NOTE

This manual documents the Model 9000A-68000 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-68000
Interface Pod

Instruction Manual

P/N 652594
APRIL 1983 Rev. 1 3/84

FLUKE®

WARRANTY

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

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

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

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

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

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

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

SHIPPING TO MANUFACTURER FOR REPAIR OR ADJUSTMENT

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

CLAIM FOR DAMAGE IN SHIPMENT TO ORIGINAL PURCHASER

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

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

*For European customers, Air Freight prepaid.

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

Table of Contents

(continued on page ii)

TABLE OF CONTENTS, *continued*

(continued on page iii)

TABLE OF CONTENTS, *continued*

List of Tables

List of Illustrations

Section 1
Introduction

NOTE

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

1-1. PURPOSE OF INTERFACE POD

The purpose of the 9000A-68000 Interface Pod (hereafter referred to as the pod) is to interface any 9000 Series Micro System Troubleshooter (hereafter referred to as the troubleshooter) to equipment employing a 68000 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 is designed to accommodate devices with up to 32 address lines and 32 data lines. The pod adapts the general purpose architecture of the troubleshooter to a specific microprocessor or microprocessor family. 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 microprocessor socket, 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. The pod consists of a pair of printed circuit board assemblies mounted within an impact- resistant case. The pod contains a 68000 microprocessor along with the supporting hardware and control software that is required to do the following:

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

The troubleshooter supplies operating power (+5V, +12V and -5V) for the pod. The UUT provides the external clock signal required by the pod for operation. Using the UUT clock signal allows the troubleshooter and pod to function at the designed operating speed of the UUT.

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

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

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

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

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

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

A 64-pin zero-insertion force self test socket provided on the pod enables the troubleshooter to check pod operation. The ribbon cable plug must be connected to the self test socket during self test operation. The ribbon cable plug should also be inserted into this socket when the pod is not in use to provide protection for the plug.

1-3. SPECIFICATIONS

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

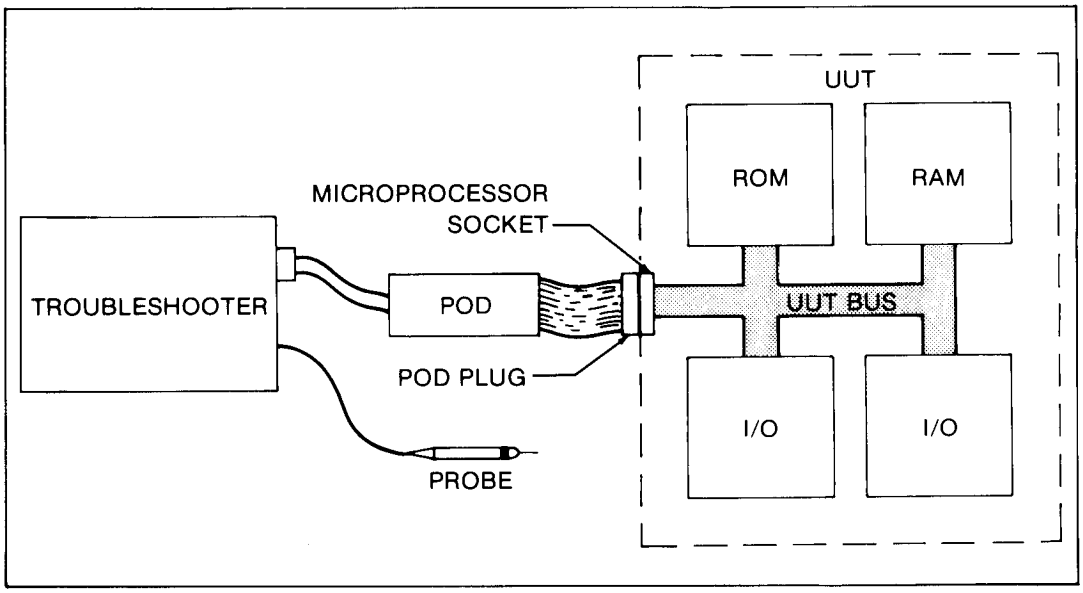


Figure 1-1. Communication Between the Troubleshooter, the Pod, and the UUT

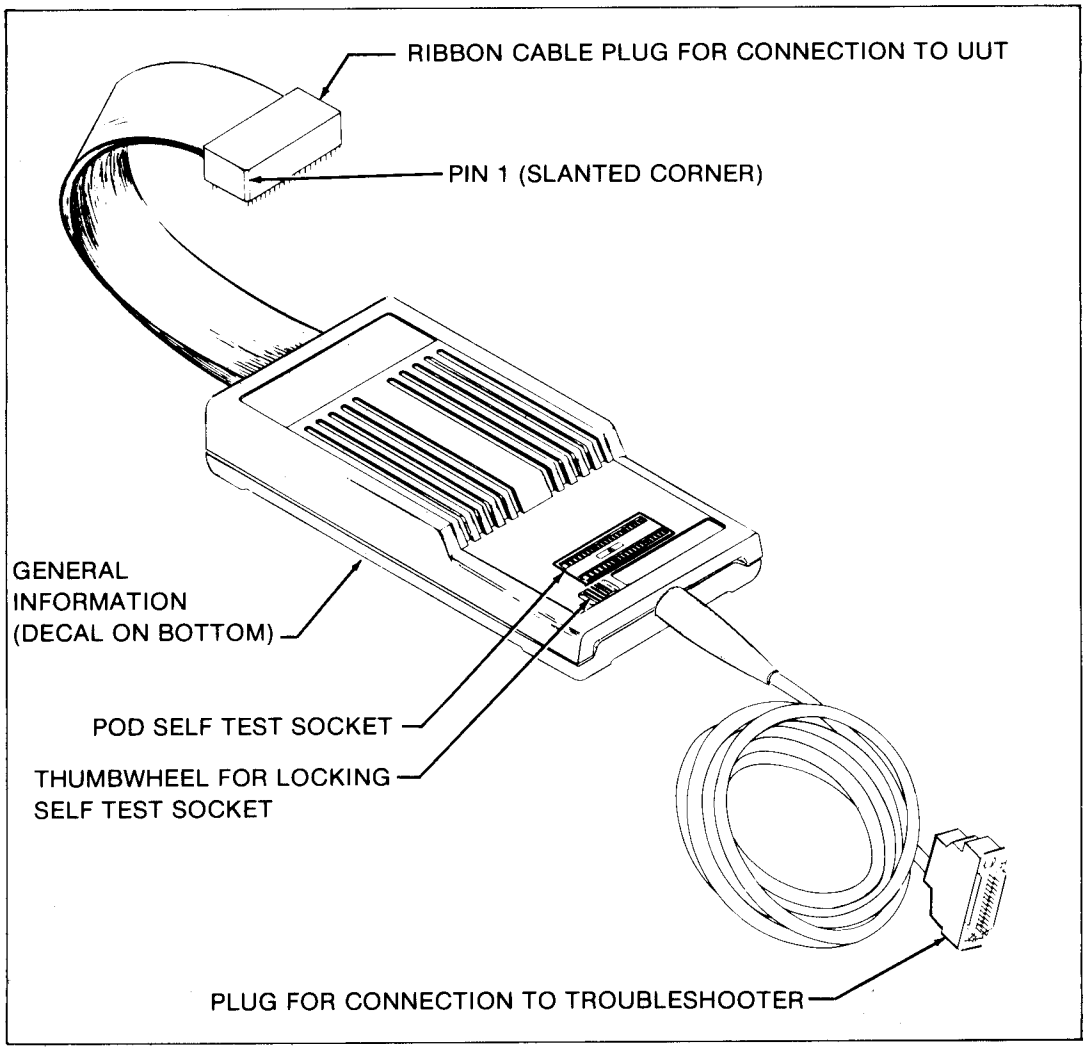


Figure 1-2. External Features of the 68000 Interface Pod

Table 1-1. 68000 Pod Specifications

ELECTRICAL PERFORMANCE

Power Dissipation	9.3 watts max.
Maximum External Voltage	-7V to +12V may be applied between ground and any ribbon cable plug pin continuously.

MICROPROCESSOR SIGNALS*

Input Low Voltage	-0.3V min., 0.8V max.
Input High Voltage	2.0V min., 5.0V max.
Output Low Voltage	0.5V max. at rated current
Output High Voltage	2.4V min. at -400 μA
Tristate Output Leakage Current	±0.02 mA typical, +0.1 to -0.2 mA max.

Input Current

E, $\overline{\text{HALT}}$, $\overline{\text{RESET}}$	-0.5 mA max.
CLK	-0.6 mA max.
All Other Input Lines	±0.02 mA typical, +0.1 to -0.2 mA max.

Table 1-1. 68000 Pod Specifications (cont)

TIMING CHARACTERISTICS	
Maximum External Clock Frequency	10.0 MHz typical
Added Delays to 68000 Signals	
Input signals	12 ns typical
Output signals	15 ns typical
UUT POWER DETECTION	
Detection of Low Vcc Fault	Vcc < +4.5V
Detection of High Vcc Fault	Vcc > +5.5V
Pod protection from UUT Low Power	Vcc < +3.3V**
GENERAL	
Size	5.7 cm H x 14.5 cm W x 27.1 cm L (2.2 in H x 5.7 in W x 10.7 in L)
Weight	1.5 kg (3.3 lbs)
Environment	
STORAGE	-40°C to +70°C, RH < 95% non-condensing
OPERATING	0°C to +40°C, RH < 95% non-condensing +40°C to +50°C, RH < 75% non-condensing

*Signals are specified as they appear at the ribbon cable plug pins.
**Pod outputs tri-state or drive lines to low logic level.

Section 2
Installation and Self Test

2-1. INTRODUCTION

The procedures for performing the pod self test and connecting the pod to the troubleshooter and the UUT are given in the following paragraphs.

2-2. PERFORMING THE POD SELF TEST

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

1. Remove power from the troubleshooter.

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

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

4. Apply power to the troubleshooter.

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

If the troubleshooter displays the message *POD SELF TEST 68000 OK*, the pod is operating properly.

If the troubleshooter displays any message other than *POD SELF TEST 68000 OK*, the pod may not be operating properly. Make sure the pod ribbon cable plug is properly positioned in the self test socket and try the self test again.

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

2-3. 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 between the troubleshooter and 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. Disassemble the UUT to gain access to the UUT microprocessor socket. If the UUT microprocessor is still in the socket, remove the microprocessor.

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

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

6. Reassemble the UUT using extender boards if necessary.

CAUTION

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

7. Apply power to the troubleshooter and the UUT.

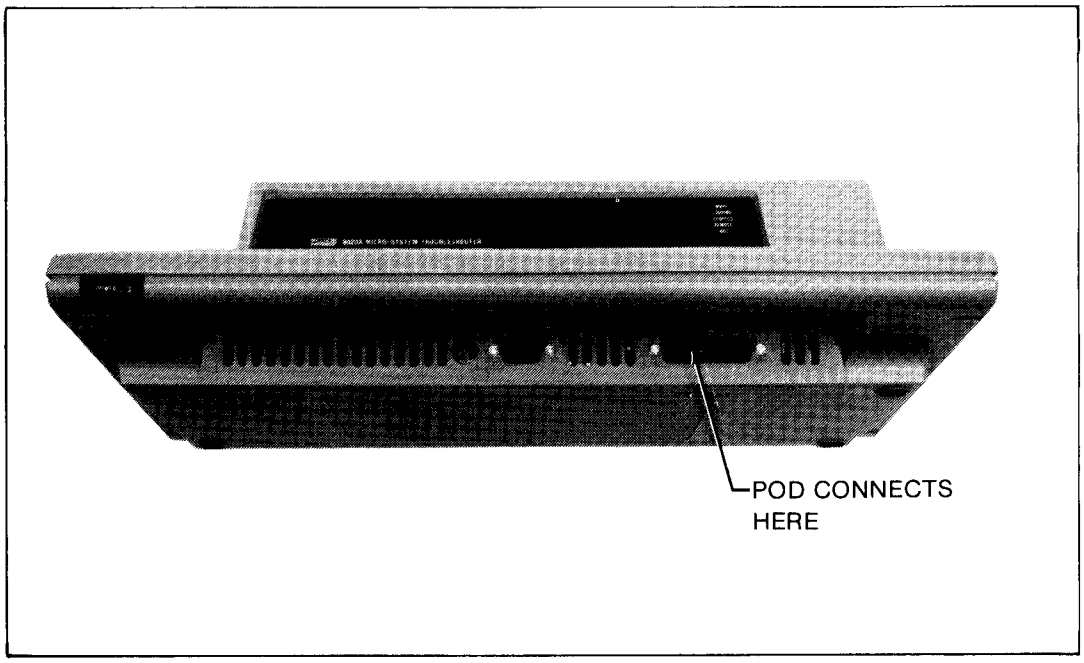


Figure 2-1. Connection of Interface Pod to Troubleshooter

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 68000 signals and pin assignments.

3-2. MICROPROCESSOR SIGNALS

Table 3-1 lists all of the 68000 microprocessor signals and provides a brief description of each signal. Figure 3-1 shows the 68000 pin assignments along with a brief summary of the signals.

Table 3-1. Signal Descriptions

<table>
<tr><th>SIGNAL NAME</th><th>DESCRIPTION</th></tr>
<tr><td>A1-A23</td><td>The Address Bus consists of 23 unidirectional tri-state output lines, and provides the bus address for all processor operations except Interrupt Vector fetch cycles. During Interrupt Acknowledge cycles, address lines A1, A2, and A3 indicate what level interrupt is being serviced, while address lines A4 through A23 are set high. Address bit A0 is used internally by the processor for byte operations.</td></tr>
<tr><td>D0-D15</td><td>The 68000's general purpose data path is a 16-bit, bidirectional, tri-state data bus.</td></tr>
<tr><td>FC0-FC2</td><td>The Function Code lines are tri-state outputs that indicate the state (user or supervisor) and type (program or data) of the current bus cycle. They are also used to indicate an Interrupt Acknowledge cycle, as shown below:
<table>
<tr><th>FC2</th><th>FC1</th><th>FC0</th><th>Cycle Type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>(Undefined, Reserved)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>User Data</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>User Program</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>(Undefined, Reserved)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>(Undefined, Reserved)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Supervisor Data</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Supervisor Program</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Interrupt Acknowledge</td></tr>
</table>
</td></tr>
</table>

Table 3-1. Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
$\overline{AS}$	The Address Strobe line is a tri-state output which indicates that valid data is present on the Address and Function Code lines, and that a bus cycle is in progress.
R/$\overline{W}$	The Read/ Write line indicates the direction data is to be transferred on the Data Bus.
$\overline{UDS}$, $\overline{LDS}$	The Upper Data Strobe and Lower Data Strobe lines are used to indicate if the upper byte, lower byte, or both bytes are to be used for a data bus transation. The exception to this is during the Interrupt Acknowledge cycle when both $\overline{UDS}$ and $\overline{LDS}$ are asserted, but only the lower byte is read.
$\overline{DTACK}$	The Data Transfer Acknowledge line indicates that the bus data transfer will be completed at the end of the current processor clock cycle. During Read Bus cycles, the processor latches the input data at the end of the clock cycle in which $\overline{DTACK}$ is recognized. When reading or writing, $\overline{DTACK}$ is used to terminate the current bus cycle unless superseded by one of the following: ● Bus Error ● Reset
$\overline{BR}$	The Bus Request input is used to indicate to the processor that some other device is requesting control of the bus.
$\overline{BG}$	The Bus Grant line output indicates that the processor will relinquish control of the bus at the end of the current bus cycle.
$\overline{BGACK}$	The Bus Grant Acknowledge input indicates that a device other than the processor has assumed control of the bus.
$\overline{IPL0}$, $\overline{IPL2}$	The Interrupt Priority Level lines indicate the encoded priority level of an interrupting device. These lines have the following characteristics: ● Level 0 indicates no interrupt is pending. ● Level 7 is highest priority interrupt. ● Interrupt levels 1 through 6 are maskable, while level 7 is not.
$\overline{BERR}$	The Bus Error line is used to indicate a problem with the current bus cycle. This line shows the following characteristics when asserted: ● When asserted concurrently with the $\overline{Halt}$ line, the processor will rerun the current bus cycle if the $\overline{Halt}$ line is released first. ● If asserted during Reset Vector Acquisition, the processor will halt. ● If asserted for two consecutive bus cycles, the processor will halt.

Table 3-1. Signal Descriptions (cont)

SIGNAL NAME	DESCRIPTION
$\overline{\text{BERR}}$ (cont)	• If asserted alone, it will cause the processor to execute a non-maskable interrupt to the Bus Error Vector (Vector 2).
$\overline{\text{RESET}}$	The $\overline{\text{Reset}}$ line is a bidirectional open collector line used to reset the state of the processor. When asserted externally in conjunction with the $\overline{\text{Halt}}$ line, the $\overline{\text{Reset}}$ line causes a non-maskable interrupt at Reset Vector (Vector 0).* It may also be utilized by the processor to reset the state of the external environment. This line is asserted by the processor for 124 clock cycles when a Reset instruction is executed. *NOTE* **This is the only exception vector accessed as a supervisory program fetch instead of a supervisory data fetch.*
$\overline{\text{HALT}}$	The $\overline{\text{HALT}}$ line is a bidirectional open collector line that is asserted when the processor stops due to an unrecoverable error sequence. External devices may pull this line low in the following circumstances: • During a bus cycle to stop the processor at the end of the cycle. • To rerun the last bus cycle (used in conjunction with $\overline{\text{BERR}}$). • To reset the processor (when paired with $\overline{\text{RESET}}$).
E	The Enable line output is used to simulate a 6800 type processor clock for interface with 6800 family peripherals. The signal is obtained by dividing the processor clock by ten (duty cycle: six clock periods low, four clock periods high) and runs continuously.
$\overline{\text{VPA}}$	The $\overline{\text{Valid Peripheral Address}}$ line is an input used to request automatic vectoring during Interrupt Acknowledge bus cycles. During all other bus cycles, this line is asserted to request 6800 type bus transactions.
$\overline{\text{VMA}}$	The $\overline{\text{Valid Memory Address}}$ line is an output used to enable 6800 type peripheral devices; it indicates that the processor is synchronized to the Enable line and has placed a valid address on the address bus lines.
CLK	The Clock line is an input used to derive the clocks needed internally by the processor.

Signal Name	Mnemonic	Input/Output	Active State	Driver
Address Bus	A1-A23	output	high	tri
Address Strobe	$\overline{AS}$	output	low	tri
Bus Error	$\overline{BERR}$	input	low	TTL
Bus Grant Acknowledge	$\overline{BGACK}$	input	low	TTL
Bus Grant	$\overline{BG}$	output	low	TTL
Bus Request	$\overline{BR}$	input	low	TTL
Clock	CLK	input	—	TTL
Data Bus	D0-D15	input/output	high	tri
Data Transfer Acknowledge	$\overline{DTACK}$	input	low	TTL
Enable	E	output	high	TTL
Function Code Output	FC0-FC2	output	high	tri
Ground	GND	input	—	—
Halt	$\overline{HALT}$	input/output	low	OC
Interrupt Priority Level	$\overline{IPL0}$ $\overline{IPL2}$	input	low	TTL
Lower Data Strobe	$\overline{LDS}$	output	low	tri
Reset	$\overline{RESET}$	input/output	low	OC
Read/Write	R/$\overline{W}$	output	read-high write-low	tri
Upper Data Strobe	$\overline{UDS}$	output	low	tri
Power Input	Vcc	input	—	—
Valid Memory Address	$\overline{VMA}$	output	low	tri
Valid Peripheral Address	$\overline{VPA}$	input	low	TTL

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

PIN ASSIGNMENTS

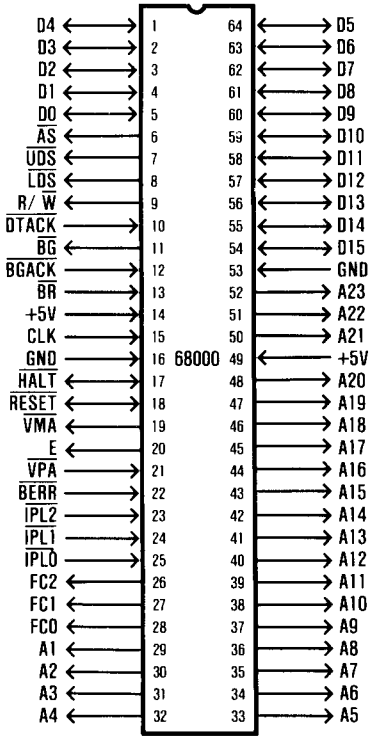


Figure 3-1. 68000 Pin Assignments

Section 4
Operating Information

4-1. INTRODUCTION

This section contains information which pertains to operating the troubleshooter with 68000-based systems. This additional information complements the information in the troubleshooter operator and programming manuals, and covers such items as the following:

- Address space assignment
- Special address functions
- Characteristics of 68000 memory addressing
- Definition and bit assignment of status lines
- Definition of forcing and interrupt lines
- Definition and characteristics of user-writable control lines
- Bit assignment of control lines
- Interrupt handling
- Characteristics of Bus Test, Learn, and Run UUT
- Marginal UUT problems

4-2. GETTING STARTED

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

NOTE

The $\overline{BR}$ and $\overline{BGACK}$ lines are enabled as a pair, and are collectively called BR/ACK in the setup message.

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

NOTE

Operating the pod with the status lines disabled will cause UUT errors if the 68000 microprocessor is required to halt or allow DMA accesses while under test.

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

The troubleshooter may also display the *ACTIVE FORCE LINE* message if the Default Address and/or Function Code are specified at addresses unused by the UUT.

If intermittent problems occur during testing of memory or I/O address space, it is possible that changes to the Standby Address or DTACK Delay may be necessary. These are discussed in a later section titled Special Features of the 68000 Pod.

If the troubleshooter displays an *ACTIVE FORCE LINE* message during the performance of BUS TEST on a properly functioning UUT, it may be necessary to change the Bus Test address using the Setup function of the troubleshooter, or it may be necessary to inhibit reporting of forcing line errors by using the Setup function of the troubleshooter, or by using the forcing line error mask special address. Refer to the subsection titled Force Line Error Mask within the section: Special Features of the 68000 Pod.

4-3. ADDRESS SPACE ASSIGNMENT

4-4. Introduction

The 68000 has 23 physical address lines, A1 through A23, which allow the 68000 to address 16 megabytes (16,777,216 bytes) of memory. The 68000 can address either words (16 bits of data) or bytes (8 bits of data).

The 68000 pod accepts only even addresses for word accesses; if odd addresses are specified for word access, the troubleshooter defaults to the next lower (even) address and displays an error message. Bytes may be accessed at any address.

In addition to the address lines, the 68000 has three function control lines (FC0 through FC2) which allow memory management and indicate when an Interrupt Acknowledge bus cycle is in progress. These lines can be considered an extension of the address bus.

The seventh (most significant) digit of the address entered into the troubleshooter defines the coding of the three function code lines. The function code bits asserted to the UUT are the same as the least significant bits (LSB's) of the seventh digit of the address. Five function codes are currently defined for the processor, but all eight possible combinations may be asserted by the pod. The most significant bit (MSB) of the seventh digit is used to determine whether a word or byte access is to be performed (see Table 4-1).

Table 4-1. Address Space and Function Code Assignments

FUNCTION	ADDRESS	DESCRIPTION	FC2	FC1	FC0
WORD ACCESSES*	100 0000 to 1FF FFFE	User data word access	0	0	1
	200 0000 to 2FF FFFE	User program word access	0	1	0
	500 0000 to 5FF FFFE	Supervisor data word access	1	0	1
	600 0000 to 6FF FFFE	Supervisor program word access	1	1	0
	7FF FFF2 to 7FF FFFE	Interrupt Acknowledge (used to simulate an Interrupt Acknowledge Bus Cycle	1	1	1
BYTE ACCESSES	900 0000 to 9FF FFFF	User data byte access	0	0	1
	A00 0000 to AFF FFFF	User program byte access	0	1	0
	D00 0000 to DFF FFFF	Supervisor data byte access	1	0	1
	E00 0000 to EFF FFFF	Supervisor program byte access	1	1	0

*Only even addresses are allowed in this address space. Odd addresses default to the next lower address.

