

9010A TROUBLESHOOTING SEMINAR

STUDENT WORKBOOK

P/N 805663

Copyright (c) 1985, John Fluke Mfg. Co., Inc., all rights reserved.

Course Description
for
9010A Troubleshooting Seminar

This two day course is designed to teach you the 9000 Series Troubleshooting Philosophy through actual troubleshooting applications. You will learn how to use the 9010A built-in tests and interpret the associated error messages so that the proper stimulus can be applied. You will then use this troubleshooting strategy to fault-isolate all 20 "hidden" faults on a specially modified single board computer.

On the first day, faults in the uP Bus, RAM, ROM, and I/O are covered, with heavy emphasis placed on testing and troubleshooting I/O. Writing functional tests for Basic I/O registers is also discussed, as the built-in I/O test is for read/writeable I/O only.

On the second day, you will reinforce the 9010A operations and microcomputer troubleshooting strategies learned on day one by isolating faults in all areas of the single board computer. Again, the I/O section is emphasized, as more complex I/O LSI devices are covered. Also, the Asynchronous Signature Probe Option is used to fault-isolate problems in the DMA Display circuit.

While no previous knowledge of the 9010A is necessary, each participant should have a basic understanding of microcomputer circuitry. This course will benefit the experienced technician as well as the technician who is getting started in microcomputer repair. Those who have attended the course have found the open discussions on troubleshooting strategies, and the practical troubleshooting experience, a valuable resource in using the 9010A on their own micro-system applications.

Table of Contents

SECTION	TITLE	PAGE
	Course Objectives	1
I	Preparing the 9010A for Testing	5
II	Micro-System Troubleshooting Strategy	7
III	Entering the Memory Map	11
IV	Using the BUS Test	13
V	Using the RAM Short Test	21
VI	More About RAM Testing	31
VII	The ROM Test	37
VIII	Troubleshooting I/O	43
IX	Forcing Lines	57
X	Signature Gathering Using the Probe	63
XI	Peripheral Interface Adapter (PIA)	71
XII	Asynchronous Signature Probe Option	81
XIII	More Hands-On Training	99
Appendices		
A	Extra Help for Section XIII Exercises	113
B	UUT Schematics and Diagrams	125

COURSE OBJECTIVES

- I. Preparing the 9010A for testing.
Upon successful completion of this section the student will be able to:
 - A. Connect the pod and probe cables.
 - B. Perform the self tests.
 - C. Perform the power up sequence for the 9010A & the UUT.

- II. Micro-System Troubleshooting Strategy.
Upon successful completion of this section the student will be familiar with:
 - A. The three basic steps of the Troubleshooting Strategy.
 - B. The 9010A keys which will be used during the different steps of the Troubleshooting Strategy.
 - C. How the probe fits into the Troubleshooting Strategy.
 - D. The built-in tests.

- III. Entering the Memory Map.
Upon successful completion of this section the student will be able to:
 - A. Execute the Learn feature.
 - B. Enter memory map and clear the memory map descriptors.
 - C. Enter memory map descriptors manually.
Upon successful completion of this section the student will be familiar with:
 - A. The Learn rules.
 - B. The limitations of the Learn feature.
 - C. How the memory map can be downloaded through the RS232 or cassette interface.

- IV. Using Bus Test to identify driveability problems.
Upon successful completion of this section the student will be able to:
 - A. Determine which uP lines Bus Test will check.
 - B. Interpret the Bus Test error messages.
 - C. Stimulate the uP bus by "looping" on an error.
 - D. Use the probe in the free run mode for measuring logic levels.
 - E. Isolate shorted bus lines.
 - F. Test and troubleshoot a bus related problem that has been inserted onto the TK-80 board with the fault switches.

Course Objectives

V. Using the RAM Short Test.

Upon successful completion of this section the student will be able to:

- A. Determine which uP lines and circuits the RAM Test will check.
- B. Interpret the RAM Test error messages.
- C. Determine when the free run and synchronized modes of the probe should be used.
- D. Stimulate RAM circuits using the write and read Troubleshooting functions.
- E. Test and troubleshoot a RAM Short R/W failure on the TK-80 board when the appropriate fault switch is activated (one bit not read/writeable).

VI. More about RAM testing.

Upon successful completion of this section the student will be able to:

- A. Determine when to use a RAM Short and a RAM Long Test.
- B. Determine how address specifications and the Setup selections can be used with the RAM Long Test.
- C. Determine the difference between troubleshooting a R/W error with all bits not read/writeable and a R/W error with less than all bits not read/writeable.
- D. Test and troubleshoot a RAM Short R/W failure on the TK-80 board when the appropriate fault switch is activated (all bits not read/writeable).

