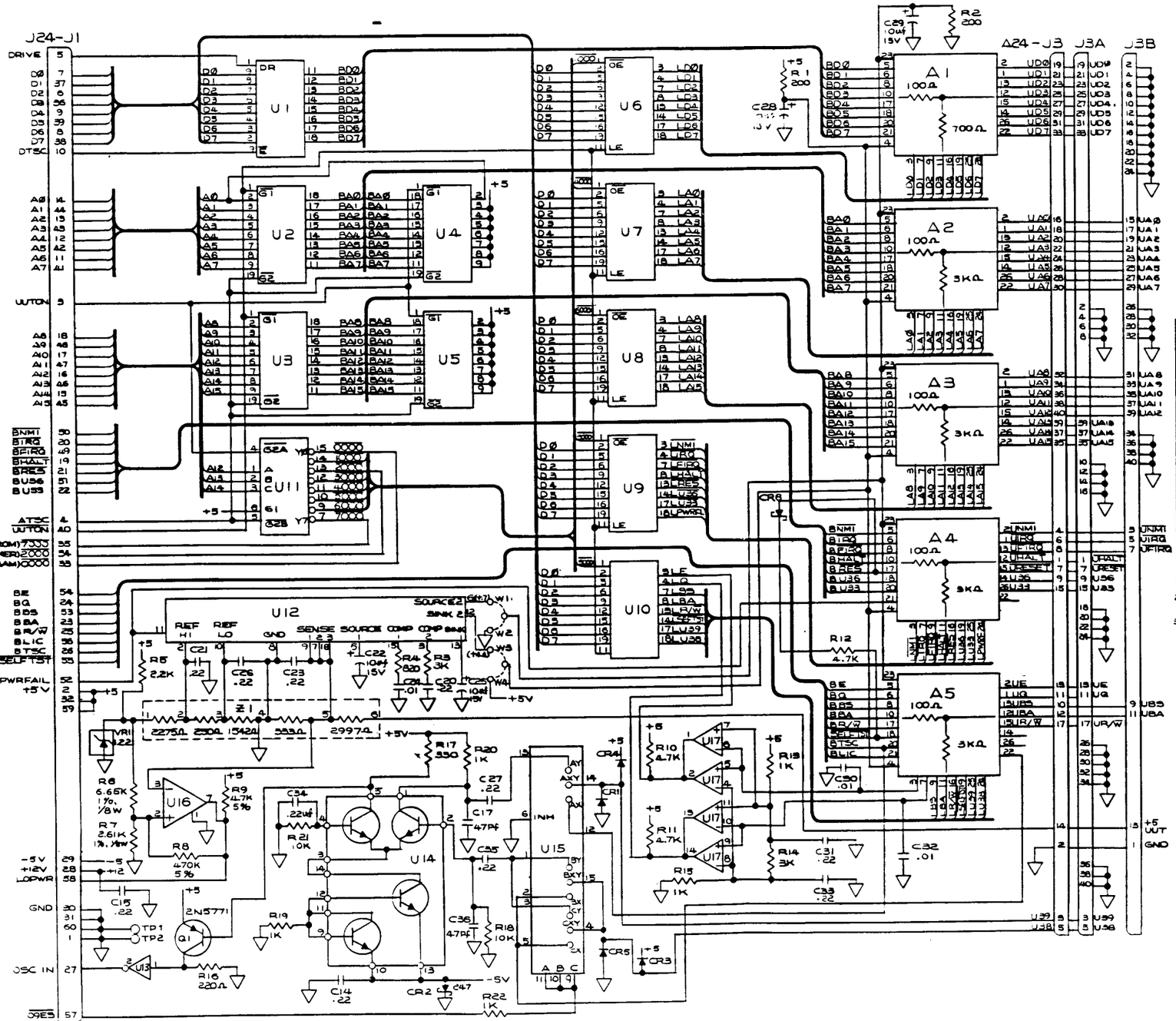
Figure 8-1. A23 Processor PCB Assembly (cont)



CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY

9000A-6809-1672

Figure 8-2. A24 Interface PCB Assembly



REF DES	DEVICE	+5	GND	+12	-5	QTY
U1	74ALS245	20	10			1
U2,3,4,5	74LS541	20	10			4
U6,7,8,9,10	74SC374	20	10			5
U11	74LS138	16	8			1
U12	9000A-8076	1,4	8,14,15		17	1
U13	74LS04	14	7			1
U14	CA 3046				1,9,13	1
U15	CD 4053		8	16	7	1
U16	LM 311	8	1		4	1
U17	LM 339		12	3		1
Z1	54 -4087T		4			1
A1	9000A-4H02T					1
A2,3,4,5	9000A-4H01T					4

NOTES:

1. ALL RESISTORS ARE 1/4W, 5%, CF. OR AS NOTED.
2. ALL CAPACITORS ARE CERAMIC. EXCEPT AS NOTED.
3. ALL DIODES ARE 1N4448 EXCEPT AS NOTED.

-5 — C18,19 — .22

+5 — C1-C13,16 — .22uf

REFERENCE DESIGNATIONS

HIGHEST NO.	DES. NOT USED
A5	
U17	
R22	
CR6	
C36	
VR1	
TP2	
Z1	
W4	

9000A-6809-1072

Figure 8-2. A24 Interface PCB Assembly (cont)

INDEX

The number following each index entry indicates the page number.