NOTE

The pod substitutes the designated function code bits for those provided by the 68000, so a function code/address combination not considered valid for a properly functioning 68000 can be driven to the UUT as easily as a proper function code/address combination.

4-5. Special Addresses

In addition to the regular address spaces listed in Table 4-1, the pod recognizes special addresses that are used to access information in the pod or to cause the pod to perform special functions. These special functions include indirect vectoring of RUN UUT, Quick looping, Quick RAM and ROM tests, interrupt handling, and other miscellaneous controls and indicators. These functions are discussed in the section titled Special functions of the 68000 Pod.

4-6. STATUS/CONTROL LINES

4-7. Introduction

The troubleshooter classifies the signals at the microprocessor pins in four categories; address, data, status, and control. The address and data categories are self-explanatory. Status lines are inputs to the microprocessor and control lines are outputs from the microprocessor. The bi-directional lines $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ are considered both status and control lines. The pod permits the operator to monitor the state of the status and control lines and to control the operation of the UUT and the pod by manipulating these lines. The following paragraphs describe these capabilities.

4-8. Status Line Bit Assignments

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

For example: If a Read Status operation is performed and $\overline{\text{DTACK}}$ is the only active status line, the troubleshooter displays:

READ @ STS = 0000 0111 0011 1111 OK

Table 4-2. Status and Control Line Bit Assignments

STATUS LINES		CONTROL LINES	
BIT NUMBER	FUNCTION	BIT NUMBER	FUNCTION
13	**BOTH $\overline{\text{DTACK}}$ AND $\overline{\text{VPA}}$ ASSERTED	12	E
12	**NEITHER $\overline{\text{DTACK}}$ NOR $\overline{\text{VPA}}$ ASSERTED	11	$\overline{\text{AS}}$
		10	R/$\overline{\text{W}}$
		9	$\overline{\text{UDS}}$
11	INTERRUPT VECTOR	8	$\overline{\text{LDS}}$
10	†$\overline{\text{IPL2}}$		
9	†$\overline{\text{IPL1}}$	7	FC2
8	†$\overline{\text{IPL0}}$	6	FC1
		5	FC0
7	PWR FAIL	4	*$\overline{\text{RESET}}$
6	$\overline{\text{DTACK}}$		
5	$\overline{\text{VPA}}$	3	—
4	**$\overline{\text{RESET}}$	2	*$\overline{\text{BG}}$
		1	*$\overline{\text{VMA}}$
3	**$\overline{\text{BERR}}$	0	*$\overline{\text{HALT}}$
2	†**$\overline{\text{BGACK}}$		
1	†**$\overline{\text{BR}}$		
0	†**$\overline{\text{HALT}}$		

*User Writeable
**Forcing Lines
†User Enableable

Bit number 6 is low to indicate the active $\overline{\text{DTACK}}$ line. Note that most status lines are active low; the exceptions are bits 7,11,12 and 13. Bits 14 and 15 are not used and are always zero.

NOTE

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

NOTE

Status bits 7, 11, 12, and 13 do not represent particular 68000 signals, but are generated within the pod to indicate significant events to the troubleshooter operator. (Refer to the section titled Status Lines Generated by the Pod.)

4-9. User-Enableable Status Lines

The 68000 has several inputs which the operator can individually enable or disable ($\overline{\text{HALT}}$, $\overline{\text{BR}}$, $\overline{\text{BGACK}}$, and interrupts) by selecting the proper Setup message using the Setup function of the troubleshooter.

When these inputs are disabled, the UUT-generated signals appearing at these inputs are prevented from affecting the pod.

For example: A stuck-low $\overline{\text{HALT}}$ line would cause the 68000 within the pod to remain in the halted state, preventing normal pod/troubleshooter operation. When the Setup function of the troubleshooter is used to disable this input to the 68000 (by means of pod hardware), the $\overline{\text{HALT}}$ signal is prevented from halting the 68000 pod, allowing normal pod operation.

Any of these status lines may be enabled or disabled using the troubleshooter Setup function. The relevant Setup display message is *SET-ENABLE xxxxxx?* where *xxxxxx* is the name of the status line (there is one setup message for each enable line except the $\overline{\text{BR}}$ and $\overline{\text{BGACK}}$ lines, named BR/ACK by the troubleshooter, which are enabled or disabled as a pair). 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 active condition of a disabled line from being reported on the troubleshooter display

4-10. Status Lines Generated by the Pod

The 68000 pod has several status lines that do not represent particular 68000 signals. These lines are used to provide helpful additional information to the operator. These pod-generated lines are as follows: Power Fail, Interrupt Vector, Neither $\overline{\text{DTACK}}$ Nor $\overline{\text{VPA}}$ Asserted, and Both $\overline{\text{DTACK}}$ and $\overline{\text{VPA}}$ Asserted.

4-11. POWER FAIL STATUS LINE

The Power Fail Status Line is set high by the pod whenever the UUT power supply voltage drops below 4.5V or rises above 5.5V.

4-12. INTERRUPT VECTOR

The Interrupt Vector status bit is set high by the pod whenever interrupt vector information is available. This vector may be read as described in the section titled Interrupt Handling

4-13. NEITHER $\overline{\text{DTACK}}$ NOR $\overline{\text{VPA}}$ ASSERTED

This status bit is set high if neither the $\overline{\text{DTACK}}$ nor the $\overline{\text{VPA}}$ lines were asserted (low) during the previous UUT access bus cycle. The troubleshooter will display an error message if this condition is detected and error reporting for this line has been enabled via the forcing line error mask special address. Refer to the forcing line error mask information presented later in this section.

4-14. BOTH $\overline{\text{DTACK}}$ AND $\overline{\text{VPA}}$ ASSERTED

This status bit is set high if both the $\overline{\text{DTACK}}$ and $\overline{\text{VPA}}$ input lines were asserted (low) during the previous UUT access bus cycle. The troubleshooter displays an error message if this condition is detected and error reporting for this line has been enabled via the forcing line error mask special address. Refer to the forcing line error mask information presented later in this section.

4-15. Forcing Lines

One of the troubleshooter error messages that may be displayed is *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 the following: $\overline{\text{HALT}}$, $\overline{\text{BR}}$, $\overline{\text{BGACK}}$, $\overline{\text{BERR}}$, $\overline{\text{RESET}}$, Neither $\overline{\text{DTACK}}$ nor $\overline{\text{VPA}}$ asserted, and Both $\overline{\text{DTACK}}$ and $\overline{\text{VPA}}$ asserted. The status bits for these functions are shown on the pod decal and in Table 4-2.

Notice that all of the forcing lines, with the exception of $\overline{\text{BERR}}$ and $\overline{\text{RESET}}$, are user-enableable lines. If these user-enableable lines are disabled (via the Setup function or by writing to a Special Address), their inputs to the pod microprocessor are disabled, but the pod continues to monitor their condition; if they are asserted, the pod reports to the troubleshooter that a forcing line is active. If these lines are enabled, they are not considered forcing lines even when they are active, and no *ACTIVE FORCE LINE* message will be displayed.

4-16. Interrupt Lines

Interrupt inputs to the 68000 consist of the three status lines $\overline{\text{IPL0}}$, $\overline{\text{IPL1}}$ and $\overline{\text{IPL2}}$. The pod will enable these interrupt lines and gather interrupt vector information if interrupts have been enabled using the troubleshooter SETUP function. For more detail, refer to the section titled Interrupt Handling.

NOTE

The reporting of active interrupt request lines $\overline{IPL0}$ through $\overline{IPL2}$ is disabled at power on. Reporting of active interrupt lines is enabled by selecting the troubleshooter Setup function message SET-TRAP ACTIVE INTERRUPT? YES.

4-17. User-Writable Control Lines

The 68000 has several control lines which the troubleshooter can assert using the Write Control function. This feature is used by Bus Test to check lines which cannot be toggled by normal read and write operations. It is also useful for helping troubleshoot these lines. The Write Control function is described in the following paragraphs as it pertains to the 68000 pod. Note that the Write Control function only sets a line low (asserted) for approximately one UUT bus cycle, just long enough to verify that it can be driven.

The user writable control lines consist of $\overline{\text{HALT}}$, $\overline{\text{VMA}}$, $\overline{\text{BG}}$, and $\overline{\text{RESET}}$. Table 4-2 summarizes the signal names and bit assignments.

4-18. 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.

Logic levels should be entered into the troubleshooter as they should be driven by the processor.

For example: To drive only the $\overline{\text{BG}}$ line low, the WRITE CTL = 11011 command should be used. Note that when the $\overline{\text{RESET}}$ line is asserted using the Write Control function, it is not asserted long enough to meet the reset requirements of many peripheral devices. If a longer $\overline{\text{RESET}}$ assertion is required, use the Reset instruction provided by special address F000 000A, described in the section titled Special Features of the 68000 pod.

When performing a Bus Test or various other troubleshooter functions, the troubleshooter may detect that one or more control lines are not drivable. For example: Suppose that during a Bus Test the troubleshooter detects that the $\overline{\text{LDS}}$ line is not drivable. The troubleshooter displays the message *CTL ERR 00000001 00000000-LOOP?*. The zeros and ones correspond to the bit numbers assigned to the control lines

as listed in Table 4-2. Bit 8 is set to 1 because the $\overline{LDS}$ line was detected as not drivable. All error messages that pertain to non-drivable control lines use the same bit number assignments as listed in Table 4-2.

NOTE

The logic level of the $\overline{VPA}$ line at the end of a bus cycle determines the expected logic level of the $\overline{VMA}$ line. If a faulty UUT has asserted the $\overline{VPA}$ line long enough to cause the processor to start a 6800-type bus cycle, then releases this line before the end of the bus cycle when $\overline{AS}$ goes high, the pod will report a $\overline{VMA}$ control line error.

4-19. SPECIAL FEATURES AND CONTROLS OF THE 68000 POD

The 68000 pod offers several special functions which enhance its usefulness. These special functions reside in the pod rather than the troubleshooter and are accessed by reading or writing to special addresses outside the standard address space of the 68000 microproccessor.

4-20. QUICK FUNCTIONS

The pod can perform three "quick" functions, the Quick Looping Read and Write, Quick RAM Test, and Quick ROM Test. As their names imply, the advantage of the Quick functions is that they are executed more quickly than the corresponding ordinary functions (Looping Read and Write, RAM Test and ROM Test). The software routines that control Quick functions reside in the pod and not in the troubleshooter, reducing communication overhead and greatly reducing execution time. The special addresses for the quick functions are listed in Table 4-3.

4-21. Quick Looping Read or Write

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

NOTE

To synchronize the troubleshooter TRIGGER OUTPUT to the UUT bus access, the Address Sync mode should be selected. Refer to the section titled Probe and Oscilloscope Synchronization Modes.

To select the Quick Looping function, specify a read or write operation at the address *1XXX XXXX*. The pod first performs a read or write operation at address *XXX 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 many times faster than the ordinary Looping function specified by pressing the LOOP key on the troubleshooter keyboard. During the Quick Loop, the pod does not check for any UUT system errors. Quick Looping continues until the operator selects another operation.

For example: If the operator specifies the operation *READ @ 1600 0000,* the pod will perform a looping read operation at the address 00 0000 (with function code bits 110). If the operator specifies the operation *WRITE @ 1502 00FE = 8C17,* the pod will perform a looping write operation at address 02 00FE (with function code bits of 101), writing the data 8C17.

The Quick Looping function may be used for read or write operations at any of the valid 68000 addresses listed in Table 4-1. The Quick Looping function may not be used with any of the other troubleshooting functions or tests.

Table 4-3. Special Addresses for Quick Functions

FUNCTION	SPECIAL ADDRESSES AND OPERATIONS	DESCRIPTION OF USE
QUICK LOOPING FUNCTION	READ @ 1XXX XXXX WRITE @ 1XXX XXXX = YYYY	A read or write at address 1XXX XXXX will cause the pod to perform a quick looping READ or WRITE (with data YYYY) at address XXX XXXX. UUT system errors are reported only during the first execution of read or write, and not during the succeeding execution loops.
QUICK ROM TEST	Specifying the Test	
	WRITE @ 3XXX XXXX = 0	Specifies starting address (XXX XXXX) for Quick ROM Test.
	WRITE @ 3YYY YYYY = Z1	Specifies ending address (YYY YYYY) and the address increment (Z) for Quick ROM Test. If Z = 0, the address increment defaults to 2 for word addresses 1 for byte addresses.
	Requesting Information About Test Execution	
	READ @ ENTER	After test has been specified, operator can request information about execution results by pressing the keys READ ENTER (address specification is defaulted). The resulting code that is displayed on the troubleshooter indicates the following: CODE MEANING 0000 No test requested 00A0 Aborted test, new command entered 00A1 Aborted test, illegal data in command 00A2 Aborted test, illegal address in command 00A3 Aborted test, illegal address increment XXB0 Busy, test in progress 00C0 Complete, no errors 00C1 Complete, inactive bits detected (x=variable digit; see text)
	READ @ F000 3000 READ @ F000 3002	Low word of starting address High word of starting address
	READ @ F000 3004 READ @ F000 3006	Low word of ending address High word of ending address
	READ @ F000 300C READ @ F000 300E	Checksum* Hex mask indicating inactive bits detected during test
	READ @ F000 30Fx (x = don't care)	Most recent code returned (same as code obtained by READ @ ENTER)

*The checksum obtained by the Quick ROM Test is not related to the ROM signature obtained by the ordinary ROM Test.

Table 4-3. Special Addresses for Quick Functions (cont)

FUNCTION	SPECIAL ADDRESSES AND OPERATIONS	DESCRIPTION OF USE
QUICK RAM TEST	Specifying the Test	
	WRITE @ 2XXX XXXX = 0	Specifies starting address (XXX XXXX) for Quick RAM Test.
	WRITE @ 2YYY YYYY = Z1	Specifies ending address (YYY YYYY) and the address increment (Z) for Quick RAM Test. If Z = 0, the address increment defaults to 2 for word addresses, 1 for byte addresses.
	Requesting Information About Test Execution	
	READ @ ENTER	After test has been specified, operator can request information about execution results by pressing the keys READ ENTER (address specification is defaulted). The resulting code that is displayed on the troubleshooter indicates the following: CODE MEANING 0000 No test requested 00A0 Aborted test, new command entered 00A1 Aborted test, illegal data in command 00A2 Aborted test, illegal address in command 00A3 Aborted test illegal address increment XXB0 Busy, performing read/write check XXB1 Busy, performing address decoding check 00C0 Complete, no errors 00F0 Failed, read/write error 00F1 Failed, address decoding error (x=variable digit; see text)
	READ @ F000 2000 READ @ F000 2002	Low word of starting address High word of starting address
	READ @ F000 2004 READ @ F000 2006	Low word of ending address High word of ending address
	READ @ F000 2008 READ @ F000 200A	Low word of error address High word of error address
	READ @ F000 200C READ @ F000 200E	Data expected at error address Actual data returned from error address
	READ @ F000 2012	Hex mask where ones correspond to bad data bits from read/write failure or address decoding failure*
	READ @ F000 20Fx (x = don't care)	Most recent code returned (same as code obtained by READ @ ENTER)

*If a read/write failure occurred (code F0), the hex mask indicates bits that are not read/writeable both high and low.

*If an address decoding failure occured (code F1), the hex mask corresponds to address bits that are probably at fault. For example, a hex mask of 0080 indicates that address bit 7 is at fault. Note that the first 8 bits in the hex mask correspond to address bits 16 through 24 as well as to address bits 0 through 7 when using word addressing. In addition, when using byte addressing, the first 8 bits in the hex mask correspond to possible decode failures of address bits 8 through 15.

If both error reporting and the Quick Looping feature are desired, you may apply the ordinary troubleshooter Looping function to the Quick Looping read or write, such as *READ @ 1512 3456 ENTER LOOP.* The troubleshooter will command read operations at supervisor data space address 12 3456 at the normal looping speed with full error reporting. For every ordinary read operation, the pod will interject several Quick Looping read operations (with no error reporting) which will enhance oscilloscope viewing.

4-22. Quick RAM Test

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

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

The starting and ending addresses for the Quick RAM Test are specified in a different manner than for the usual RAM Test. The starting and ending addresses are entered by writing to the special addresses listed in Table 4-3. To specify the starting address, enter *WRITE @ 2XXX XXXX = 0* where *XXX XXXX* is the address. To specify the ending address, enter *WRITE @ 2YYY YYYY = Z1,* where YYY YYYY is the address and Z is the desired address increment value. Either word or byte addresses may be used. If Z is not specified, the default value is two for word addressing and one for byte addressing. The ending address must be greater than the starting address, and both addresses must be even for word addresses. The function code assigned to the ending address will also be used for the beginning address. The address increment value must be even for word addresses.

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

To determine if the Quick RAM Test is still in progress, or has been completed, aborted, or failed, the troubleshooter operator should perform a *READ @ ENTER* operation (which commands a READ operation at the last entered address). In response, the pod returns a two byte word, displayed by the troubleshooter in hexadecimal format (with leading zeroes suppressed). The lower byte of this word indicates the status of the test or the test results. The status codes and their meanings are shown in Table 4-4. The upper byte of the pod response shows bits 12 thru 19 of the address under test, which enables the operator to monitor the progress of the pod as it proceeds through the test.

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

The following example shows how to specify the Quick RAM Test over the address block 500 4000 through 500 FFFE, with an address increment of 4 bytes:

WRITE @ 2500 4000 = 00

WRITE @ 2500 FFFE = 41

Table 4-4. Special Address Summary

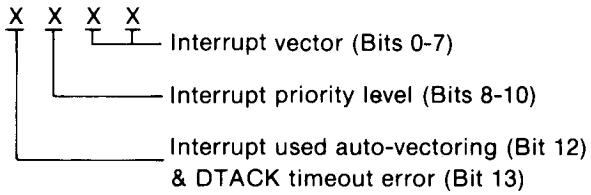
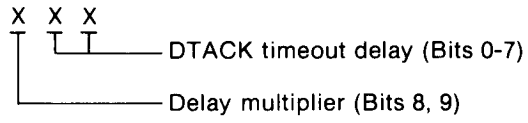
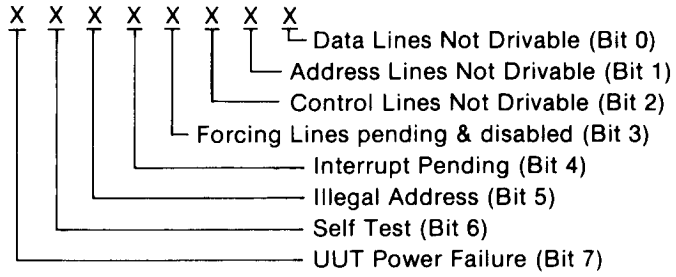
ADDRESS	NAME
F000 0000	Self Test Diagnostic (paragraph 4-27)
F000 0002	Read Interrupt (paragraph 4-28)  NOTE: Bit 11 is always 0.
F000 0004	Read/Enable Interrupt (paragraph 4-29)
F000 0006	Set Interrupt Mask (paragraph 4-30)
F000 000A	Execute Reset Instruction (paragraph 4-31)
F000 0010	DTACK Delay (paragraph 4-32)  0 = 1x 1 = 8x 2 = 64x 3 = 1024x
F000 0014	Standard Function code (paragraph 4-34)
F000 0016	Standby Address (paragraph 4-35)
F000 0018	Standby Function Code (paragraph 4-36)
F000 001A	Default Address (paragraph 4-37)
F000 001C	Default Function Code (paragraph 4-38)
F000 0020	Last Error Summary (paragraph 4-39) 
F000 0022	Last Control Errors (paragraph 4-40)
F000 0024	Last Force Line Errors (paragraph 4-41)
F000 0026	Last Status (paragraph 4-42)

Table 4-4. Special Address Summary (cont)

ADDRESS	NAME
F000 0028	Error Summary Mask (paragraph 4-43)
F000 002A	Control Error Mask (paragraph 4-44)
F000 002C	Force Line Error Mask (paragraph 4-45)
F000 0030	RUNUUT Supervisor Stack Pointer - lower (paragraph 4-46)
F000 0032	RUNUUT Supervisor Stack Pointer - upper (paragraph 4-47)
F000 0034	RUNUUT User Stack Pointer - lower (paragraph 4-48)
F000 0036	RUNUUT User Stack Pointer - upper (paragraph 4-49)

Note that the Quick RAM Test has more flexibility than the ordinary RAM Test with regard to the address increment and specification. For example, only word addresses may be tested with the ordinary RAM Test. However, consider the following Quick RAM Test specification:

WRITE @ 2D0E 0000 = 0

WRITE @ 2D0E 07FF = 21

The two preceding write operations specify a Quick RAM Test which is to take place at the odd byte addresses over the address block E 0001 through E 07FF using a function code of 101. The value of the address increment is two bytes. This would be useful for testing memory in cases where only the lower byte of RAM has been implemented in the UUT.

4-23. Quick ROM Test

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

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

The Quick ROM Test is specified in a manner similar to the Quick RAM Test. To specify the starting address, enter *WRITE @ 3XXX XXXX = 0* where *XXX XXXX* is the address. To specify the ending address, enter *WRITE @ 3YYY YYYY = Z1* where YYY YYYY is the address and Z is the desired address increment value. Either word addresses or byte addresses may be specified. If Z is not specified, the default address increment is two bytes for word addressing and one for byte addressing. The ending address must be greater than the starting address, and both addresses must be even for word addresses. The function code assigned to the ending address is also used for the beginning address. The address increment must be even for word addresses.

Like the Quick RAM Test, the Quick ROM Test may be aborted by selecting another operation. To determine if the Quick RAM Test is still in progress, or has been completed, aborted, or failed, the troubleshooter operator should perform a *READ @ ENTER* operation (which commands a Read operation at the last entered address). In response, the pod returns a two byte word, which is displayed by the troubleshooter in

hexadecimal format (with leading zeroes suppressed). The lower byte of this word indicates the status of the test or the test results. The status codes and their meanings are shown in Table 4-4. The upper byte of the pod response shows bits 12 through 19 of the address under test; therefore, the operator can monitor the progress of the pod as the test proceeds.

4-24 RUN UUT INDIRECT VECTORING

Run UUT Indirect vectoring uses a vector contained in the UUT supervisor data address space or supervisor program address space as the beginning address for the Run UUT command.

4-25. Run UUT Indirect

The addresses F5XX XXXX or F6XX XXXX are used for Run UUT Indirect functions.

If F5XX XXXX is used, addresses XX XXXX and XX XXXX+2 of the UUT supervisor data address space contain the address where instruction execution begins. Control is transferred to that address.

If F6XX XXXX is used, the contents of addresses XX XXXX and XX XXXX+2 of the supervisor program space are loaded into the supervisor stack pointer, and the contents of addresses XX XXXX+4 and XX XXXX+6 of the supervisor program space are used as the beginning address. Control is transferred to the beginning address, with the stack pointer indicating a stack in a new location.

4-26. SPECIAL FEATURES OF THE 68000 POD

The following paragraphs describe additional special features of the 68000 pod accessed through special addresses, which are summarized in Table 4-4.

NOTE

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

4-27. Self Test Diagnostic (Address F000 0000)

The self test operation consists of 8 steps. After self test is complete, this location contains the number of the first step, if any, that failed. Refer to Section 6 for additional information.