VII. The ROM Test.

Upon successful completion of this section the student will be able to:

- A. Determine what circuits the ROM Test will check.
- B. Determine what a ROM signature is and where it comes from.
- C. Stimulate a ROM circuit.
- D. Determine when the ROM Test should be used.
- E. Determine what the Auto Test does.
- F. Test and troubleshoot a faulty ROM circuit when the appropriate fault switch is activated.
- G. Use the D-TOG, A-TOG, WALK, and RAMP Troubleshooting Functions to stimulate the UUT.

VIII. Troubleshooting I/O.

Upon successful completion of this section the student will be able to:

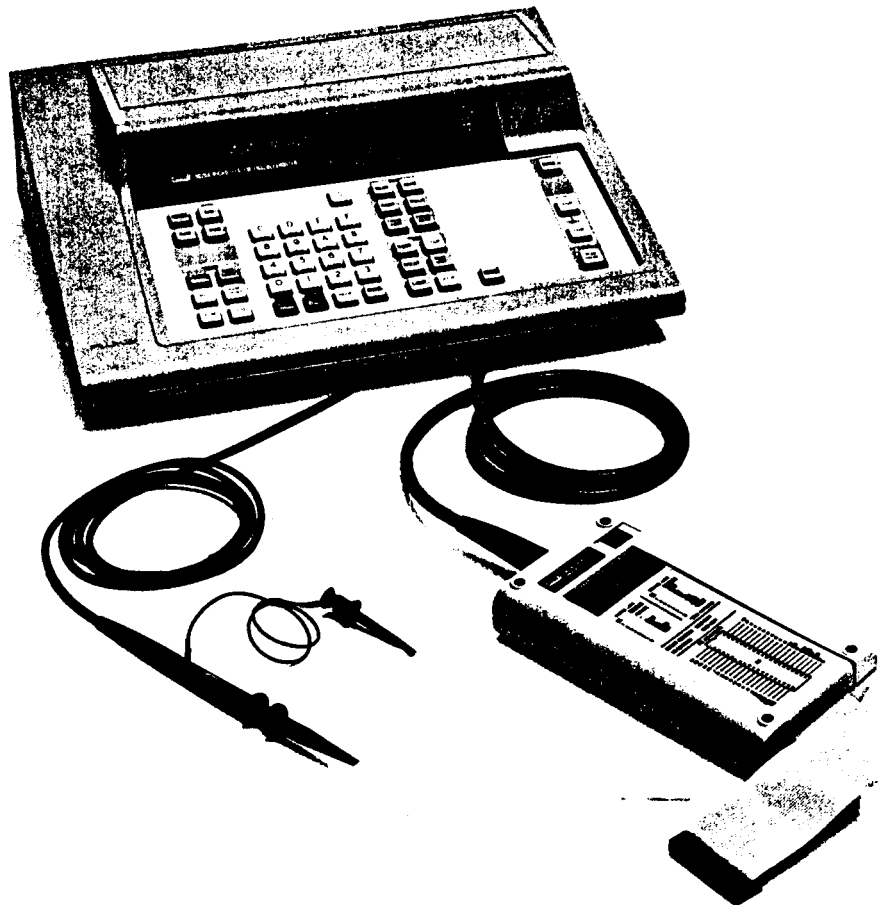
- A. Differentiate an I/O address from a memory address.
- B. Operate the TK-80 I/O devices.
- C. Use the Probe as a Pulser.
- D. Create, edit, and execute a program.
- E. Write a functional test program for basic I/O.
- F. Stimulate and measure I/O circuits.
- G. Test and troubleshoot basic I/O circuits on the TK-80 board using functional tests, troubleshooting functions, and the probe.

- IX. Forcing Lines.
Upon successful completion of this section the student will be able to:
- A. Identify an active Forcing Line.
 - B. Use the SETUP Key to disable Forcing Lines.
 - C. Test and Troubleshoot a faulty Forcing Line on the TK-80 board when the appropriate fault switch is activated.
- X. Signature Gathering using the Probe.
Upon successful completion of this section the student will be able to:
- A. Identify the Probe response data in Register 0.
 - B. Gather a Signature using the Probe.
 - C. Write a Signature Gathering program.
 - D. Test and troubleshoot basic I/O circuits on the TK-80 board using functional tests, troubleshooting functions, and the Probe.
- XI. Peripheral Interface Adapter (PIA).
Upon successful completion of this section the student will be able to:
- A. Operate the TK-80 PIA I/O circuits.
 - B. Write functional test programs for the PIA I/O circuits.
 - C. Test and Troubleshoot PIA I/O circuits on the TK-80 board using functional tests, troubleshooting functions, and the Probe.
- XII. Asynchronous Signature Probe Option.
Upon successful completion of this section the student will be able to:
- A. Connect the Clock Module to the 9010A mainframe.
 - B. Determine how the Clock Module timing inputs are used.
 - C. Modify Setup selections for Clock Module timing.
 - D. Determine the timing of the TK-80 DMA circuit.
 - E. Gather the same signatures and counts on the UUT as are in the example.
 - F. Use the Async Option for troubleshooting faults in the TK-80 DMA circuitry.
 - G. Gather probe data in a program using the Async Option programs in their proper sequence.
 - H. Use the Clock Module inputs for other UUT applications.
- XIII. More Hands-On Training.
Upon successful completion of this section the student will be able to:
- A. Identify which microcomputer circuit is faulty.
 - B. Determine how to stimulate the faulty circuit for troubleshooting.
 - C. Isolate the fault by executing the stimulus and probing the affected nodes.