4-28. Read Interrupt (Address F000 0002)

Reading at this address returns the latched interrupt information, but does does not enable interrupts. The bit assignments for this address are as follows:

Bits 15, 14	Not Used
Bit 13	Interrupt $\overline{\text{DTACK}}$ timeout error
Bit 12	Interrupt used Auto-Vectoring
Bit 11	Not Used
Bits 10 - 8	Priority Level of Interrupt
Bits 7 - 0	Interrupt Vector

For example: A display reading of *READ @ F000 0002 = 15FF OK* indicates that auto-vectoring with an interrupt priority level of 5 occured. The vector FF indicates that the data bus was probably tri-stated.

If the display read: *READ @ F000 0002 = 03A5,* this would indicate an interrupt priority level 3 and an interrupt vector of A5.

4-29. Read/Enable Interrupt - Address F000 0004

Reading at this address returns the latched interrupt information and enables interrupts. The bit assignments are the same as address F000 0002.

4-30. Set Interrupt Mask - Address F000 0006

Writing to this address will set the interrupt mask bits of the 68000 status register as follows:

Bits 15 - 3 Not Used

Bits 2 - 0 Interrupt Priority Mask bits I2-I0

The bit default value is 0, allowing all interrupt priority levels.

For example: *WRITE @ F000 0006 = 4* allows interrupt levels of 5, 6, and 7 only.

4-31. Execute Reset - Address F000 000A

Writing anything to this address causes the pod processor to execute a RESET instruction which causes the RESET line to be asserted to the UUT for 124 clock pulses.

For example: *WRITE @ F000 000A = nn* (where nn is any hexadecimal integer) will cause the processor to assert the RESET line to the UUT for 124 clock pulses.

4-32. $\overline{\text{DTACK}}$ Delay - Address F000 0010

When the pod attempts to access the UUT, it waits for a $\overline{\text{DTACK}}$ or $\overline{\text{VPA}}$ assertion to terminate the bus transaction. If neither signal is asserted, the pod will wait until the $\overline{\text{DTACK}}$ timer times out, terminating the bus transaction.

Writing to this address sets the $\overline{\text{DTACK}}$ delay value using bits 9, 8 and 7-0. The default unprogrammed $\overline{\text{DTACK}}$ delay value is A0, causing the pod to wait for 100μ sec before internally terminating a bus transaction.

Bits 15 - 10 Not Used

Bits 9, 8 Delay multiplier Bits

Multiplier	Bit 9	8
1X	0	0
8X	0	1
64X	1	0
1024X	1	1

Bits 7 - 0 This sets the $\overline{\text{DTACK}}$ Timeout Delay in multiples of 10 clock pulses with a maximum value of F8. Values larger than F8 disable the timer so that the pod will wait until the troubleshooter times out. Troubleshooter timeout is explained in more detail in Section 4M of the 9010A Operators Manual.

For example: *WRITE @ F000 0010 = 8* sets $\overline{\text{DTACK}}$ Delay to 8μ sec (10 MHz UUT clock) which programs the pod to wait for 8 μ sec before internally assserting $\overline{\text{DTACK}}$. *WRITE @ F000 0010 = 2A0* causes the pod to wait for 10.2 ms before internal $\overline{\text{DTACK}}$ assertion.

4-33. Standard Function Code - Address F000 0014

The standard function code bits are used if the pod receives a function code of all zeros. This allows the user to specify short addresses rather than having to specify seven digits of address for every pod operation. (If the operator desires to assert a function code of all zeros to the UUT, he should write zero to this address.) The default value for this address is 0006. The bit assignments are as follows:

Bits 15 - 3 Not Used

Bits 2 - 0 Standard Function Code

For example: Entering the command *WRITE @ F000 0014 = 02* will cause the pod to substitute a function code of 2 for any subsequently entered UUT address if the entered function code is zero (or not entered). Thus, after the above entry, the command *READ @ A4* is equivalent to the command *READ @ 200 00A4.*

4-34. Standby Address - Address F000 0016

When the pod is not performing an actual UUT access such as Read or Write, the Standby Address is asserted to the UUT. Only the upper sixteen bits of the Standby Address are programmable. The lower eight bits are always zero. The bit assignments for this address are as follows:

Bits 15 - 0 Address bits 23 - 8 while idle

Address bits 7 - 0 always zero

Default Value = 0000

For example: To assert the address 4F FF00 on the address bus when not performing a UUT access, enter: *WRITE @ F000 0016 = 4FFF.*

4-35. Standby Function Code - Address F000 0018

When the pod is not performing an actual UUT access such as Read or Write, the standby function code is asserted to the UUT. The bit assignments for Standby Function Code are as follows:

Bits 15 - 3 Not Used

Bits 2 - 0 Function Control While Idle

Default Value = 0006

For example: To assert the function code 101 on 68000 processor lines FC2-FC0 when not performing a UUT access, enter: *WRITE @ F000 0018 = 5.*

4-36. Default Address-Address F000 001A

Certain commands such as Read Status and Write Control do not have explicit addresses associated with them. However, the pod does access the UUT as part of the command execution. The address used for this access is the Default Address. The upper 16 address bits are programmable. The lower eight address bits are always zero. Word accessing is always used. The bit assignments are as follows:

Bits 15 - 0 Address bits 23 -8
Address bits 7 - 0 are always 0

Default value = 0000

For example: The pod must perform a read operation on the UUT bus to determine the state of the status lines for the READ STS operation. Normally, the pod would perform the read at address 00 0000 (using the default function code bits as described below). To change the address used to AB CD00, enter: *WRITE @ F000 001A = ABCD.*

4-37. Default Function Code - Address F000 001C

Certain commands such as Read Status and Write Control do not have explicit addresses associated with them. However, the pod does access the UUT as part of the command execution. The function code used for this access is the Default Function Code. The bit assignments are as follows:

Bits 15 - 3 Not Used

Bits 2 - 0 Default Function Code

Default value = 0006

For example: The pod must perform a read operation on the UUT bus to determine the state of the status lines for the READ STS operation. Normally, the pod would perform the read at address 00 0000 (using the default function code value of 110). To change the function code bits used to 001, enter: *WRITE @ F000 001C = 1*

4-38. Last Error Summary - Address F000 0020

The user may inhibit the reporting of errors detected by the pod by using the setup functions of the troubleshooter or the Error Summary Mask special address. This address is used to determine the pod error status even though error reporting by the troubleshooter has been inhibited. A summary of any errors detected by the pod during the immediately previous UUT operation may be read from this address. The bit assignments are as follows:

Bit 7 UUT Power Failure

Bit 6 Self Test

Bit 5 Illegal Address

Bit 4 Interrupt Pending

Bit 3 Forcing Line(s) Pending and Disabled

Bit 2 Control Line(s) Not Drivable

Bit 1 Address Line(s) Not Drivable

Bit 0 Data Line(s) Not Drivable

For example: If the pod UUT connector is plugged into the self test socket and all error reporting is inhibited by writing 0 to the Error Summary Mask Special Address, performing a *READ @ F000 0020 = 0018* indicates that:

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

NOTE

When self test reporting is inhibited by the error summary mask, it is not reported in the error summary.

4-39. Last Control Errors - Address F000 0022

The control error word from the immediately previous UUT operation may be read from this address. The normal control line bit assignments are used. Refer to Table 4-2 or the pod decal (on the bottom of the pod) for control line bit assignments.

For example: If the user has disabled the reporting of control line drivability errors (to avoid interruptions while running a pre-programmed series of tests, for instance), performing a *READ @ F000 0022 = 0002 OK* would indicate that the $\overline{\text{VMA}}$ line was not drivable when the last UUT operation was performed.

4-40. Last Forcing Line Error - Address F000 0024

The forcing line error word from the immediately previous UUT operation may be read from this address. Status line bit assignments are used. Refer to Table 4-2 or the pod decal (on the bottom of the pod) for status line bit assignments.

For example, *READ @ F000 0024 = 0002* indicates an error with the BR forcing line in the previous UUT operation.

4-41. Last Status - F000 0026

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

For example, *READ @ F000 0026 = 073F* shows DTACK to be the only active status line. Compare this to the READ @ STS example under paragraph 4-8, Status Line Bit Assignments.

4-42. Error Summary Mask - Address F000 0028

The reporting of any class or classes of errors (including Self Test) may be suppressed by setting the appropriate bits of the error summary mask to zero. The complete error summary can still be read from the Last Error Summary special address, but the suppressed classes of errors will not be reported by the troubleshooter. The bit assignments are as follows:

Bit 7 UUT Power Failure

Bit 6 Self Test

Bit 5 Illegal Address

Bit 4 Interrupt Pending

Bit 3 Forcing Line(s) Pending and Disabled

Bit 2 Control Line(s) Not Drivable

Bit 1 Address Line(s) Not Drivable

Bit 0 Data Line(s) Not Drivable

Default Value = FF

For example: To inhibit the reporting of pending interrupts without using the setup function (this is useful for running tests under program control), enter: *WRITE @ F000 0028 = EF.*

4-43. Control Drivability Error Mask - Address F000 002A

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

Default value = FFFF

For example: Some 68000 UUT's will not allow the processor to drive the $\overline{\text{RESET}}$ line low. If this is considered normal, performing a WRITE @ F000 002A = FFEF will inhibit the reporting of $\overline{\text{RESET}}$ line drivability errors during BUS TEST, while allowing drivability error reporting for all of the other control lines.

4-44. Forcing Line Error Mask - Address F000 002C

The reporting of any individual forcing line error (e.g., forcing lines asserted but not enabled) may be suppressed by setting the appropriate bits of the forcing line error mask to zero. The bit assignments correspond to those shown in Table 4-2 (Status and Control Line Bit Assignments). The complete error summary can be read from the Last Control Errors special address, however errors corresponding to the suppressed bits will not be reported by the troubleshooter. Bits 12 and 13 are normally suppressed.

Default value = 0FFF

For example: Many 68000 UUT's assert the $\overline{\text{BERR}}$ status line in response to a bus read or write at an unimplemented address. However, the troubleshooter BUS TEST operation is very likely to assert an unmpliented address to the UUT while checking the drivability of the address lines. Performing a *WRITE @ F000 002C = FFF7* will inhibit $\overline{\text{BERR}}$ reporting while allowing forcing line error reporting for other status inputs.

4-45. Supervisor Stack Pointer - Lower - Address F000 0030

The contents of this location are loaded into the lower 16 bits of the 68000's supervisor stack pointer when the RUN UUT command is executed. However, if an F6 prefix has been used as a part of the RUN UUT address (see the previous section titled Run UUT Indirect), the supervisor stack pointer will not receive the contents of the Supervisor Stack Pointer special address.

Default value = 4000

4-46. Supervisor Stack Pointer - Upper - Address F000 0032

The contents of this location are loaded into the upper 16 bits of the supervisor's stack pointer when the RUN UUT command is executed. Again, the restrictions when dealing with an F6 prefix also apply here.

Default value = 0000

4-47. User Stack Pointer - Lower - Address F000 0034

The contents of this location are loaded into the lower 16 bits of the user stack pointer when the RUN UUT command is executed.

Default value = 4000

4-48. User Stack Pointer - Upper - Address = F000 0036

The contents of this location are loaded into the upper 16 bits of the User Stack pointer when the RUN UUT command is executed.

Default value = 0000

4-49. DEFAULT ADDRESSES FOR LEARN, BUS TEST, AND RUN UUT

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

4-50. Learn Operation Default Address

If the Learn operation is selected and the operator does not specify the starting and ending addresses for the operation, the pod specifies the default address spaces of 100 0000 through 1FF FFFE, 200 0000 through 2FF FFFE, 500 0000 through 5FF FFFE, and 600 0000 through 6FF FFFE. The Learn operation is performed over these address spaces. It would take the troubleshooter several days to learn such a large memory space, so it is wise to specify a smaller address range(s) if possible.

4-51. Bus Test Default Address for Data Line Testing

When the operator selects Bus Test, no address is explicitly required. However, as part of Bus Test, the data lines are tested at a particular address supplied by the troubleshooter. For the 68000 pod, the data line testing occurs at address 100 0FFE. Note that the operator may change this address with the troubleshooter Setup function by entering the desired address for the Setup message *SET-BUS TEST @ 100 0FFE-CHANGE?*

4-52. Run UUT Mode

The Run UUT mode allows the pod to emulate the UUT microprocessor by executing a program directly from UUT memory. When the operator selects Run UUT, the operator may either explicitly specify the address where execution begins, or the operator may invoke the Run UUT default execution address which is supplied by the pod. The default execution address is normally F600 0000, but may be changed by entering the device address for the Setup message *SET-RUN UUT @ F600 0000 CHANGE?*. Run UUT at the F600 0000, default address will cause the pod to start execution as it would if the UUT were reset. That is, the contents of supervisor program memory locations 0 and 2 will be read and loaded into the supervisor stack pointer of the 68000, and the contents of supervisor program memory locations 4 and 6 will be used as the starting address. (See the previous section titled Run UUT Indirect.)

4-53. INTERRUPT HANDLING

Using the Setup function of the troubleshooter, the operator has the option of enabling or not enabling interrupts. When interrupts are enabled, the pod responds to an interrupt request from the UUT with an interrupt acknowledge bus cycle. Although no attempt is made to execute the interrupt service routine of the UUT, the pod does record the interrupt type and vector.

To enable interrupts, set the microprocessor interrupt mask level lower than the expected interrupt level by writing 000x to Special Address F000 0006, where x is less than the expected interrupt priority level. Also, ensure that the troubleshooter Setup function, *SET - TRAP ACTIVE INTERRUPT?*, is *YES* to enable interrupt processing. When this has been done, bit 11 in the status word will indicate whether or not the pod has sensed an interrupt. For example, a *READ @ STS = xxxx 0xxx xxxx xxxx OK* indicates that interrupt information has not yet been obtained.

If an interrupt of higher priority than the processor interrupt mask level occurs, the pod is forced to perform an interrupt acknowledge bus cycle. A *READ @ STS = xxxx 1xxx xxxx xxxx OK* indicates, by the enabled INTERRUPT VECTOR status bit (bit 11), that

interrupt data is now saved. Further interrupts are disabled at this point to protect the interrupt data from being overwritten before it is accessed. Interrupt data may now be read at either address F000 0002 or F000 0004. Reading at location F000 0004 will reenable interrupts, while reading at location F000 0002 will allow interrupts to remain disabled.

If interrupts have been enabled but none have occurred, interrupt information obtained from the special addresses will be all zeroes.

NOTE

If a BUS TEST or SYNC operation is performed, the latched interrupt information is lost and the interrupts enabled.

Interrupts may occur at any time, and will be acknowledged as soon as they are requested by the UUT (at the completion of the current 68000 instruction), except for brief periods before, during, and after normal troubleshooter UUT accesses.

4-54. PROBE AND OSCILLOSCOPE SYNCHRONIZATION MODES

The operator may use the troubleshooter Synchronization function (selected with the SYNC key) to synchronize probe operation and rear panel TRIGGER OUTPUT pulses to the pod's microprocessor bus events. With the 68000 pod, there are four synchronization modes available:

A = Address Sync

D = Data Sync

F = Free-Run

I = Interrupt Sync

The pod generates a sync signal which is used by the mainframe for the probe and trigger output signals. In either address or data sync modes, the pod sync signal begins one microprocessor clock cycle before the start of the UUT access and ends at the completion of the UUT bus cycle (as $\overline{AS}$ goes high). In the interrupt sync mode, the pod sync signal will be similar to those for the address and data sync modes, but will occur only during an interrupt acknowledge bus cycle.

If the probe is being used as a pulser, it will pulse at the selected level (high or low) for the entire time between the beginning and end of the pod sync signal. The probe will latch the signal level present at its tip at the end of the pod sync signal. The TRIGGEROUT signal will generate a negative trigger pulse at the beginning of the pod sync signal and a positive pulse at the end of the sync signal. See TRIGGEROUT in Figure 5-4.

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

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

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

4-55. 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 paragrahs which follow.

4-56. 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.

4-57. UUT Noise Levels

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

4-58. 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.

4-59. Clock Loading

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

4-60. POD DRIVE CAPABILITY

As a driving source on the UUT bus, the pod provides equal to or better than normal 68000 current drive capability. All pod inputs and outputs are TTL compatible.

4-61. 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.

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

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

Section 5
Theory of Operation

5-1. INTRODUCTION

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

5-2. GENERAL POD OPERATION

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

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

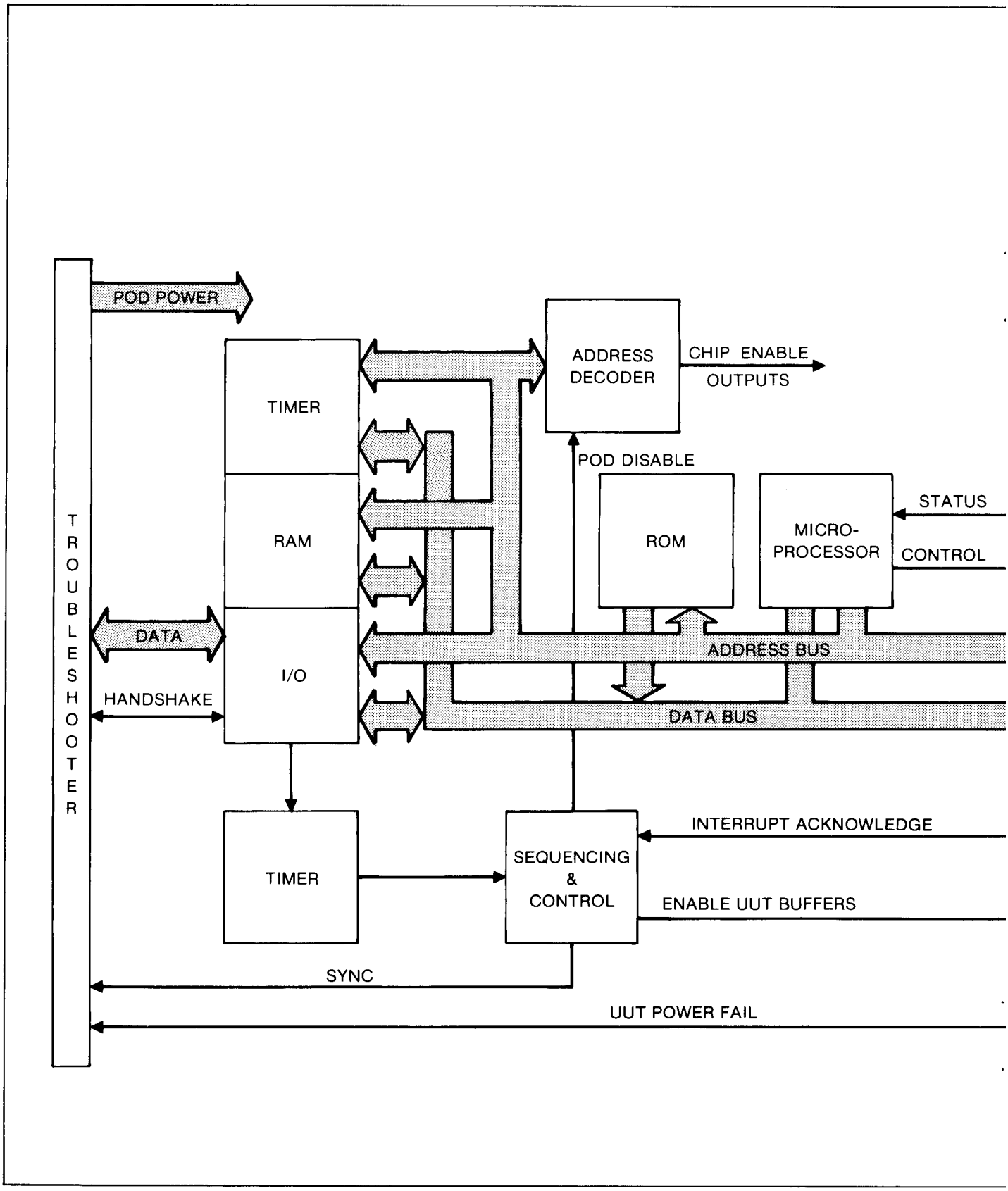
The pod may be divided into the following three major sections:

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

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

5-3. Processor Section

The Processor Section, shown in Figure 5-1, is made up of the microprocessor, RAM, ROM, I/O, and various latches and buffers. These elements, along with some timing signals, comprise a small microsystem which receives commands from the troubleshooter and performs specific operations in response to these commands.


POD POWER
TROUBLESHOOTER
TIMER
RAM
I/O
DATA
HANDSHAKE
ADDRESS DECODER
CHIP ENABLE OUTPUTS
POD DISABLE
ROM
MICRO-PROCESSOR
STATUS
CONTROL
ADDRESS BUS
DATA BUS
TIMER
SEQUENCING & CONTROL
INTERRUPT ACKNOWLEDGE
ENABLE UUT BUFFERS
SYNC
UUT POWER FAIL

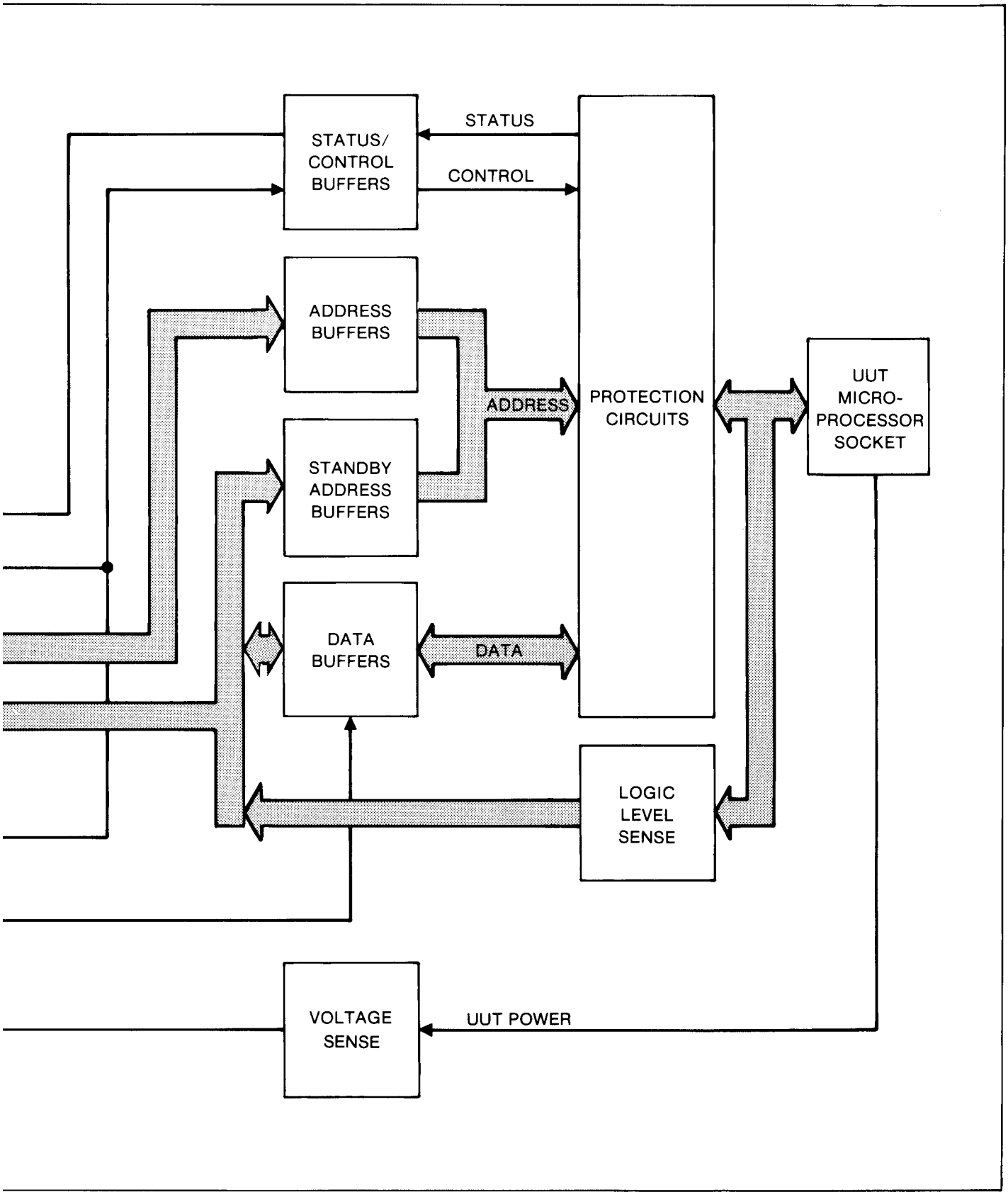


Figure 5-1. 68000 Pod General Block Diagram

For the 68000 pod, the microprocessor inputs are referred to as status lines, and the outputs are referred to as control lines. This nomenclature is not always in agreement with the manufacturer's literature. This convention, however, allows consistency between pods when implementing the troubleshooter functions which involve status or control lines, such as Read Status or Write Control. The $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ lines are bidirectional and are considered both status and control lines.

All pod status lines which could adversely affect the pod operation are either automatically disabled by the pod, or may be disabled by the operator using the troubleshooter Setup function; the disabling is accomplished with input buffers. Disabling these status lines allows the pod to operate in hostile UUT environments where malfunctioning status lines, such as $\overline{\text{HALT}}$, could prevent the pod from performing any tests. The one microprocessor input which may not be disabled, of course, is the UUT clock. The clock signal must always be present for pod operation. All the status lines are enabled in the Run UUT mode.

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

5-4. UUT Interface Section

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

Each line to the UUT contains a protection circuit. The protection circuit consists of a 38-ohm resistor in series with a pair of clipping diodes. This circuit prevents overvoltage or undervoltage conditions from damaging pod components. A 3-kilohm resistor in series with the inputs of the CMOS latches of the detection circuits limits the current to protect the internal protection diodes.

The detection circuits consist of latches connected through the 3 kilohm protection resistors to the UUT side of the 38-ohm resistors. The latches are clocked during a UUTON cycle, when the signals are expected to be at a known level. If a signal cannot be driven through the 38-ohm resistor, it will be detected when the latches are individually read by the Processor Section, and the values are compared with the expected values.

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

Also, anytime the UUT power supply drops below about 3.4V, all active pod outputs are disabled or written to their low logic level. This feature has been incorporated to protect UUT circuits from 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, or, if the power-fail error message has been disabled, the troubleshooter will indicate a pod timeout 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.

5-5. Timing and Control Section

The Timing and Control Section, shown in Figure 5-1, consists of a timer and internal timing and control logic. The Timing and Control Section receives inputs from the

Processor Section and the Interface Section. The bus switch is accomplished by buffer control signals and chip enable signals generated by the Timing and Control Section in response to inputs from the timer, the output latches as set by the microprocessor, the status lines from the UUT, and the control lines from the microprocessor.

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

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

During the time the pod is not communicating with the UUT, the UUT needs the proper signals so that it can perform dynamic memory refresh operations and other similar tasks. In order to provide these signals to the UUT, the pod performs what are called transparent reads. A transparent read is a read operation that is performed at the standby address. Transparent reads generate the transparent or fake control signals required to simulate a normal microprocessor read operation. This allows UUT operation even when the pod microprocessor is not communicating with the UUT.

5-6. DETAILED BLOCK DIAGRAM DESCRIPTION

A detailed block diagram of each major pod section is presented in Figure 5-2. Each major section is described in the following paragraphs along with a description of the Self Test circuit.

5-7. Detailed Description of the Processor Section

The microprocessor (U1), RAM (U2, U3), ROM (U4, U5), and address decoder (U8) form a small microprocessor system which is the heart of the pod.

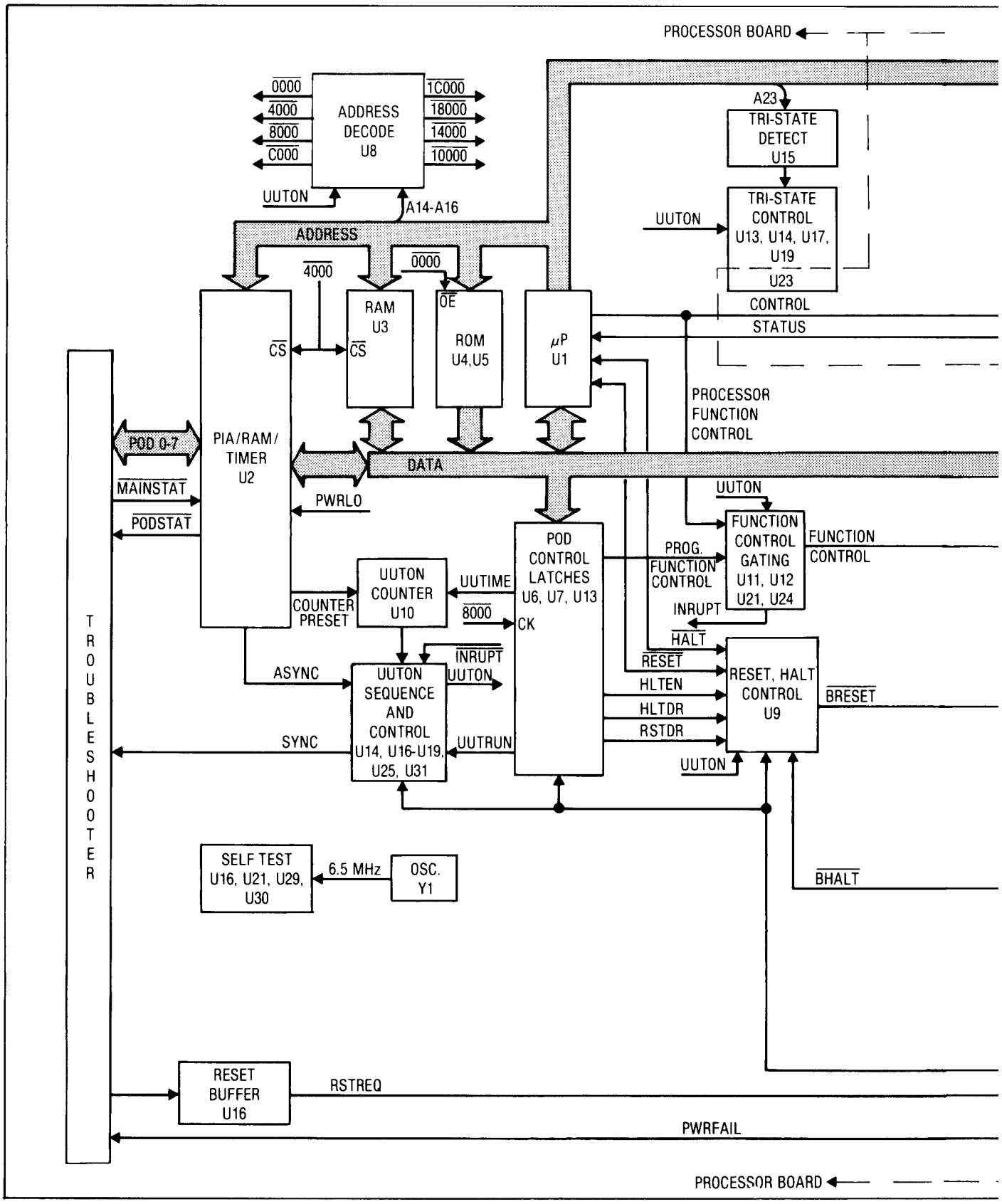
All communication between the pod and the troubleshooter uses the handshake protocol shown in Figure 5-3. The two handshake lines are $\overline{\text{MAINSTAT}}$ and $\overline{\text{PODSTAT}}$. $\overline{\text{MAINSTAT}}$ is driven by the troubleshooter and monitored by the pod. $\overline{\text{MAINSTAT}}$ initiates all data transactions and $\overline{\text{PODSTAT}}$ indicates the pod response.

All signals used for communication with the troubleshooter are routed through the I/O section of U2. U2 also contains the lower byte of RAM storage as well as a $\overline{\text{DTACK}}$ timer. This timer is used to restart the pod if the UUT does not respond with a $\overline{\text{DTACK}}$ in a specified amount of time. This response time is programmed using the $\overline{\text{DTACK}}$ delay special address. Outputs of U2 also provide inputs to Counter U10. U10 controls when the microprocessor will perform a UUT bus access, and selects either the interrupt or address/data synchronization output modes. The ASYNC line is high for address/data sync, low for interrupt sync.

When performing internal pod operations, the 68000 microprocessor is forced into the 6800 bus timing mode by holding the $\overline{\text{VPA}}$ processor input low. This is done to ease timing requirements on the RAM and ROM and to maintain compatibility with the bus requirements of U2.

5-8. Detailed Description of the UUT Interface Section

Devices A1 through A8 are Fluke-designed hybrid circuits containing current limiting resistors and clipping diodes to protect pod circuitry from overvoltage conditions.


PROCESSOR BOARD
0000
4000
8000
C000
ADDRESS DECODE U8
1C000
18000
14000
10000
UUTON
A14-A16
A23
TRI-STATE DETECT U15
TRI-STATE CONTROL U13, U14, U17, U19
U23
UUTON
ADDRESS
4000
0000
RAM U3
ROM U4,U5
μP U1
CONTROL
STATUS
CS
OE
PROCESSOR FUNCTION CONTROL
POD 0-7
PIA/RAM/ TIMER U2
DATA
MAINSTAT
PODSTAT
PWRLO
UUTON
FUNCTION CONTROL GATING U11, U12 U21, U24
FUNCTION CONTROL
PROG. FUNCTION CONTROL
POD CONTROL LATCHES U6, U7, U13
UUTON COUNTER U10
UUTIME
8000
CK
COUNTER PRESET
INRUPT
HALT
RESET
HLTEN
HLTDR
RSTDR
RESET, HALT CONTROL U9
BRESET
ASYNC
INRUPT
UUTON
UUTON SEQUENCE AND CONTROL U14, U16-U19, U25, U31
SYNC
UUTRUN
UUTON
TROUBLESHOOTER
SELF TEST U16, U21, U29, U30
6.5 MHz
OSC. Y1
BHALT
RESET BUFFER U16
RSTREQ
PWRFAIL
PROCESSOR BOARD

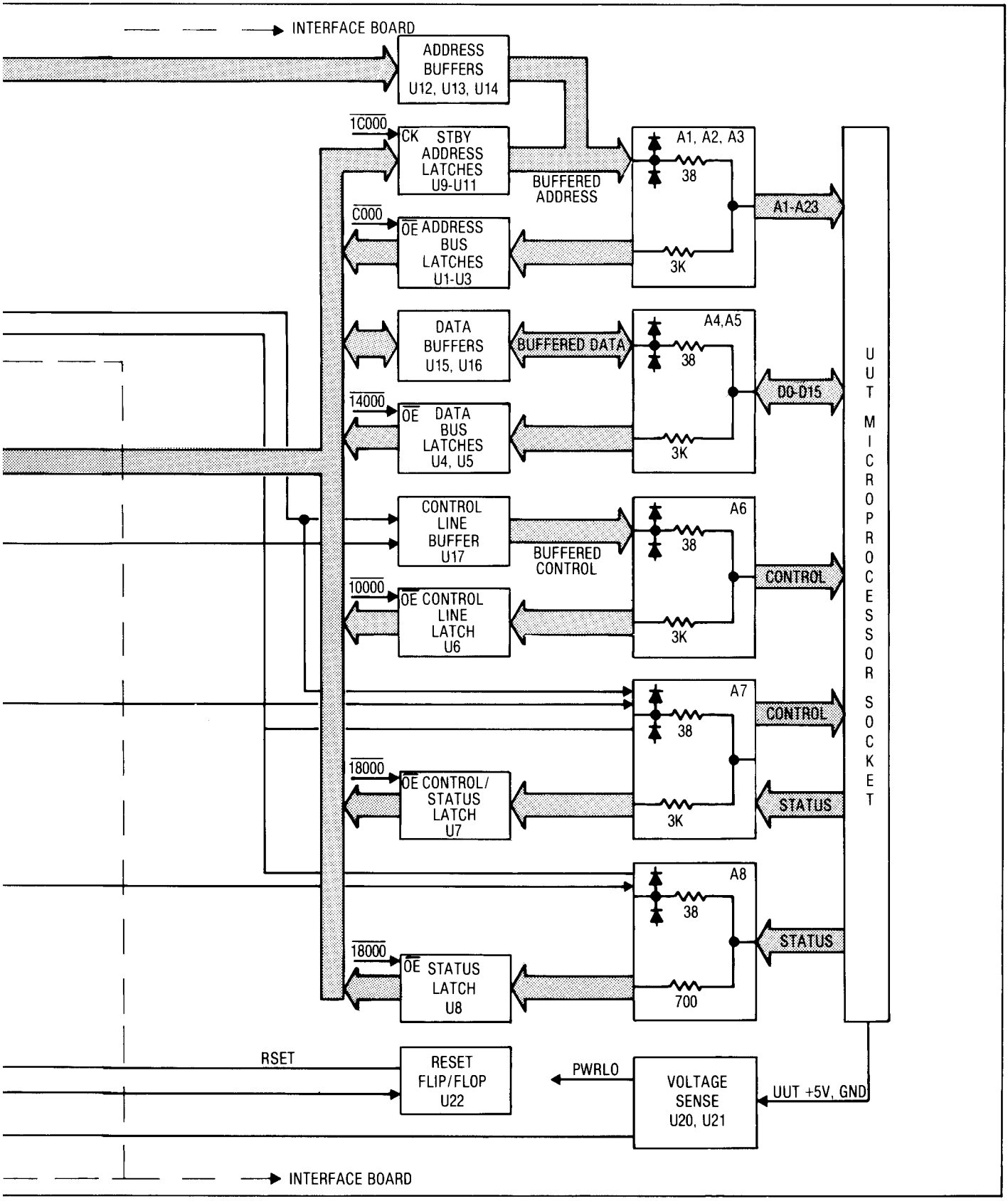


Figure 5-2. 68000 Pod Detailed Block Diagram

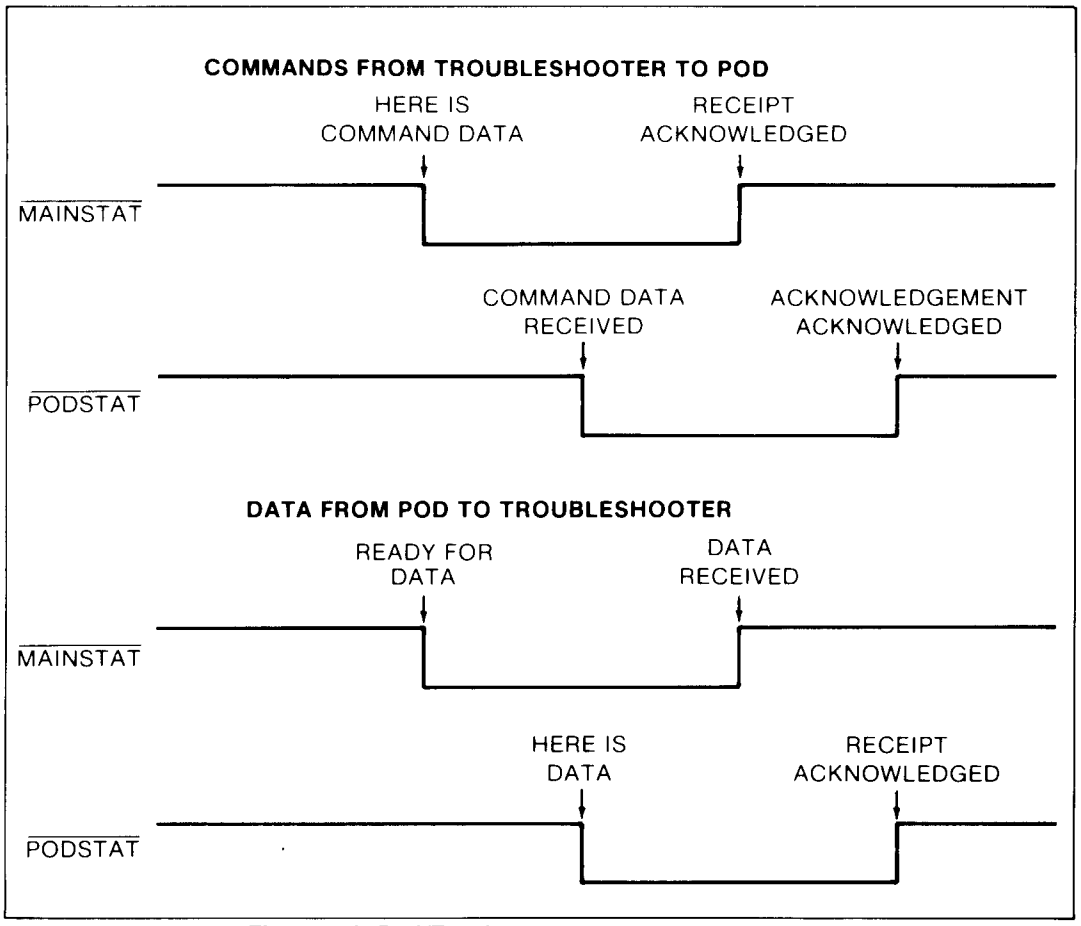


Figure 5-3. Pod/Troubleshooter Handshake Protocol

Latches U1 through U8 monitor the logic states of all of the microprocessor lines. The inputs to the latches are connected to the UUT lines through 3-kilohm resistors in the hybrid protection circuits (except inputs to U8, which are connected through 700-ohm resistors). The latches are clocked repetitively during the UUT access cycle. The last clock pulse occurs just before the end of the cycle and holds these logic states for later examination by the pod microprocessor.

Tri-state latches U9 through U11 are used to supply a known, fixed (standby) address to the UUT when the pod is not performing a UUT access. The upper 16 bits of this standby address are programmable through the Standby Address Special Address, while the lower 7 bits are always zero.

Flip-flop U19A is used to check activity on the $\overline{AS}$ line. Since the $\overline{AS}$ line is always low during a UUT access, U19A samples the state of the $\overline{AS}$ line just before it falls, to confirm that it is high, and thus not stuck low.

Exclusive-OR gate U18D and flip-flop U19B comprise an E line error detector. U18D compares the logic levels of the E line from the processor and at the UUT socket. If the logic levels are not the same, the error condition is latched and saved by U19B at the falling edge of the microprocessor clock.

Comparators U20, U21, and their associated circuits are used to monitor the UUT power supply voltage. U20 sends an error signal to the pod microprocessor and the

troubleshooter if the UUT power supply voltage is not between 4.5V and 5.5V. This signal indicates that the UUT power supply is operating improperly. To prevent possible damage to the UUT, U21 is used to inhibit all pod outputs from going high if the UUT power supply voltage falls below approximately 3.4V.

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

Output latches U6 and U7 are used to control various gating functions of the pod. The outputs of these latches are used to control which status inputs are enabled to the microprocessor, and which control output lines are asserted to the UUT during a Write Control operation. The latch outputs also determine the states of the Function control lines during pod standby and UUT access operations, and enable the timing of a UUT access.

Counter U10 and flip-flops U14A and U31B comprise the UUT access and sync timing logic. U10 is preset as long as the UUTIME signal is held low. When UUTIME goes high, the counter advances one count at the leading (high to low) transition of each assertion of the microprocessor $\overline{AS}$ signal. When the counter presents a carry out to the J input of U14A, the trailing edge of the $\overline{AS}$ signal causes the assertion of PREUUTON which gates a sync signal to the troubleshooter.

One-half clock cycle later, the UUTON signal is asserted by U31B, causing the standby address drivers to be tri-stated, and enabling the microprocessor address, data, and control buffers to drive the UUT. These buffers remain enabled until one-half clock cycle after the trailing edge of the subsequent $\overline{AS}$ assertion unless the UUTRUN signal has also been asserted. If UUTIME and UUTRUN are asserted simultaneously, the UUTON signal will remain asserted until the pod is reset. This is done to allow the Run UUT operation.

The pod will also enter the UUTON state if the microprocessor executes an interrupt acknowledge bus cycle. If the INTEN signal is written high through U13A, gates U28A, U28B, and U28C allow interrupt inputs to the microprocessor. If the microprocessor acknowleges a subsequent interrupt request, microprocessor Function Control lines FC0 through FC2 will go high. This condition is recognized by U21A and causes the function control bits to be allowed to the UUT via multiplexers U11 and U12, and also sets flip-flop U14A causing a UUTON bus cycle. The low byte data drivers are not enabled during this cycle, however, so the microprocessor always sees an interrupt vector of FF, since these data lines are pulled to the logic 1 state through resistor networks Z4 and Z5 (on the interface board). As soon as flip-flop U14A is set by an interrupt, flip-flop U13A is cleared, disabling any further interrupts.

Interface board flip-flops U23A and U23B and microprocessor board flip-flops U14B and U13B determine when the microprocessor tri-states the address, data, and control lines due to a halt or bus request. The U23 flip-flops ensure that the $\overline{HALT}$ or $\overline{BR}$ inputs are asserted early enough to be recognized by the microprocessor before the end of a bus cycle, and thus for the bus to be tri-stated before the next bus cycle starts. U14 delays the recognition of the bus tri-state request until the end of the current bus cycle; if no bus cycle is in progress, U14 allows a bus grant from the microprocessor to request the bus tri-state condition. U13B synchronizes the timing of the bus tri-state enabling and disabling with the timing of the microprocessor. Note that this logic does not tri-state the address bus at the beginning of each bus cycle.

Comparator U15 and its associated circuitry is used to detect the address bus tri-state condition. If it is crucial for the address bus circuitry to be in a tri-state condition at the beginning of each bus cycle, this tri-state detection circuitry may be used to tri-state the address bus drivers in addition to the normal logic (as described in the previous

paragraph). The tri-state detection circuitry is not normally used because the detection circuit delay (approximately 25 ns at the start of tri-state and 20 ns at the end of tri-state) reduces address settling time at the UUT to an unacceptably low value when using a fast microprocessor clock (8-10 MHz).

U9 is a programmed gate array which drives the bi-directional, open collector $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ lines. Figure 8-1 is a schematic of the programmed array. U9 is used to arbitrate which direction and under what conditions inputs to and outputs from the microprocessor are allowed to and from the UUT. The gating functions are summarized as follows:

- $\overline{\text{RESET}}$ is asserted from the pod to the UUT when either of the following conditions exist:

  1. When the 68000 is asserting $\overline{\text{RESET}}$.

  2. When RSTDR is asserted and UUTON is true (used for the Write Control function).

- $\overline{\text{RESET}}$ is inhibited from the pod to the UUT when the troubleshooter is resetting the pod.

- $\overline{\text{RESET}}$ is asserted from the UUT to the pod only when the UUT is asserting $\overline{\text{RESET}}$ during Run UUT.

- $\overline{\text{HALT}}$ is asserted from the pod to the UUT when either of the following conditions exist:

  1. The 68000 is asserting $\overline{\text{HALT}}$.

  2. HLTDR is asserted and UUTON is true (used for the Write Control function).

- $\overline{\text{HALT}}$ is inhibited from the pod to the UUT when the troubleshooter is resetting the pod.

- $\overline{\text{HALT}}$ is asserted from the UUT to the pod when either of the following conditions exist:

  1. HLTEN is true.

  2. The UUT is asserting $\overline{\text{HALT}}$ during Run UUT.

Both $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ are asserted to the pod microprocessor when the pod is reset by the troubleshooter.

Flip-flop U22B is used to delay resets from the troubleshooter until the end of an internal pod bus cycle so that the pod is not reset in the middle of a write operation to RAM, which might destroy valuable data.