INSTRUCTOR-LED EXERCISE

Preparing the 9010A

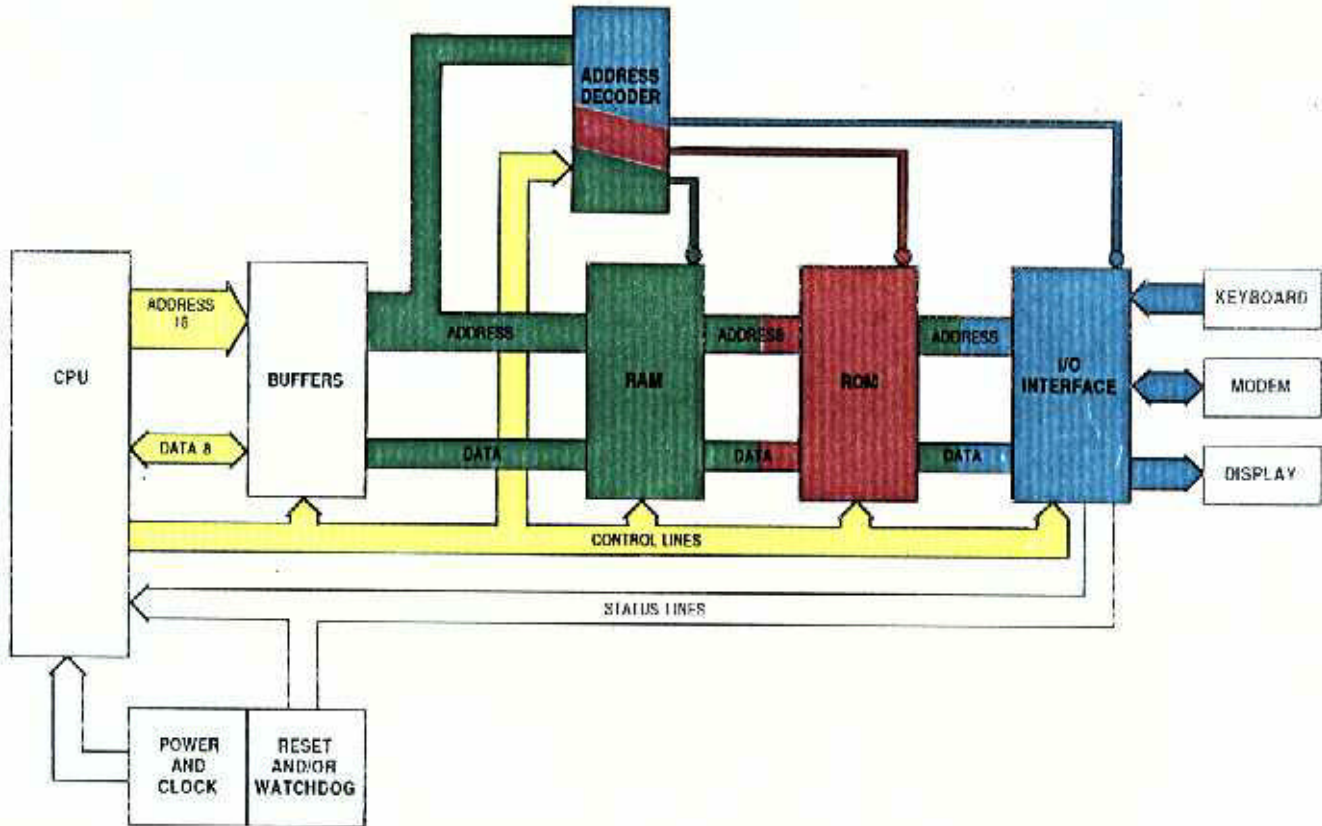
1. Cable connections.
2. Self tests.
 - a. Mainframe checksum.
 - b. Pod checksum and UUT.
 - c. Error message (check manual).
3. Power-up 9010A before UUT to activate POD protection circuits.
4. UUT connection.



Section II
Micro-System Troubleshooting Strategy

MICRO-SYSTEM TROUBLESHOOTING STRATEGY

1. Identify which microcomputer circuit is faulty (see Figure 1).
2. Determine how to stimulate the faulty circuit for troubleshooting (use one of the troubleshooting functions or "loop" on the error).
3. Isolate the fault by executing the stimulus and probing the affected nodes. Isolate further by removing and/or replacing components.