Flip-flop U22A and gate U26 condition the E clock signal that goes to PIA/RAM U2. The clock signal going to U2 must not rise when the address inputs to it are changing, a condition which may occur immediately after a UUT access or during Run UUT. (Memory errors may occur regardless of the state of the chip enable inputs.) Therefore, the clock signal to U2 is not allowed to occur if $\overline{\text{AS}}$ is high or if the pod is in Run UUT.

Timing diagrams for typical 68000 cycle, read and write operations are shown in Figures 5-4, 5-5 and 5-6.

5-10. Detailed Description of the Self Test Circuit

During self test, the pod outputs are connected to pod inputs, causing the pod to appear to the troubleshooter to be a small UUT with well defined characteristics. The troubleshooter performs a series of operations and compares the expected behavior with the actual behavior to determine whether or not the pod is operating properly.

When the UUT cable is plugged into the self test socket, pin 16 is pulled to a low logic level. The logic state of pin 16 is checked as part of every UUT access and when low, indicates that the pod should execute the self test routines. This information is also sent to the troubleshooter which requests several pod operations as part of the self test procedure.

Y1 is an oscillator which provides the microprocessor with a 6.5 MHz clock signal during pod self test.

Resistor networks Z1 and Z2 allow the microprocessor to read the lower 15 bits of the address lines when performing read operations, and to test the drive capability of the lower 15 address bits when performing write operations. Tri-state drivers U29 and U30 provide the ability to read the state of most of the control lines and the upper 8 address bits. The status lines are driven through resistors R14, R15, and R16 to test their drivability and gating to the microprocessor.

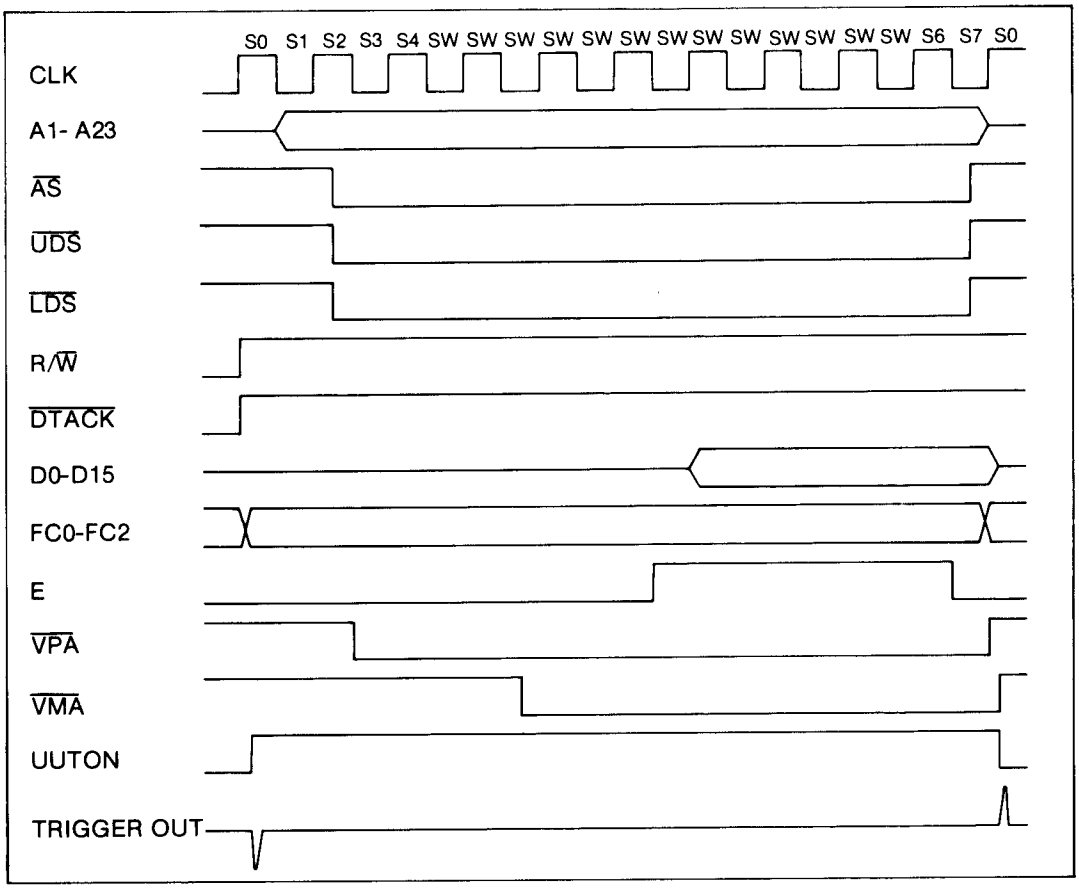


Figure 5-4. 68000 Cycle Operation

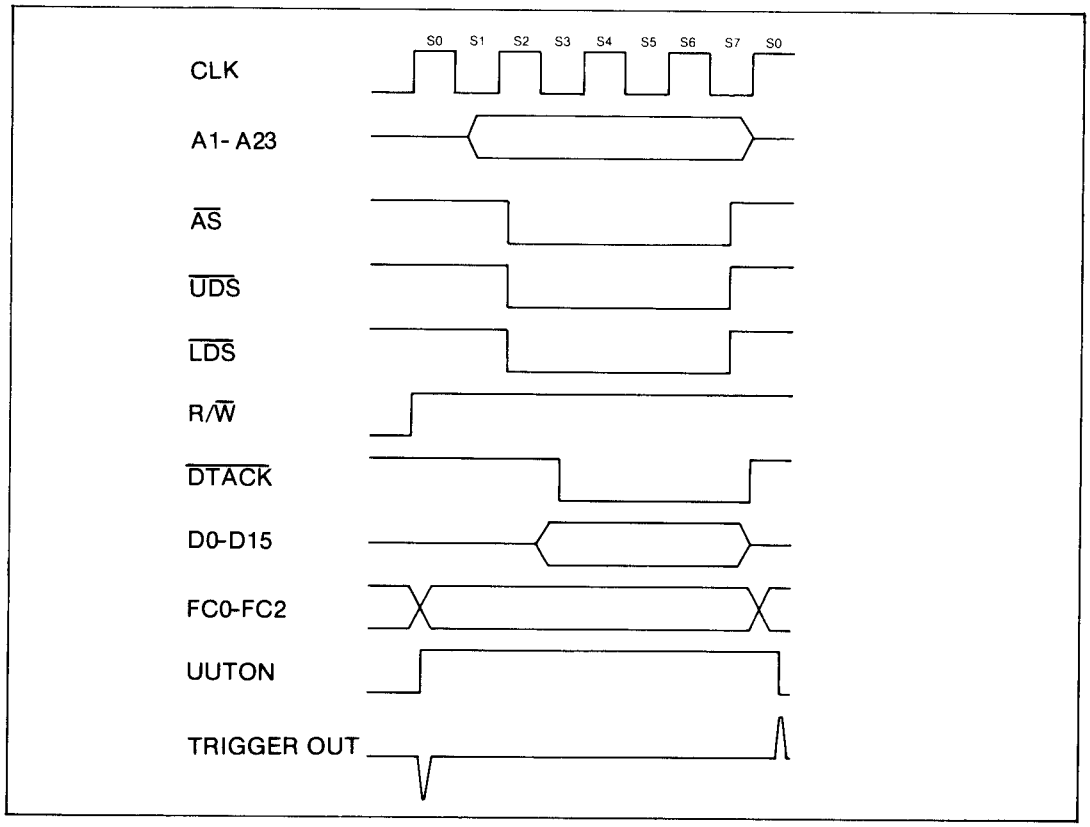


Figure 5-5. 68000 Read Timing (No Wait States)

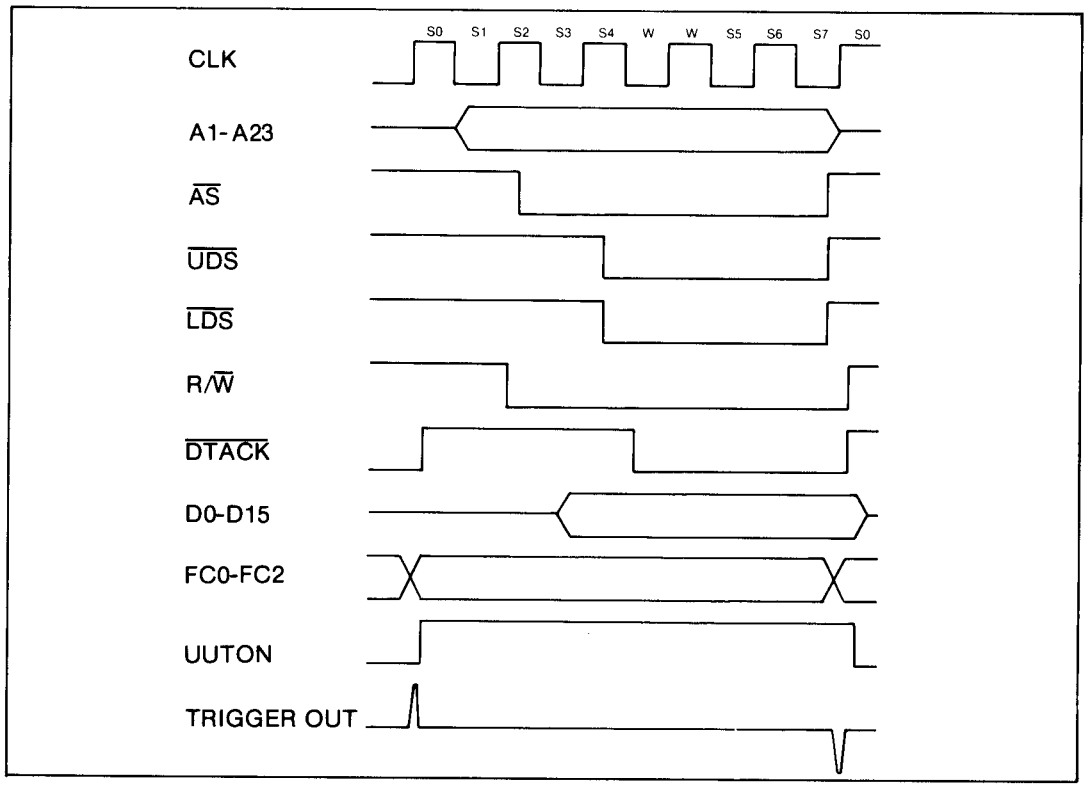


Figure 5-6. 68000 Write Timing (2 Wait States)

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. Test Equipment Required for Pod Troubleshooting

EQUIPMENT TYPE	REQUIRED TYPE
Micro System Troubleshooter	Fluke 9000 Series
Interface Pod	Fluke 9000A-68000
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 68000 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

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 68000 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 68000 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 F000 0000 (F000 0000 is not an address within the normal address space of the 68000, 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 @ F000 0000 = 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 F000 0000 is anything other than *READ @ F000 0000 = FF OK,* then the pod failed the enhanced self test. Proceed to the next section titled The Enhanced Self Test for appropriate troubleshooting procedures.

6-6. The Enhanced Self Test

The enhanced self test consists of 8 self test routines that are performed on the pod circuitry. The test routines are listed and described in Table 6-2a.

Whenever the pod fails the enhanced self test, the pod causes the troubleshooter to display the message *POD SELF TEST 68000 FAIL 00.* To confirm that the enhanced self test failed, perform a read operation at address F000 0000. If the troubleshooter displays any message other than *READ @ F000 0000 = 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.

Table 6-2a. Enhanced Pod Self Test Failure Codes

TEST ROUTINE/ FAILURE CODE	POD OPERATION	OPERATOR ACTIONS TO RECREATE TEST
0001	Internal pod RAM read/write test	Operator cannot duplicate test
0002	Internal pod ROM checksum test	Operator cannot duplicate test
0003	READ @ 2AA AAAA Expected data: ABAA	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0016 = 5555 (Set standby address) WRITE @ F000 0018 = 5 (Set standby function code) READ @ 12AA AAAA (Quick looping read)
0004	READ @ 555 5554 Expected data: 5455	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0016 = AAAA (Set standby address) WRITE @ F000 0018 = 2 (Set standby function code) READ @ 1555 5554 (Quick looping read)
0005 (a)	READ @ D55 5555 Expected data: 0055	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0016 = AAAA (Set standby address) WRITE @ F000 0018 = 4 (Set standby function code) READ @ 1D55 5555 (Quick looping read)
0005 (b)	$\overline{BG}$ drivability test	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0018 = 3 (Set standby function code) WRITE @ F000 001A = 5555 (Set default address) WRITE @ F000 001C = 5 (Set default function code) WRITE @ CNTL = 11011 LOOP ($\overline{BG}$ drivability test)
0006	READ @ AAA AAAA Expected data: 0072	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0016 = 5555 (Set standby address) WRITE @ F000 0018 = 4 (Set standby function code) READ @ 1AAA AAAA (Quick looping read)
0007 (a)	WRITE @ 055 5554 = AAAA	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0014 = 0 (Set standard function code) WRITE @ F000 0016 = 5555 (Set standby address) WRITE @ F000 0018 = 6 (Set standby function code) WRITE @ 1055 5554 = AAAA (Quick looping write)
0007 (b)	$\overline{HALT}$ drivability test	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0018 = 3 (Set standard function code) WRITE @ F000 001A = 5555 (Set default address) WRITE @ F000 001C = 0 (Set default function code) WRITE @ CNTL = 11110 LOOP ($\overline{HALT}$ drivability test)
0008 (a)	WRITE @ 6AA AAAA = 5555	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0016 = AAAA (Set standby address) WRITE @ F000 0018 = 1 (Set standby function code) WRITE @ 16AA AAAA = 5555 (Quick looping write)
0008 (b)	$\overline{RESET}$ drivability test	WRITE @ F000 0010 = 2 (Set DTACK delay) WRITE @ F000 0018 = 3 (Set standby function code) WRITE @ F000 001A = AAAA (Set default address) WRITE @ F000 001C = 6 (Set default function code) WRITE @ CNTL = 01111 LOOP ($\overline{RESET}$ drivability test)

For example, assume the pod self test is performed and the troubleshooter displays the message *POD SELF TEST 68000 FAIL 00.* The following sequence of keystrokes provides information about the pod failure detected:

PRESS	DISPLAY	COMMENT
READ F000 0000 ENTER	*ADDR ERR @ F000 0000-LOOP?*	The enhanced self test discovered an error.
MORE	*ADDR BITS 10 0000-LOOP?*	Address bit 20 may be incorrect. (The hexadecimal 100000 in the display message corresponds to bit 20).
CONT	*DATA ERR @ F000 0000-LOOP?*	A possible data line failure was also discovered.
MORE	*DATA BITS 0010-LOOP?*	Data bit 4 may be incorrect.
CONT	*READ @ F000 0000 =0003 FAIL*	The failure occurred during test routine 3.

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 @ F000 0000; the troubleshooter does not loop on the test routine in which the failure occurred.

The previous sequence of messages indicates that address bit 20 and data bit 4 were found to be in error while the pod was performing test routine 3 (as described in Table 6-2a). Note that although data and address errors usually imply a drivability error, when encountered with the enhanced self test these errors indicate 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 F000 0000 ENTER	*ACTIVE FORCE LINE @ F000 0000-LOOP?*
MORE	*STS BITS 0000 0000 0000 0001-LOOP?*
CONT	*READ @ F000 0000 = 0004 FAIL*

The previous sequence of messages indicates that status bit 0 ($\overline{\text{HALT}}$) 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 the forcing status 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.

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

4. Press the Bus Test key on the troubleshooter to initiate the Self Test, then press the Stop Key. Perform a *WRITE @ F000 0028 = BF;* this disables the pod Self Test by preventing the pod from reporting to the troubleshooter that it is plugged into the Self Test socket (the pod Self Test may be re-enabled by cycling pod power off and then on, or by a *WRITE @ F000 0028 = FF).*

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

   SET-TRAP BAD POWER SUPPLY? YES

   SET-TRAP ILLEGAL ADDRESS? YES

   SET-TRAP ACTIVE INTERRUPT? NO

   SET-TRAP ACTIVE FORCE LINE? NO

   SET-TRAP CTL ERR? YES

   SET-TRAP ADDR ERR? YES

   SET-TRAP DATA ERR? YES

   SET-ENABLE HALT? NO

   SET-ENABLE BR/ACK? NO

   SET-ENABLE INTR? NO

When the pod and the troubleshooter are connected in this configuration (with *SPECIAL ADDRESS F000 0028 = BF),* 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 re-enable the pod self test, you must *WRITE @ F000 0028 = FF* or perform a power-on reset.

6-8. Recreating the Enhanced Self Test Routines

The information provided in Table 6-2a describes the operations the pod performs for each test routine and describes actions the operator may take to recreate the test routines. By recreating the test routine in which the pod failure was detected, the operator may then

use a synchronized probe or an oscilloscope to trace the cause of the failure. Test routines 5, 7, and 8 in Table 6-2a perform control line drivability tests simultaneously with read or write operations. This cannot be duplicated from the troubleshooter keyboard. Thus, the read portion of the test is described by routines 5a, 7a, and 8a while routines 5b, 7b, and 8b describe the write control line portion of the routine.

Consider the first example in the preceding section where the pod detected that address bit 20 and/or data bit 4 had the incorrect value during the execution of test routine 3. 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 disabling self test reporting by *WRITE @ F000 0028 = BF).* Then press the following sequence of keystrokes to recreate test routine 3 (as described in Table 6-2a):

PRESS	DISPLAY	COMMENT
WRITE F000 0010 ENTER 2 ENTER	*WRITE @ F000 0010 = 2 OK*	Sets DTACK delay to 20 clock cycles
WRITE F000 0016 ENTER 5555 ENTER	*WRITE @ F000 0016=5555 OK*	Sets standby address = 555500.
WRITE F000 0018 ENTER 5 ENTER	*WRITE @ F000 0018=5 OK*	Sets standby function code = 101.
READ 12AA AAAA ENTER LOOP	*READ @ 12AA AAAA=ABBA OK*	Quick looping read shows wrong answer. Expected data is ABAA. Data bit 4 is high in error.

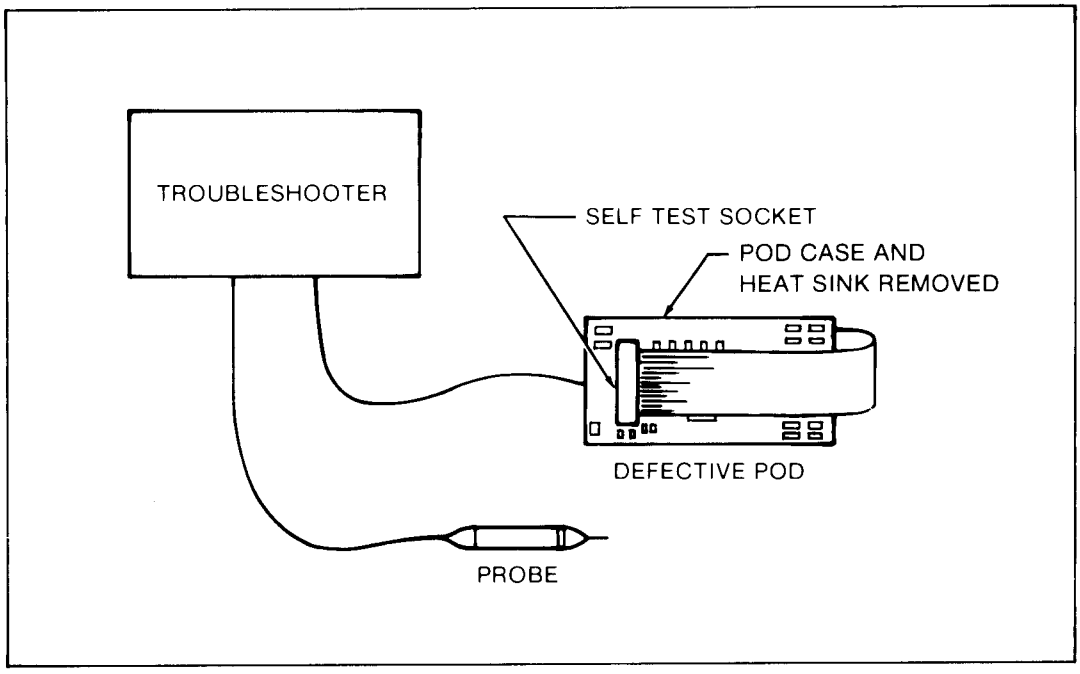


Figure 6-1. Troubleshooting a Defective Pod

Since you have confirmed that data bit 4 is in error, set the probe to the address synchronization mode and repeat the read command. While the troubleshooter continues to quick loop on the read at address 2AA AAAA, refer to the processor board (A31) schematic diagram in Section 8 and trace the circuit logic to find where the data is in error. Note that the self test socket is wired so that address bit 20 drives data bit 4 through a 2.0 kilohm resistor (unless standby function code bits 0 and 1, named FC00F and FC10F, are both low. In this example, for instance, an open line (A20) on the UUT connector cable causes data bit 4 to appear to be stuck at the high logic level.

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

NOTE

When the pod detects a self test error, it reports as errors all signal lines which deviate from their expected values, and also other lines which, if faulty, might cause the detected error(s). Use the self test socket diagram and knowledge of the 68000 processor operation to determine where to start troubleshooting if multiple errors are reported. For instance: if the pod $\overline{VPA}$ input is tied low, the pod will report the following errors for expanded self test 3; Forcing line (status) errors $\overline{HALT}$, $\overline{VPA}$, $\overline{IPL0}$, and NEITHER $\overline{DTACK}$ OR $\overline{VPA}$ ASSERTED, plus a $\overline{VMA}$ control line error. Recreating test 3 manually results in no errors reported, since it is not an error condition if all three of these status inputs are low. Examination of the schematic diagram shows that the $\overline{HALT}$, $\overline{VPA}$, and $\overline{IPL0}$ lines are connected together through the self test socket, so a short on any of these lines will affect the others. In addition, during test 3 neither the $\overline{DTACK}$ or $\overline{VPA}$ lines are normally asserted. With $\overline{VPA}$ low, the NEITHER $\overline{DTACK}$ OR $\overline{VPA}$ ASSERTED status is incorrect and is thus flagged as an error. In addition, the $\overline{VMA}$ line is normally high during test 3, but is driven low by the processor in response to the erroneous $\overline{VPA}$ input, and so is abnormal for expanded self test 3. To find the problem, unplug the UUT connector from the self test socket and examine the three status lines separately to find the shorted input.

As a troubleshooting aid, Table 6-2b shows the expected logic levels at each of the pod input latches. The logic levels saved by these latches during the pod test operation are interrogated by the self test program to determine where failures, if any, have occured.

For example: If while looping on the failure cited above, examination of the data bit 4 line at the self-test socket and at interface board U5 (data low latch) pin 17 shows the correct logic low level during the address sync period, it is likely that latch U5 has failed, and is passing incorrect data to the processor, causing the self-test failure.

Similarly, the other expected levels shown can be examined to determine the source of the pod fault.

Also shown in Table 6-2b are the expected outputs of the pod control latches (U6 and U7 on the processor board) and the outputs of the standby address latches. The standby adress latch outputs are tri-stated during the test address sync period and should be checked 1 to 10 microseconds after the end of the address sync period.

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.

If you correct a problem while using the procedures provided in this section, try the pod self test again. If the troubleshooter again displays an *ATTEMPTING RESET* message, continue with the procedures in this section. However, if the troubleshooter displays the message *POD SELF TEST 68000 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.)

Table 6-2b. Enhanced Self Test Latch Logic Levels

RECREATED TEST ROUTINE	INPUT LATCHES								OUTPUT LATCHES				
	ADDRESS			DATA		CNTL	STATUS	MIXED	POD CONTROL		STANDBY ADDRESS		
	HIGH U1*	MED U2*	LOW U3*	HIGH U4*	LOW U5*	U6*	U8*	U7*	U6†	U7†	HIGH U10**	MED U9**	LOW U11**
0003	AA	AA	AB	AB	AA	45	6B	78	98	09	55	55	00
0004	55	55	55	54	55	4A	94	19	A8	06	AA	AA	00
0005 (a)	55	55	55	A5	55	6A	44	79	B8	04	AA	AA	00
0005 (b)	55	55	00	00	55	4B	BD	01	AB	47	AA	AA	00
0006	AA	AA	AB	72	AA	54	42	79	98	08	55	55	00
0007 (a)	55	55	54	AA	AA	00	D6	79	98	02	55	55	00
0007 (b)	55	55	00	00	55	40	94	19	88	13	55	55	00
0008 (a)	AA	AA	AB	55	55	0D	29	09	A8	01	AA	AA	00
0008 (b)	AA	AA	00	01	AA	4D	29	09	A8	8B	AA	AA	00

* Located on Interface board—Pin Sequence: 3, 18, 4, 17, 7, 14, 8, 13
** Located on Interface board—Pin Sequence: 2, 19, 5, 16, 6, 15, 9, 12 (valid only after end of address sync period)
† Located on Processor board—Pin Sequence: 19, 16, 15, 12, 9, 6, 5, 2

Table 6-3. Standard Pod Self Test Failure Codes

TEST ROUTINE/ FAILURE CODE	POD OPERATION	OPERATOR ACTIONS TO RECREATE TEST
00*	Reset pod READ @ 0FF0 0FF0 Expected data: F00F	WRITE @ F000 0016 = 0000 (set standby address) READ @ 07F0 0FF0 (looping read) If a power fail error message occurs, check the power detection circuits.
01	WRITE @ 0FF0 0FF0 = 0FF0	WRITE @ F000 0016 = 0000 (set standby address) WRITE @ 07F0 0FF0 = 0FF0 (looping write)
02	Test control line	Perform a Bus Test and observe the error message. If the error involves a writable 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.
03	Send command to enable all pod enable lines and verify that a pod timeout occurs. This timeout is transparent to the operator)	WRITE @ F000 0018 = 2 (set standby function code) 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 pine of the microprocessor.

*If the self test message is POD SELF TEST 68000 FAIL 00, the enhanced self test may have failed. To check, READ @ F000 0000: If the resulting message is READ @ F000 0000 = 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 @ F000 00000 = FF OK, the pod failed the enhanced self test; refer to the previous section titled The Enhanced Self Test.

6-12. Preparation for Troubleshooting an Inoperative Pod

An inoperative pod is like any other microprocessor-based UUT that is not operating properly; the easiest way to fix an inoperative pod is by using a troubleshooter and a good pod. 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. If a second troubleshooter is available, connect the pod cable plug from the inoperative pod to the second troubleshooter to supply the inoperative pod with power. If a second troubleshooter is not available, connect a +5V dc (2 amp) and a -5V dc (200 mA) power supply to the pod as shown in Figure 6-2. An easy place to make the power connections is at the connector that usually connects the cable to the troubleshooter.

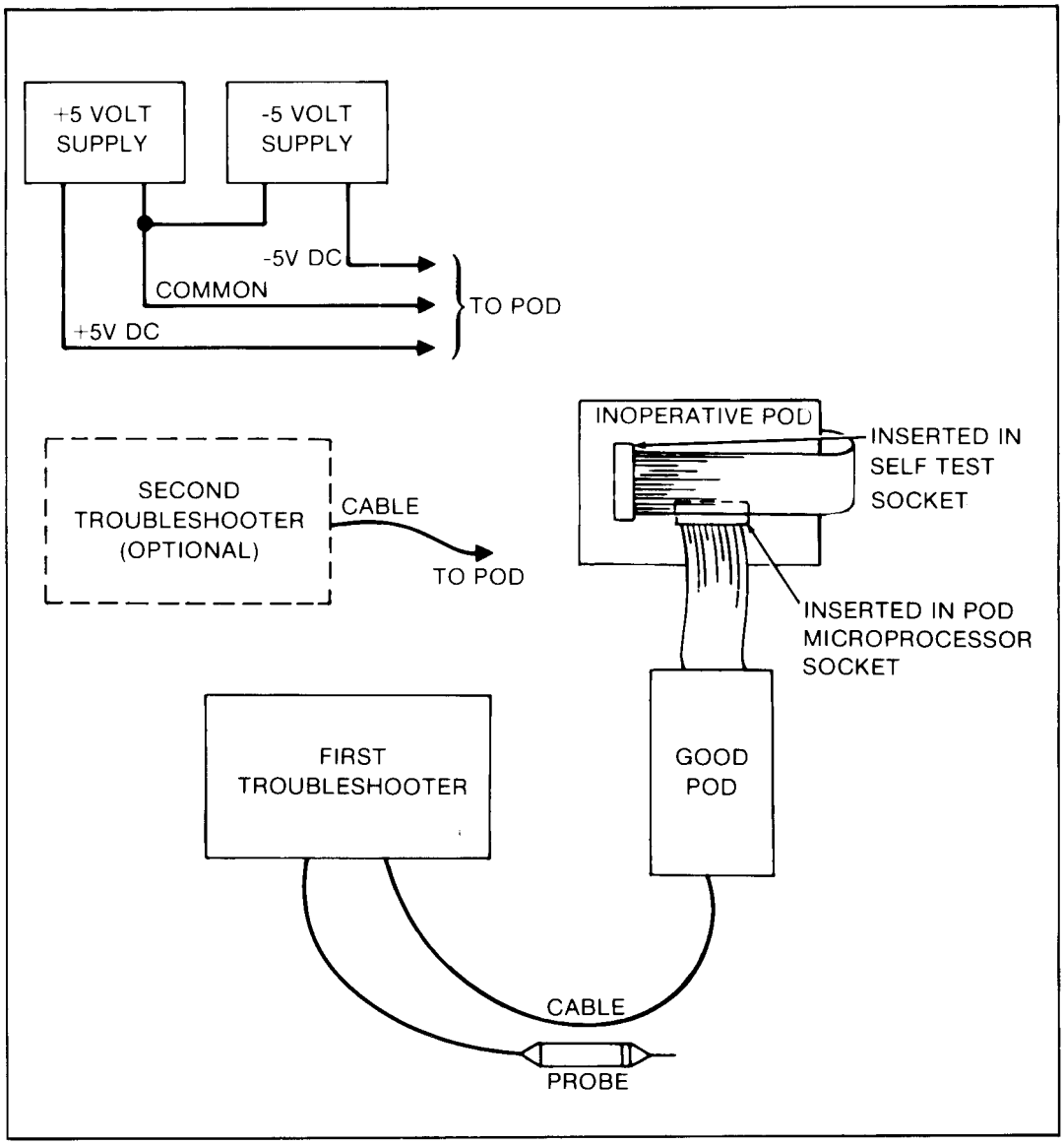


Figure 6-2. Troubleshooting an Inoperative Pod

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

6. Connect pins 2 and 15 to +5V, pin 21 to –5V, and pin 25 to ground.

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

CAUTION

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

CAUTION

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

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

2. Check that the microprocessor clock signal is present at pin 17 of protection circuit A7. (Use the terminal at the lower right corner of the board as a ground reference.)

3. Perform a *READ @ 0* operation with the troubleshooter (this causes a pod reset, which then brings setup information from the pod to the troubleshooter), then select the troubleshooter setup function and disable all the enableable lines of the good pod (HALT, BR/ACK, INTR).

4. Check the signal levels at U20 pin 11 (PWRFAIL) and U21 pin 7 (PWRLO). With proper self test socket (or UUT) voltages, both lines should be low.

5. Verify clock activity at U22 pin 11 ($\overline{E}$). This is the complement of the E clock signal from the microprocessor and should always be present.

6. Position the inoperative pod so that the processor board is facing up. Reset the pod by shorting pin 13 of U16 on the processor board ($\overline{MRSET}$) to ground. (A ground test point is located next to pin 1 of U29, near the left bottom corner of the pcb.) Verify that the $\overline{RESET}$ and $\overline{HALT}$ lines go low at the processor socket by performing a Looping Read Status operation with the troubleshooter mainframe. While shorting pin 13, the mainframe should display the following status bits:

0000 0111 0100 1110

After removing the short the status word should be:

0000 0111 0101 1111

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

NOTE

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

7. Verify that U31 pin 9 is a logic low (UUTON), and that U31 pin 8 is a logic high ($\overline{\text{UUTON}}$). After a pod reset, these lines should not change state until the pod access timer, U10, is enabled.

8. Check the pod access timer and UUTON logic. Set U10 pin 9 high by performing a *WRITE @ 8000 = 8000*. Next, set U10 inputs A-D, pins 3-6, low by performing a *WRITE @ 4106 = 5F*. Verify that U10 inputs A, B, C, and D are low. (The pod UUT connector cable may be removed temporarily from the self test socket to facilitate access.) If the U10 inputs are not low, go on to step 9. Check U10 pin 15 to monitor the counter carry output which should be pulsing high (the UUT connector cable must be plugged into the UUT or self test socket). Also check U31 pin 9, which should be high momentarily after each carry output.

9. Check the address decoder (U8) by performing looping read operations at the following addresses and noting that the corresponding outputs of U8 go low:

Read Address	U8 Output pin
0	15
4000	14
C000	12
1 0000	11
1 4000	10
1 8000	9
1 C000	7

10. Reset the pod (ground U16 pin 13 momentarily), and perform RAM short and RAM long tests at addresses 4000 - 40FE.

11. Perform a ROM test at addresses 0 - 1FFC. The expected signature can be read at location 1FFE (the last word of ROM).

NOTE

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

12. Check the output operation of I/O port A (part of U2) as follows:

a. Perform the operation *WRITE @ 4102 = FF*. This operation writes to the port A direction register and configures all the lines of I/O port A (PA0 through PA7) as outputs.

b. Perform the operation *WRITE @ 4100 = FF*. This operation sets all the port A outputs high.

c. Check the I/O port A lines (pins 8 - 15 of U2) with the probe or scope to confirm that all the levels are high. (The pod UUT connector may be temporarily removed from the self test socket.)

d. Repeat step b with 00 as the data.

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

13. Check the input operation of I/O port A as follows:

a. Select the address sync mode and turn-on the low pulse key of the troubleshooter.

b. Perform the operation *WRITE @ 4102 = 00*. This operation writes to the port A direction register and configures all the lines of I/O port A as inputs.

c. If the defective pod is getting its processor clock from the self test socket, remove the pod UUT connector from the self test socket and connect a jumper from pin 15 of the pod UUT plug to pin 15 of the self test socket. This is necessary to allow access to pins of U2 which are difficult to reach when the UUT connector is plugged into the self test socket. The pod may be left in this configuration for the remainder of the tests.

d. Perform a looping read operation at address 4100, the address of the I/O port A data register, while sequentially applying the probe to each of the I/O port A lines (pins 8 - 15 of U2). Observe the troubleshooter display and check that each input bit toggles as it is probed (only the lower 8 bits are affected).

14. Check the output operation of I/O port B (part of U2) as follows:

a. Perform the operation *WRITE @ 4106 = 5F*. This operation writes to the port B direction register and sets lines PB6 and PB4 thru PB0 of I/O port B as outputs.

b. Perform the operation *WRITE @ 4104 = 5F*. This operation sets the selected port B outputs high.

c. Check the I/O port B output lines (pins 17, 19, 21 - 24 of U2) with the probe or scope to confirm that all the levels are high.

d. Repeat step b with 00 as the data.

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

15. Check the input operation of I/O port B as follows:

a. Select the address sync mode and turn-on the low pulse key of the troubleshooter.

b. Perform the operation *WRITE @ 4106 = 5F*. This operation writes to the port B direction register and configures lines PB7 and PB5 of I/O port B as inputs.

c. Perform a looping read operation at address 4104, the address of the I/O port B data register, and apply the probe to pin 16 of U2 (PB7). Data bit 7 of the troubleshooter display should toggle (the display should change from *FFFF* to *FF7F).* Similarly, data bit 5 should toggle when the probe is applied to pin 18 of U2 (PB5). (See paragraph 13-c if you can't reach the pins of U2).

16. If repairs have been made to the inoperative pod as a result of the preceeding steps, try the pod self test again. If the message displayed is *POD SELF TEST 68000 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 the R/W and power supply lines. If the pod passes self test, but the troubleshooter displays the message *POD TIMEOUT* when the pod is connected to a UUT, check the clock output of the UUT.

6-16. 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 special address location F000 0028 = BF). If a line is found that does not cause the pod to timeout, then the logic controlling the enabling of that line should be examined.

6-17. Timing Problems

These problems are usually caused by components that are still functioning, but are not functioning within the allowable specifications. The best way to check this problem is to look at suspected signals using an oscilloscope synchronized to valid addresses. Look for slow rise or fall times or signals driven to marginal logic levels. If the part is too slow, it might fail in the UUT, but pass the pod self test because the pod clock rate is somewhat slower. The clock rate at the self test socket is approximately 6.5 MHz.

6-18. 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.

6-19. 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. Remove the four phillips screws that connect the heat sink to the processor pcb assembly and remove the heatsink.

NOTE

The heat sink is not needed for heat dissipation unless the pod is fully assembled. If the two pcb assemblies are removed from the top and bottom cases, the heat sink may be removed during pod operation and troubleshooting.

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

NOTE

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

5. 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.

Table 7-1. 9000A-68000 Final Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
	Ⓟ FINAL ASSEMBLY, 68000 POD FIGURE 7-1 (9000A-68000-5001)						
A31Ⓟ	PROCESSOR PCB ASSEMBLY	609149	89536	651562	1		
A32Ⓟ	INTERFACE PCB ASSEMBLY	609180	89536	651588	1		
H1	SCREW, PHP, SEMS, 4-40 X 1/4	185918	89536	185918	4		
H2	SCREW, PHP, 4-40 X 3/4	115063	89536	115063	4		
H3	SCREW, RHP, 4-40 X 5/16	149773	89536	149773	4		
H4	SCREW, PHP, 4-40 X 5/8	145813	89536	145813	1		
H5	WASHER, INT TOOTH, #4	110403	89536	110403	1		
MP1	ACTUATOR	582916	89536	582916	1		
MP2	CLIP, HEATSINK, 20-PIN	607671	89536	607671	1		
MP3	CLIP, HEATSINK, 24-PIN	607655	89536	607655	3		
MP4	CLIP, HEATSINK, 40-PIN	609230	89536	609230	1		
MP5	CLIP, HEATSINK, 64-PIN	609198	89536	609198	2		
MP6	DECAL, POD	685982	89536	685982	1		
MP7	DECAL, SPEC	685990	89536	685990	1		
MP8	DECAL, WARNING	659805	89536	659805	1		
MP9	HEATSINK	654483	89536	654483	1		
MP10	LABEL, STATIC CAUTION	605808	89536	605808	1		
MP11	LABEL, UUT CAUTION	634030	89536	634030	1		
MP12	SHELL, BOTTOM	648881	89536	648881	1		
MP13	SHELL, TOP	653196	89536	653196	1		
MP14	SHIELD	659789	89536	659789	1		
MP16	SPACER, HEX, 4-40 X 3/8	187575	89536	187575	1		
TM1	INSTRUCTION MANUAL, 9000A-68000	652594	89536	652594			
U1Ⓟ	IC, NMOS, 68000, 16-BIT MICROPROCESSOR	650499	04713	MC68000L10	1	1	
U2Ⓟ	IC, NMOS, RAM, I/O TIMER	536417	55576	SYP6532A	1	1	
U3Ⓟ	IC, NMOS, 128K X 8-BIT STATIC RAM	586016	07263	F68B10CP	1	1	
U4Ⓟ	IC, EPROM, PROGRAMMED	579516	89536	579516	1	1	
U5Ⓟ	IC, EPROM, PROGRAMMED	579532	89536	579532	1	1	
U9	IC, PAL, PROGRAMMED	579490	89536	579490	1	1	
W1	CABLE, POD W/RESISTOR	607184	89536	607184	1		
W2	CABLE, UUT	685610	89536	685610	1		
	RECOMMENDED SPARE PARTS KIT, 9000A-68000	705640	89536	705640			

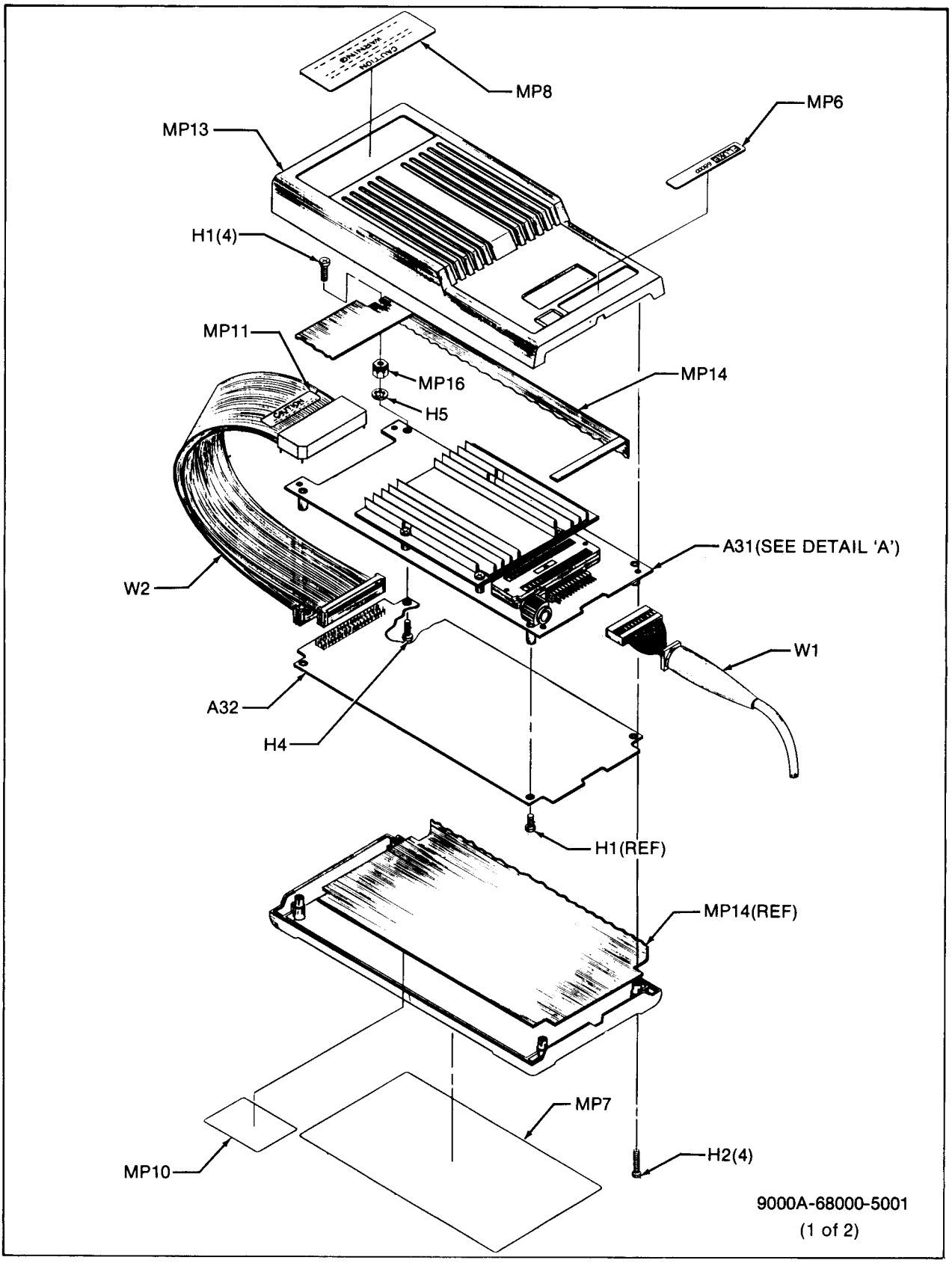


Figure 7-1. 9000A-68000 Final Assembly

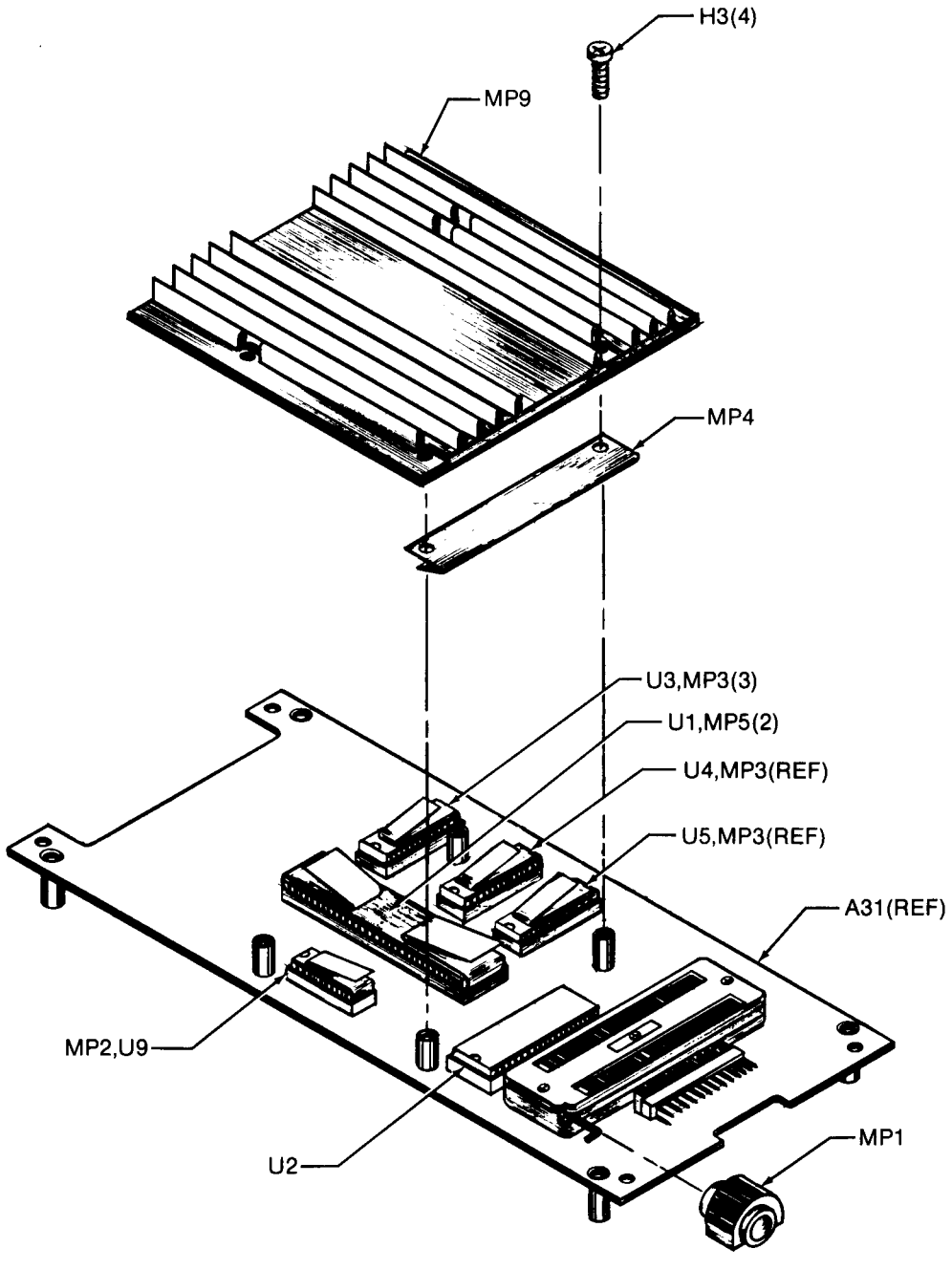


Figure 7-1. 9000A-68000 Final Assembly (cont)

Table 7-2. A31 Processor PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A31⊗	PROCESSOR PCB ASSEMBLY FIGURE 7-2 (9000A-68000-4071)	609149	89536	651562	REF		
C1, C2	CAP, CER, 0.001 UF +/-20%, 500V	402966	72982	8121-A100-W5R-102M	2		
C3-C6	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015A1	4		
C7-C10	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	9		
C12-C16	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	REF		
C17	CAP, CER, 470 PF +/-20%, 100V	358275	72982	8111-A100-W5R-471M	1		
C18	CAP, CER, 150 PF +/-20%, 50V	512988	89536	512988	1		
CR1, CR2	DIODE, SI, SMALL SIGNAL	313247	28484	HP5082-6264	2	1	
J1	CONNECTOR, RIGHT ANGLE, 26-POS	512590	89536	512590	1		
J2	SOCKET, 64-PIN, ZERO-INSERTION FORCE	585174	89536	585174	1		
MP1	SPACER, IC 14-PIN DIP (not shown)	441865	32559	814-060	1		
MP2	SPACER, STANDOFF, BRASS (not shown)	442913	89536	442913	4		
MP3	SPACER, STANDOFF, ROUND (not shown)	380329	89536	380329	4		
P1,P2	CONNECTOR, PIN	267500	00779	87022-1	113		
R1	RES, DEP. CAR, 39 +/-5%, 1/4W	340836	80031	CR251-4-5P39E	1		
R2-R7	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	9		
R8	RES, DEP. CAR, 510 +/-5%, 1/4W	441600	80031	CR251-4-5P510E	1		
R9	RES, DEP. CAR, 680 +/-5%, 1/4W	368779	80031	CR251-4-5P200E	1		
R10	RES, DEP. CAR, 1.2K +/-5%, 1/4W	441378	80031	CR251-4-5P1K2	1		
R11	RES, DEP. CAR, 270 +/-5%, 1/4W	348789	80031	CR251-4-5P270E	5		
R12	RES, DEP. CAR, 390 +/-5%, 1/4W	441543	80031	CR251-4-5P390E	1		
R13	RES, DEP. CAR, 620 +/-5%, 1/4W	442319	80031	CR251-4-5P620E	1		
R14-R17	RES, DEP. CAR, 270 +/-5%, 1/4W	348789	80031	CR251-4-5P270E	REF		
R18	RES, DEP. CAR, 2K +/-5%, 1/4W	441469	80031	CR251-4-5P2K	1		
R19,R20	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R21-R23	RES, DEP. CAR, 56 +/-5%, 1/4W	342618	80031	CR251-4-5P56E	5		
R24	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	REF		
R25,R26	RES, DEP. CAR, 56 +/-5%, 1/4W	342618	80031	CR251-4-5P56E	REF		
R27, R28	RES, COMP, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	2		
R29	RES, DEP. CAR, 470 +/-5%, 1/4W	343434	80031	CR251-4-5P470E	1		
TP1,TP2	CONNECTOR, TEST POINT	512889	02660	62395	2		
U6,U7	IC, TTL, LO-PWR SCHTKY OCTAL D TYPE F/F	454892	01295	SN74LS273N	2	1	
U8	IC, LSTTL, 3 TO 8 BIT DCDR W/ENABLE	407585	01295	SN74LS138N	1		
U10	IC, TTL, SYNCHRONOUS COUNTERS	585588	01295	SN74LS161N	1	1	
U11,U12⊗	IC, FTTL, DUAL 4-INPUT MULTIPLEXER	659912	07263	74F153PC	2	1	
U13	IC, LSTTL, LO-POWER SCHOTTKY	393124	01295	SN74LS74N	1	1	
U14⊗	IC, FTTL, DUAL JK F/F POS EDGE TRIG	654673	07263	74F109PC	1	1	
U15⊗	IC, DUAL DIFFERENTIAL COMPARATOR	647115	18324	NE522N	1	1	
U16	IC, TTL, QUAD 2-INPUT EXLUSIVE OR GATE	408237	01295	SN74LS86N	1	1	
U17,U18⊗	IC, FTTL, TRIPLE 3-INPUT NAND GATE	650531	07263	74F10PC	2	1	
U19	IC, FTTL, QUAD 2-INPUT NAND GATE	654640	07263	74F00PC	1	1	
U20	IC, TTL, QUAD, 2-INPUT, POS NAND GATE	393033	01295	SN74LS00N	1	1	
U21⊗	IC, FTTL, DUAL 4-INPUT NAND GATE	654665	07263	74F20PC	1	1	
U22⊗	IC, FTTL, QUAD 2-INPUT NOR GATE	654657	07363	74F02PC	1	1	
U23,U24⊗	IC, FTTL, HEX INVERTER	634444	07235	74F04PC	2	1	

Table 7-2. A31 Processor PCB Assembly (cont)

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
U25Ⓐ	IC, FTTL, QUAD 2-INPUT AND GATE	650523	07263	74F08PC	1	1	
U26Ⓐ	IC, FTTL, TRIPLE 3-INPUT AND GATE	650549	07263	74F11PC	1	1	
U27,U28Ⓐ	IC, FTTL, QUAD 2-INPUT OR GATE	659904	07263	74F32PC	2	1	
U29,U30	IC, TTL, OCTAL BUFFER/LINE DRIVER	634105	04713	SN74LS541N	2	1	
U31Ⓐ	IC, FTTL, D F/F POS EDGE TRIG	659508	07263	74F74PC	1	1	
W1	RECEPTACLE, JUMPER	530253	00779	530153-2	1		
W1,W2	CONNECTOR, PIN (W/W1 PIN CONNECTOR	267500	00779	87022-1	REF		
W3	WIRE, JUMPER, #22AWG	529701	89536	529701	1		
XU1	SOCKET, IC, 64-PIN	483842	06776	ICN-649-S5-T/G	1		
XU2	SOCKET, IC, 40-PIN	429282	09922	DILB40P-108	1		
XU3-XU5	SOCKET, IC, 24-PIN	376236	91506	324-AG39D	3		
XU9	SOCKET, IC, 20-PIN, DIP	454421	09922	DILB20P-108	1		
Y1	CRYSTAL, 6.5 MHZ	586933	89536	586933	1	1	
Z1,Z2	RESISTOR NETWORK, 2K, DIP	574905	89536	574905	2	1	

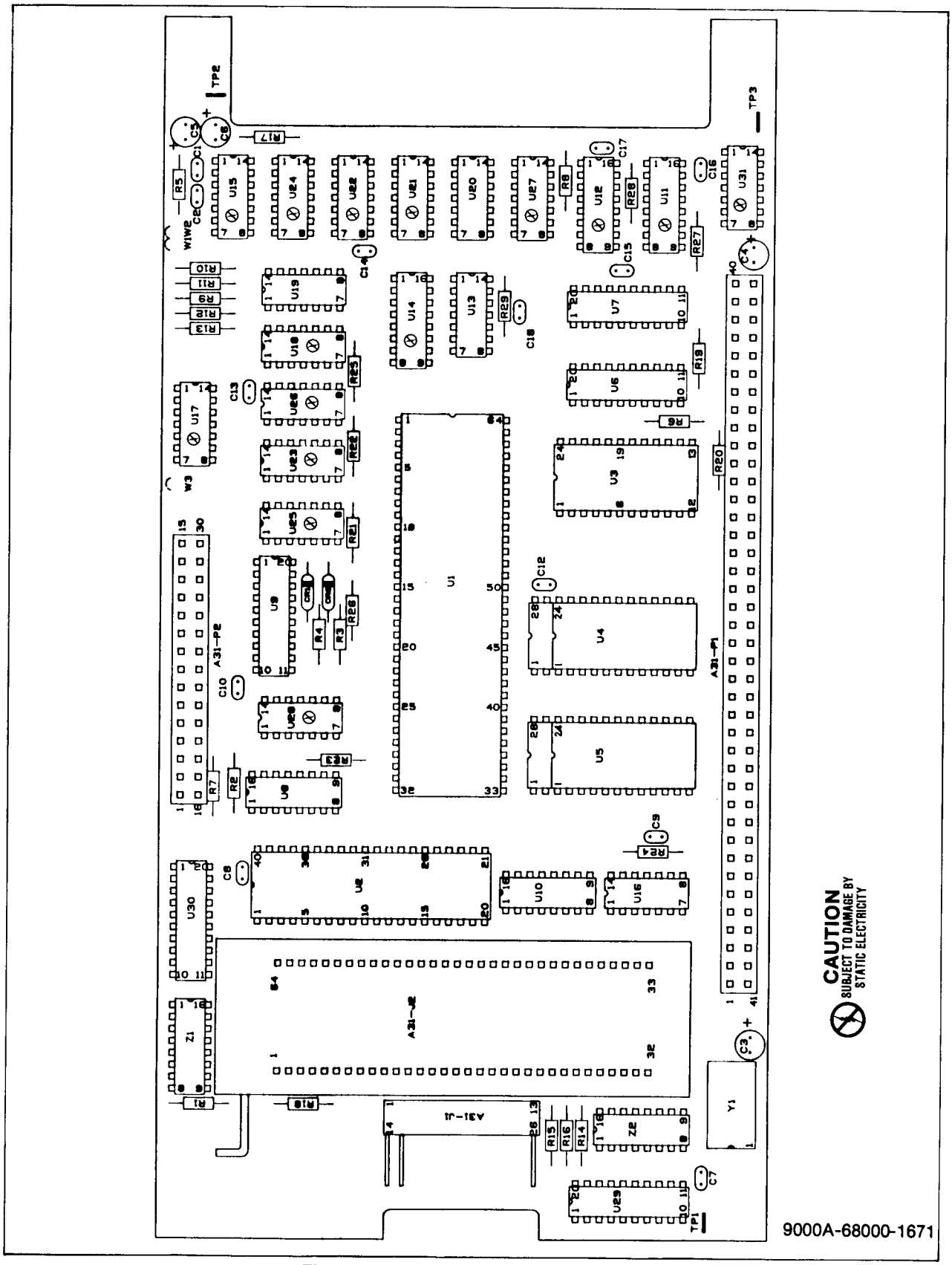


Figure 7-2. A31 Processor PCB Assembly

Table 7-3. A32 Interface PCB Assembly

REF DES	DESCRIPTION	FLUKE STOCK NO.	MFG SPLY CODE	MFG PART NO.	TOT QTY	REC QTY	NOTE
A32Ⓧ	INTERFACE PCB ASSEMBLY FIGURE 7-3 (9000A-68000-4072)	609180	89536	651588	REF		
BA1-A7	HYBRID, 38 OHM, TESTED	607663	89536	607663	7	2	
BA8	HYBRID, PROTECTION 700 OHMS, TESTED	582189	89536	582189	1	1	
C1-C4	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015A1	5		
C5	CAP, TA, 1 UF +/-20%, 35V	161919	56289	196D010X0035G	1		
C6	CAP, CER, 0.01 UF +/-20%, 100V	407361	72982	8121-A100-W5R-103M	1		
C7-C27	CAP, CER, 0.22 UF +/-20%, 50V	519157	51406	RPE111Z5U224M50V	21		
C28	CAP, TA, 10 UF +/-20%, 15V	193623	56289	196D106X0015A1	REF		
C29	CAP, CER, 100 PF +/-2%, 100V	512848	89536	512848	1		
CR1	DIODE, SI, SMALL SIGNAL	313247	28484	HP5082-6264	1	1	
J1	CONNECTOR, DBL ROW, 80-POS	602805	00779	86396-8	1		
J2	CONNECTOR, DBL ROW, 30-POS	529230	00779	86396-3	1		
J3	CONNECTOR, POST	267500	00779	87022-1	128		
R1	RES, DEP. CAR, 1K +/-5%, 1/4W	343426	80031	CR251-4-5P1K	1		
R2	RES, DEP. CAR, 470 +/-5%, 1/4W	147983	01121	CB4715	1		
R3,R4	RES, DEP. CAR, 200 +/-5%, 1/4W	441451	80031	CR251-4-5P200E	2		
R5	RES, MTL. FILM, 10K +/-1%, 1/8W	168260	91637	CMF551002F	1		
R6	RES, MTL. FILM, 5.49K +/-1%, 1/8W	334565	91637	CMF555491F	1		
R7	RES, DEP. CAR, 10K +/-5%, 1/4W	348839	80031	CR251-4-5P10K	1		
R8	RES, DEP. CAR, 680K +/-5%, 1/4W	442517	80031	CR251-4-5P680K	1		
R9	RES, COMP, 4.7K +/-5%, 1/4W	348821	01121	CB4725	1		
R10	RES, DEP. CAR, 2.2K +/-5%, 1/4W	343400	80031	CR251-4-5P2K2	1		
R11,R12	RES, DEP. CAR, 2K +/-5%, 1/4W	441469	80031	CR251-4-5P2K	3		
R13	RES, DEP. CAR, 47 +/-5%, 1/4W	441592	80031	CR251-4-5P47E	1		
R14	RES, DEP. CAR, 24 +/-5%, 1/4W	442210	80031	CR251-4-5P24E	1		
R15	RES, DEP. CAR, 2.7K +/-5%, 1/4W	386490	80031	CR251-4-5P2K7	1		
R16	RES, DEP. CAR, 2K +/-5%, 1/4W	441469	80031	CR251-4-5P2K	REF		
TP1,TP2	CONNECTOR, TEST POINT	512889	02660	62395	2		
U1-8Ⓧ	IC, CMOS, OCTAL D F/F W/TRI-STATE	585364	36665	74SC374A	8	2	
U9-U11	IC, TTL, OCTAL D-TYPE EDGE TRIGGERED F/F	473223	01295	SN74LS374N	3	1	
U12-U17	IC, LSTTL, SELECTED	609446	89536	609446	6	2	
U18	IC, TTL, QUAD 2-INPUT EXCLUSIVE OR GATE	408237	01295	SN74LS86N	1	1	
U19	IC, TTL, LPS, DUAL J-K F/F	412999	01295	SN74LS109N	1	1	
U20	IC, PROTECTION	585992	89536	585992	1	1	
U21	IC, LINEAR, OP-AMP	472894	12040	LM311N	1	1	
U22	IC, LSTTL, LO-POWER SCHOTTKY	393124	01295	SN74LS74N	1	1	
U23Ⓧ	IC, FTTL, DUAL D F/F POS EDGE TRIG	659508	07263	74F74PC	1	1	
VR1	VOLT REFERENCE, 1.22V BAND GAP	452771	89536	452771	1		
W1,W3	WIRE, JUMPER, #22 AWG	529701	89536	529701	2		
XU1-8, XU12-17	SOCKET, IC, 20-PIN DIL	454421	09922	DILB20P-108	14		
XU20	SOCKET, IC, 18-PIN	418228	91506	318-AG39D	1		
XVR1	TRANSIMOUNT (w/VR1)	175125	89536	175125	1		
Z1	NETWORK, THICK-FILM RESISTOR	583476	89536	583476	1	1	
Z2, Z3	RESISTOR NETWORK, 2K, SIP	446880	89536	446880	2	1	
Z4, Z5	RESISTOR NETWORK, 1K, SIP	448308	89536	448308	2	1	

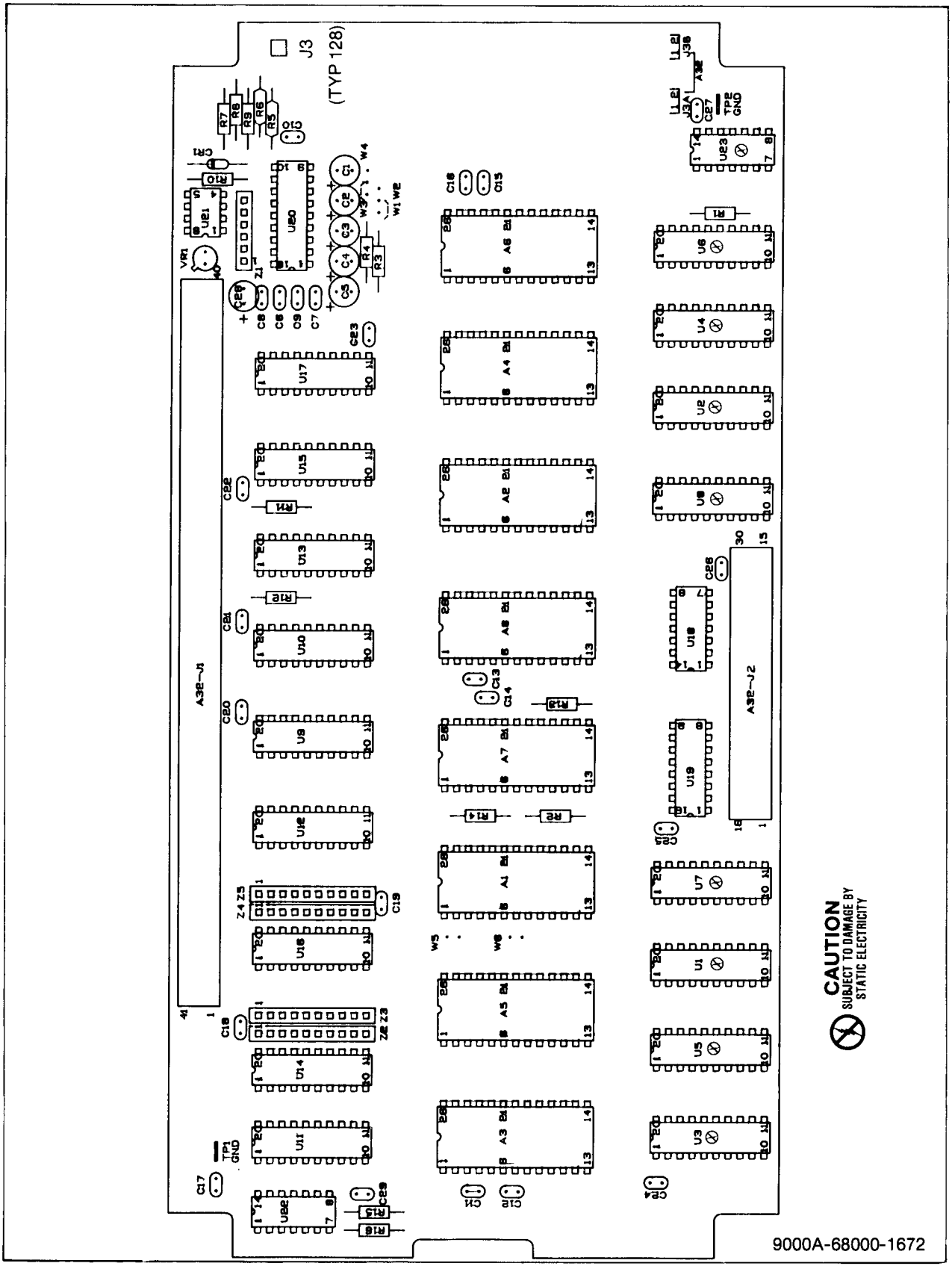


Figure 7-3. A32 Interface PCB Assembly

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								
A31	Processor PCB Assembly	651562	●	1	X																				
A32	Interface PCB Assembly	651588	●	3	2	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.

Change No. 1 17506
Add CR 3,4 Diode, 1N270
Delete C18
Delete R29
Change R27, R28 from Res.CF 1K, 1/4W.10%
to Res.CF 1/8W.10%
Disconnect U8-6 from the AS lioe
Connect U86 to +5
Delete TP3
Add XU11 Socket, 16 pin

Change No. 2 17507
Change R2 from Res.CF 1K, 1/4W.5%
to Res.CF 470, 1/4W.5%
Connect A7-3 to the LE line
Swap J1-7 and J1-18

Change No. 3 18588
Change from XU1-8, 12-17 Socket, IC, 20-Pin qty 14
to XU 12-17 Socket, IC, 20-Pin qty 6

Section 8
Schematic Diagrams

TABLE OF CONTENTS

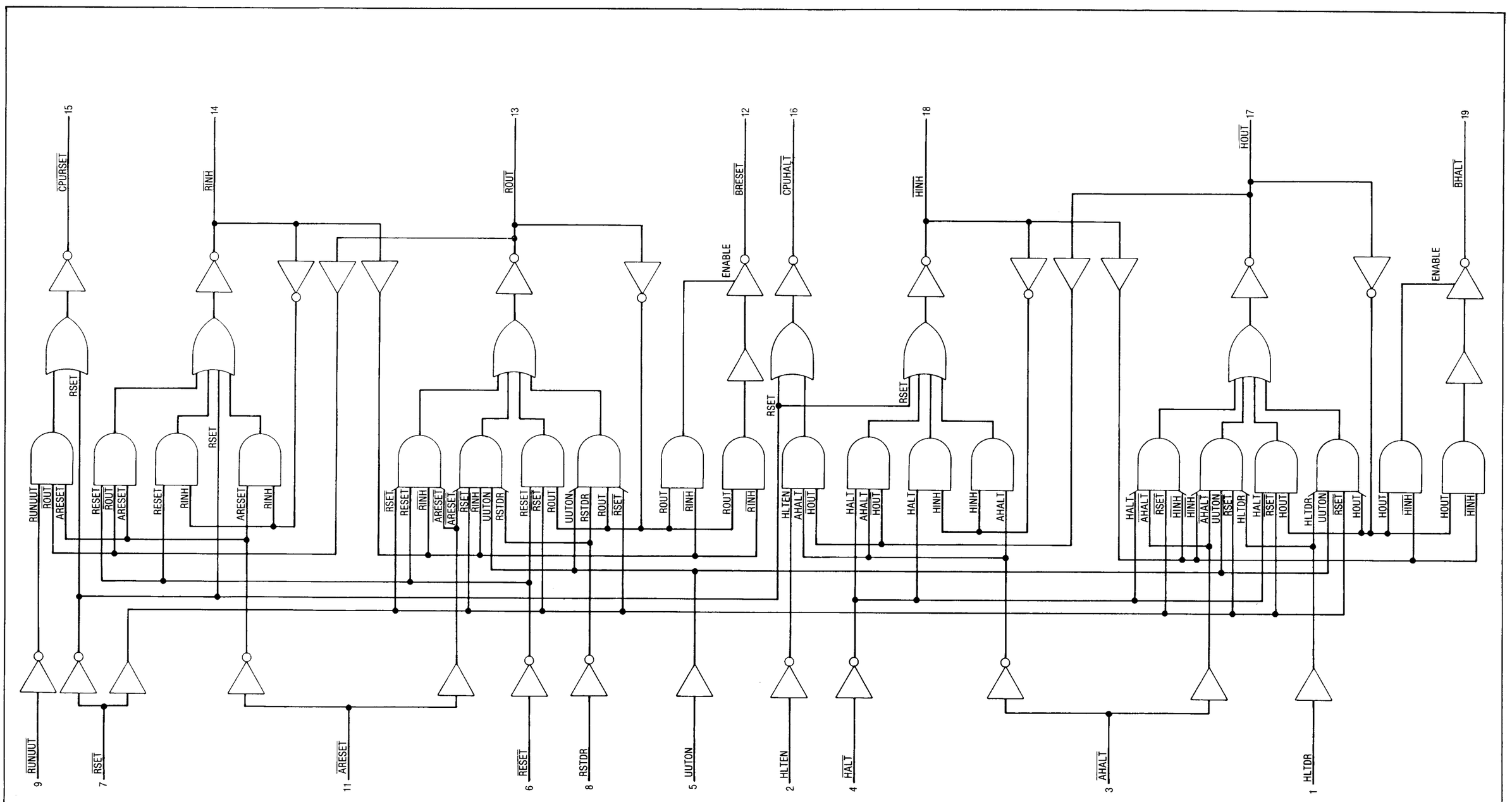
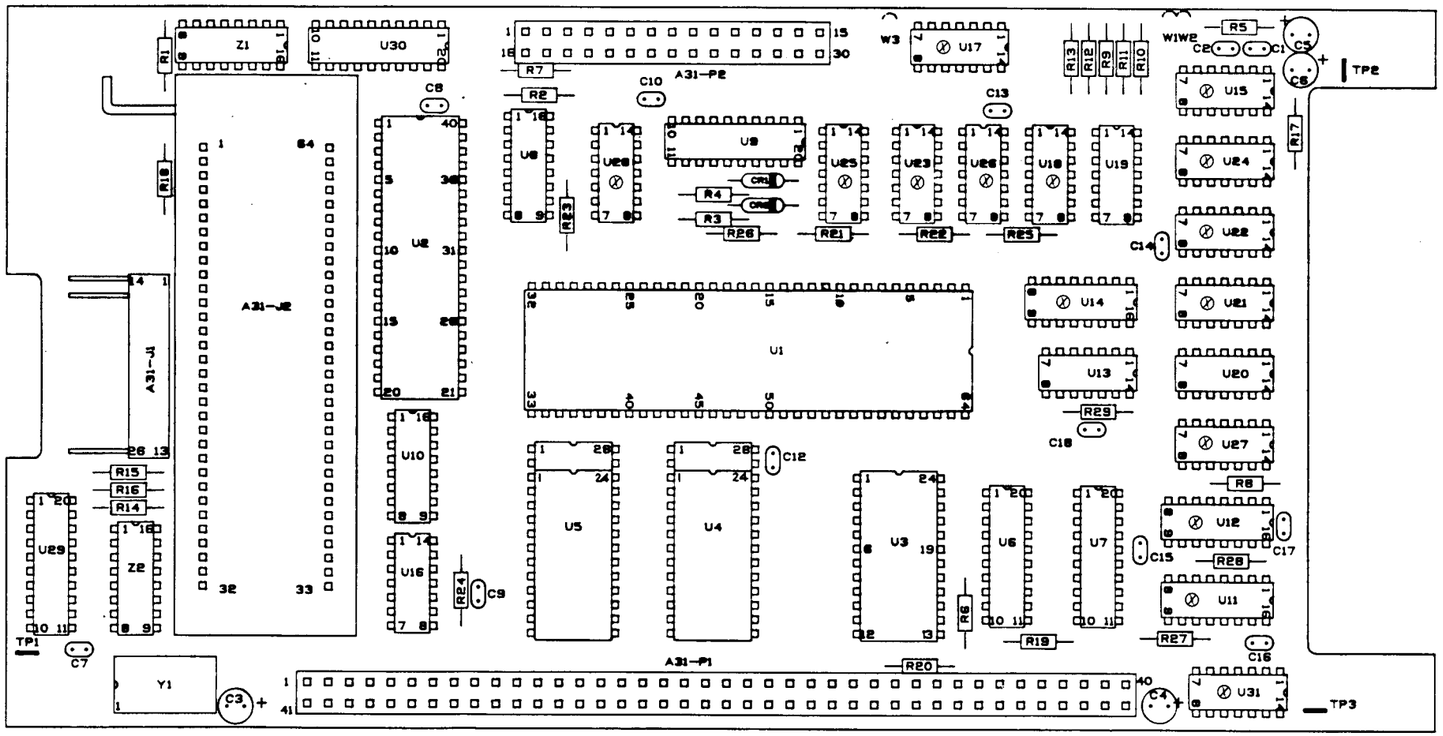
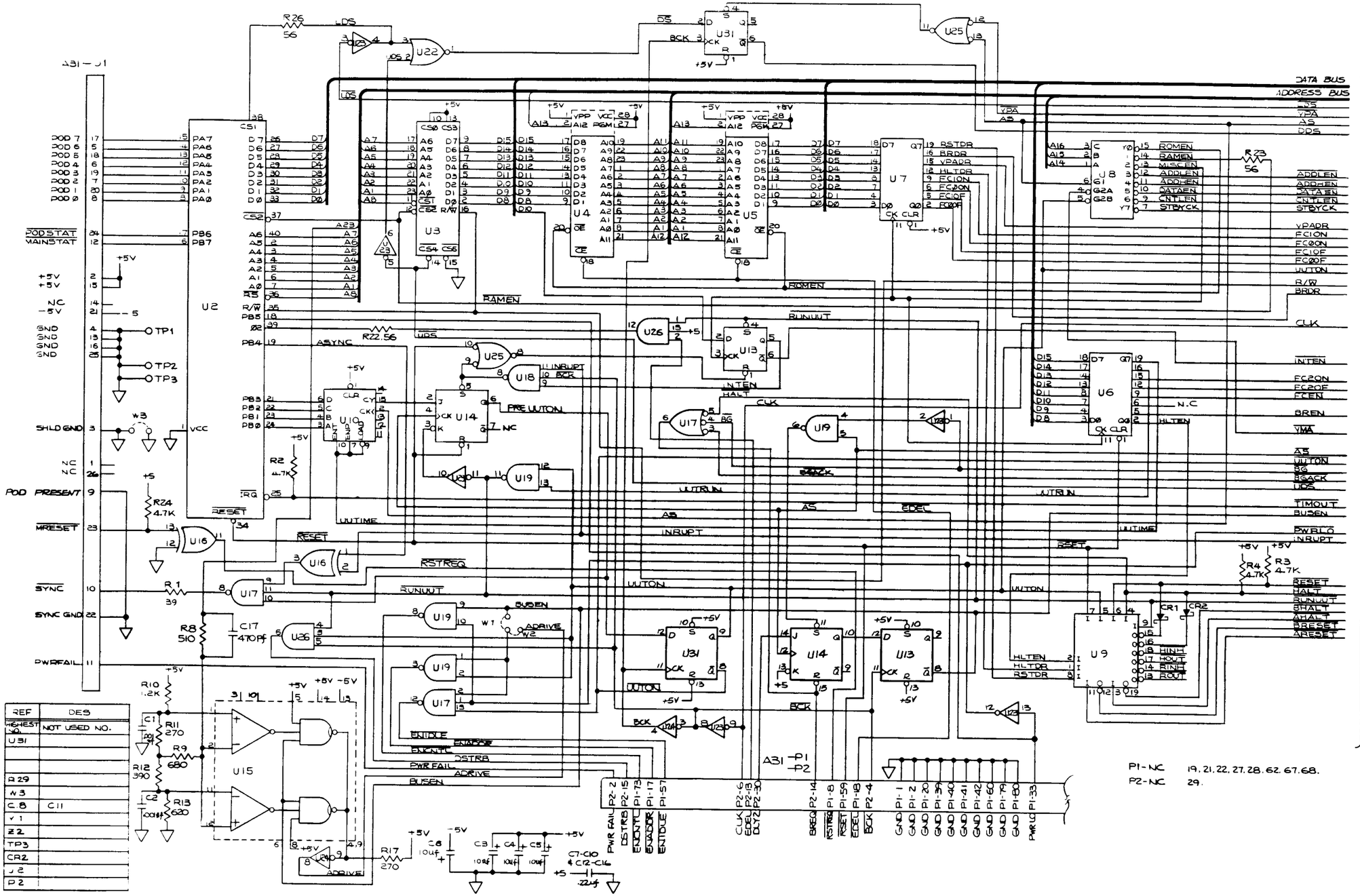


Figure 8-1. Schematic Diagram of U9 on A31 Processor PCB Assembly



9000A-68000-1671

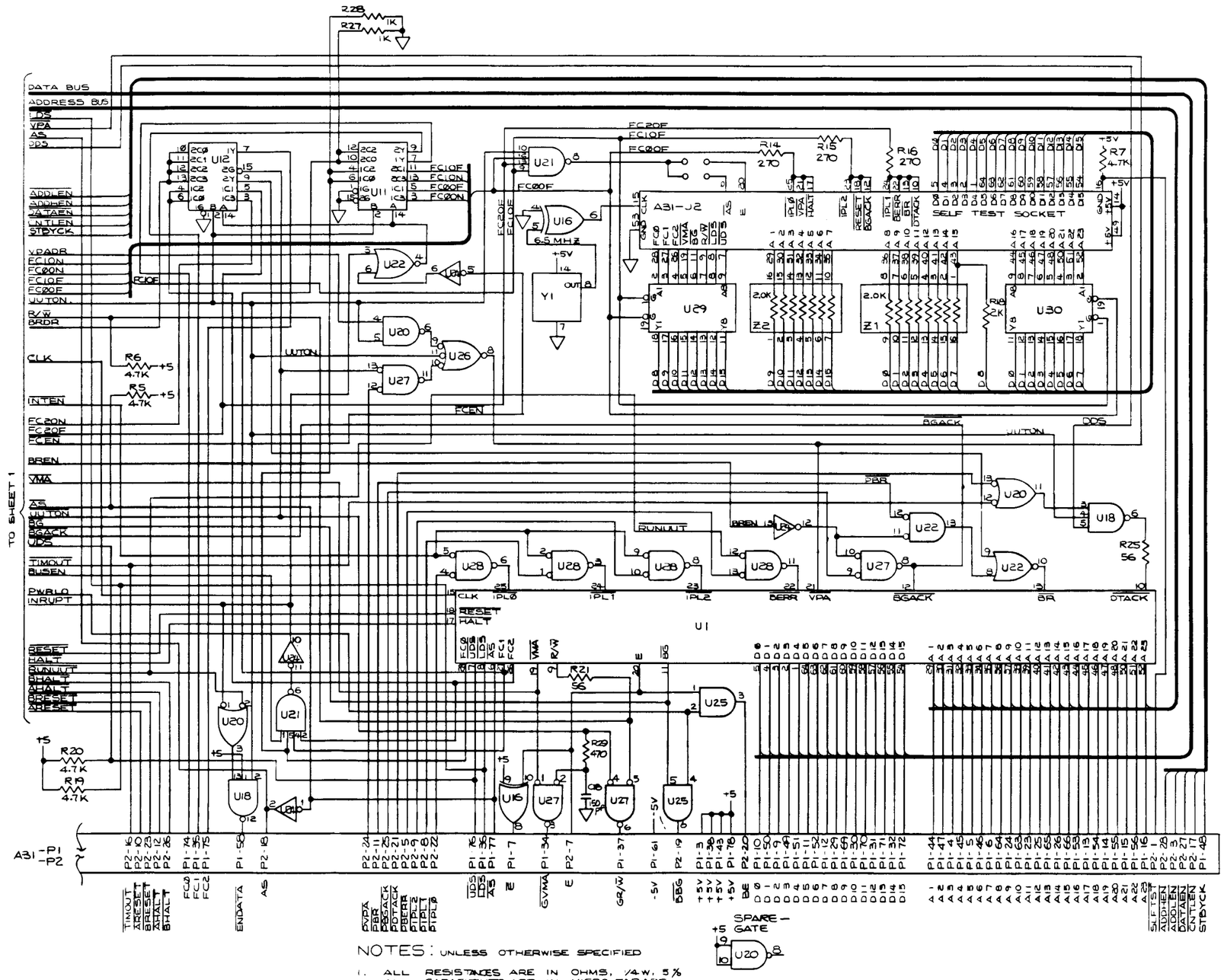
Figure 8-2. A31 Processor PCB Assembly



REF	DES
SHEET	NOT USED NO.
U31	
R29	
W3	
C.8	C11
Y1	
Z2	
TP3	
CR2	
J2	
P2	

9000A-68000-1071
(1 of 2)

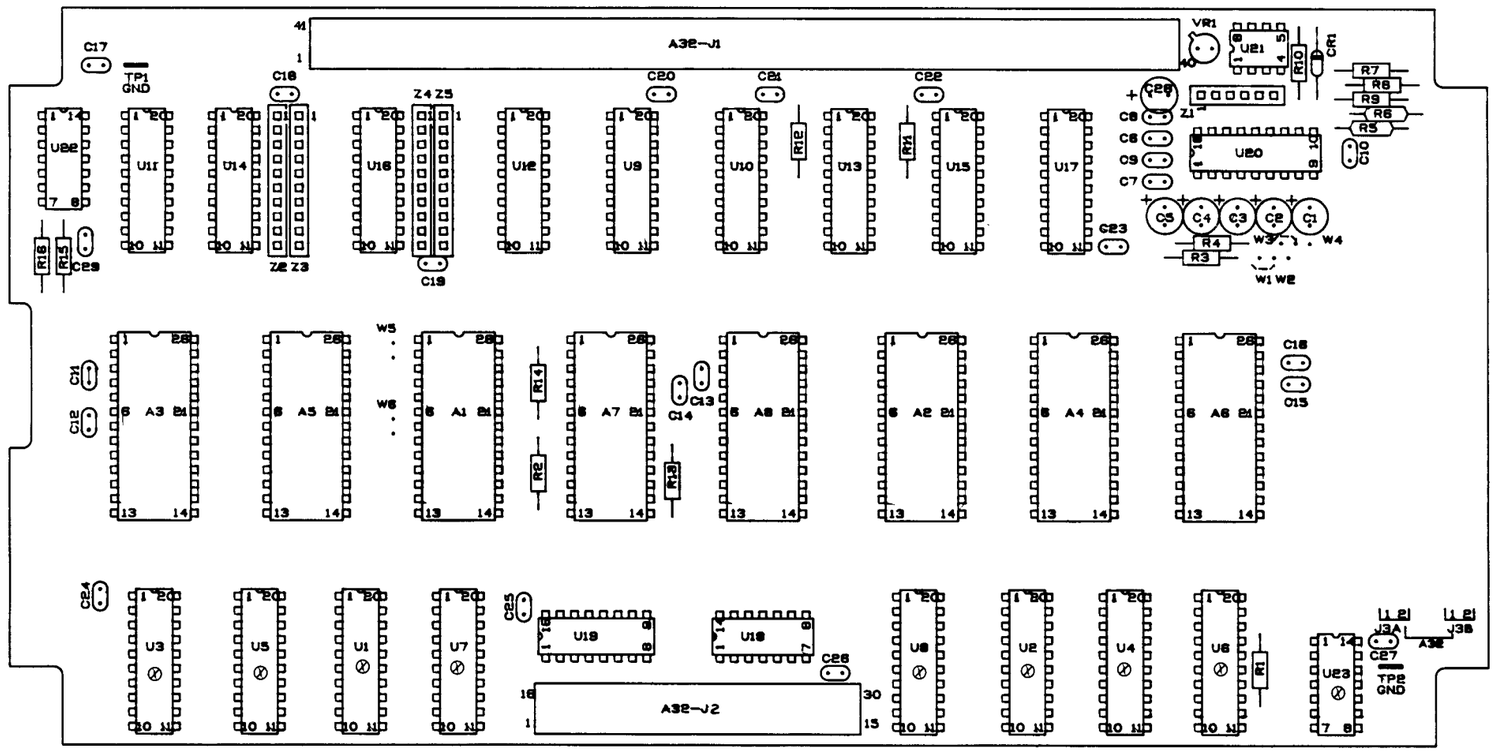
Figure 8-2. A31 Processor PCB Assembly (cont)



REF. DES.	DEVICE	PINS	+5V	GND	QTY
U1	68000	64	14,49	16,53	1
U2	6532A	40	20	1	1
U3	68B10	24	24	1	1
U4,5	2732, 2764	24/28	24/28	12/14	2
U6,7	74LS273	20	20	10	2
U8	74LS138	16	16	8	1
U9	PAL16L8	20	20	10	1
U10	74LS161	16	16	8	1
U11,12	74F153	16	16	8	2
U13	74LS74	14	14	7	1
U14	74F109	16	16	8	1
U15	NE522	14	14	7	1
U16	74LS86	14	14	7	1
U17,18	74F10	14	14	7	2
U19	74F00	14	14	7	1
U20	74LS00	14	14	7	1
U21	74F20	14	14	7	1
U22	74F02	14	14	7	1
U23,24	74F04	14	14	7	2
U25	74F08	14	14	7	1
U26	74F11	14	14	7	1
U27,28	74F32	14	14	7	2
U29,30	74LS541	20	20	10	2
U31	74F74	14	14	7	1

9000A-68000-1071
(2 of 2)

Figure 8-2. A31 Processor PCB Assembly (cont)



9000A-68000-1672

Figure 8-3. A32 Interface PCB Assembly

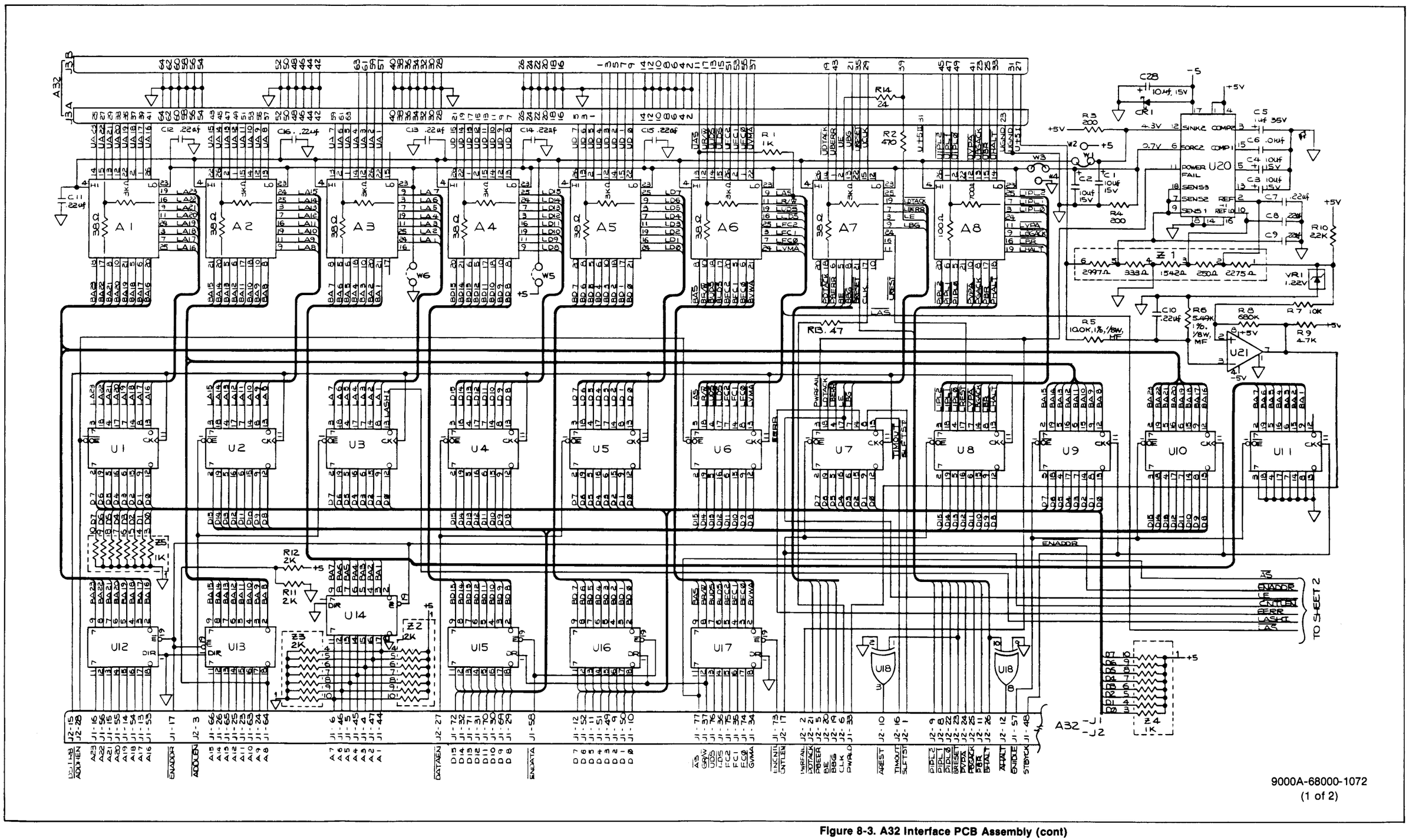


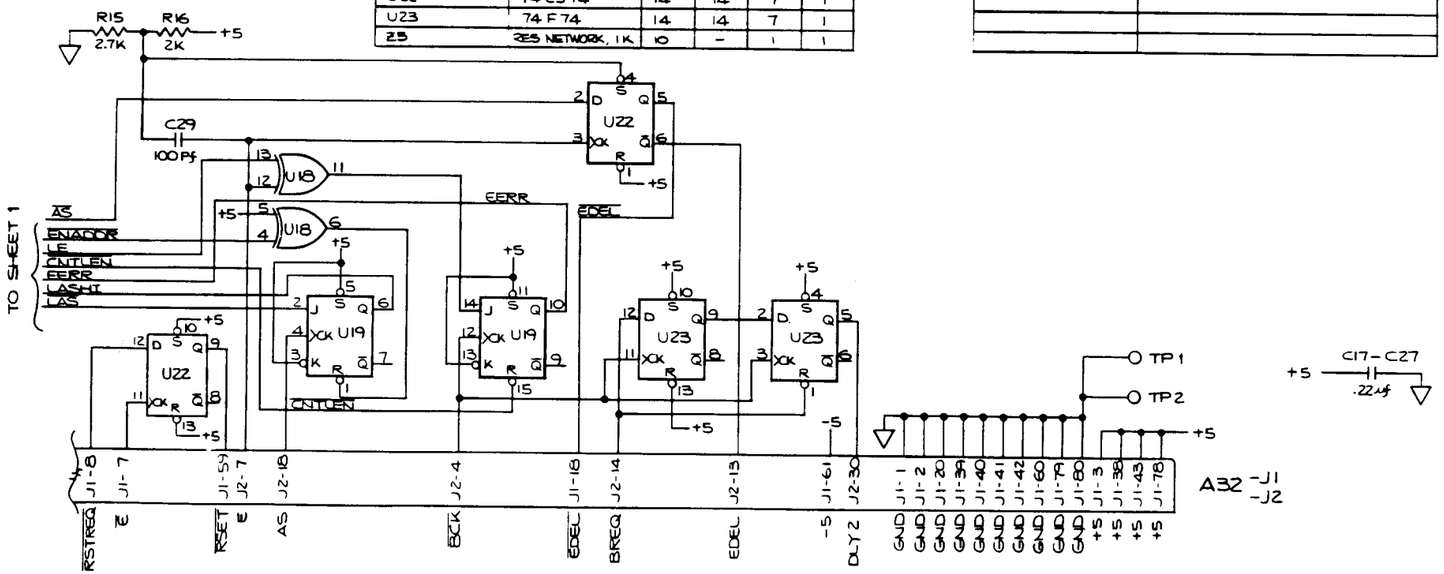
Figure 8-3. A32 Interface PCB Assembly (cont)

NOTES : UNLESS OTHERWISE SPECIFIED.

1. ALL RESISTANCES ARE IN OHMS, 5%, 1/4 W, CC.
   ALL CAPACITANCES ARE IN MICROFARADS.

REF. DES.	DEVICE	PINS	+5V	GND	QTY
U1 – U8	74SC374	20	20	10	8
U12 – U17	9000A-68000-45017	20	20	10	6
U18	74LS86	14	14	7	1
U19	74LS109	16	16	8	1
U20	9000A-8076	18	1, 4	8,14,16	1
U21	LM311	8	—	1	1
U9 – U11	74LS374	20	20	10	3
A1 – A7	9000A-4HOST	26	—	—	7
A8	9000A-4HO2T	26	—	—	1
Z1	54-4087T	—	—	4	1
Z2	RES NETWORK, 2K	10	1	—	1
Z3	RES NETWORK, 2K	10	—	1	1
Z4	RES NETWORK, 1K	10	1	—	1
U22	74LS74	14	14	7	1
U23	74F74	14	14	7	1
Z5	RES NETWORK, 1K	10	-	1	1

REF	DES
LAST USED	NOT USED
A8	
U23	
R16	
C29	
Z5	
VR1	
CR1	
TP2	
W6	
J3	



J1-NC. 62,68,19,21,22,27,28,67.
J2-NC. 29.

9000A-68000-1072
(2 of 2)

Figure 8-3. A32 Interface PCB Assembly (cont)

INDEX

The number following each index entry indicates the page number.