**MICROCOMPUTER
CIRCUIT**

uP BUS


RAM


ROM

I/O (R/W)


I/O

BUILT-IN TESTS

 — BUS TEST

 — RAM SHORT/LONG TEST

 — ROM TEST

 — I/O TEST

OTHER TESTS

 — FUNCTIONAL TESTS

 — R/W FUNCTION AND PROBE

Figure 1. Identifying the Faulty Circuit Using the Micro-System Troubleshooter

ENTERING THE MEMORY MAP

All the built-in tests, except for Bus Test, require an address or range of addresses at which the test will be performed. There are four ways to enter these test addresses, or memory map, into the Troubleshooter:

- o Using the Learn feature.
- o Manually.
- o Read from the tape.
- o Downloaded through the RS232 interface.

You will enter the UUT memory map using the first two methods in the following exercise.

INSTRUCTOR-LED EXERCISE

Entering the Memory Map

1. Learning the UUT memory map.
 - a. Use only when UUT documentation is not available.
 - b. LEARN @ ENTER will determine map of total address space.
2. Checking for extra descriptors.
 - a. Press RAM, ROM, and I/O VIEW Keys while watching for MORE light.
 - b. Press MORE key to display extra ROM descriptor.
3. Deleting memory map descriptors.
 - a. Display extra descriptor using VIEW and MORE Keys.
 - b. Press CLEAR key.
4. Entering the memory map manually.
 - a. Press VIEW key followed by ENTER key.
 - b. Enter descriptors using HEX keypad and ENTER key.
 - c. Enter ROM signature manually or press ENTER key to determine signature. (This signature needs to be determined on a good UUT.)

BUS TEST

The Bus Test is usually run first to ensure that the uP address, data, and control lines can be driven high and low, and that address and data lines are not tied together. Bus Test does not require a test address and only takes a few seconds to run. If one of the uP bus lines should fail the "driveability" check, then the test will stop and an error message will be displayed. Listed below are Bus Test error messages associated with address and data lines.

ADDR BIT dd TIED HIGH - LOOP?

ADDR BIT dd TIED LOW - LOOP?

ADDR BITS dd AND dd TIED - LOOP?

DATA BIT dd TIED HIGH - LOOP?

DATA BIT dd TIED LOW - LOOP?

DATA BITS dd AND dd TIED - LOOP?

The dd in the error messages above are the decimal numbered bit(s) that are tied (i.e., Data Bit 6, Address Bit 12, etc). The third and sixth messages occur when two or more lines are tied together. If more than two lines are tied together then press the CONT key to display the others. (There are some combinations of multiple tied lines where the Micro-System Troubleshooter is unable to identify all of the lines. In this instance it will only list one as being tied.)

When Bus Test detects a control line driveability problem, it will identify the line(s) through an error message. This message will not report whether it is tied high or low - it only identifies which line is not driveable. The format for the error message is listed below.

CTL ERR bbbbbbbb bbbbbbbb - LOOP?

Where bbbbbbbb bbbbbbbb is a binary representation of the control bits listed on the pod decal. The bits with a value of 1 are the stuck control lines that correspond with the bit numbers on the pod decal. In the example below you can see that the bit position with a 1 in it is bit 5, which is the WR line as listed on the 8080 pod decal.

CTL ERR 00000000 00100000 - LOOP?

The control error message identifies which line is not driveable. You can use the probe to determine whether it is stuck high or stuck low.

Using the Probe in the Free Run Mode to Verify Logic Levels

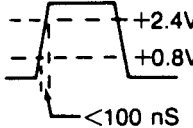
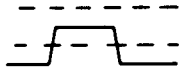
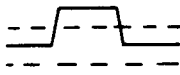
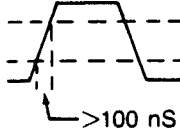
USING THE PROBE IN THE FREE RUN MODE TO VERIFY LOGIC LEVELS

One way to use the 9010A probe is for detecting logic levels. When the probe's green light is on, then a level of less than .8 volts or a logic low was detected. When the probe's red light is on, then a level of more than 2.4 volts or a logic high was detected. When both lights are on, then the probe is on a node that is toggling between a logic high and a logic low.

The probe has several sync modes that allow logic levels to be latched at different intervals. One of these, the free run mode, updates the probe lights with new levels by a 1 kHz free-running clock. The 9010A is powered up in the free run mode.

Table 1 contains a more detailed description of what the probe light activity is indicating.

Table 1. Probe Light Activity Indications

CONDITION	DESCRIPTION
Green on continuously, red off	Indicates a steady dc low.
Red on continuously, green off	Indicates a steady dc high.
Both lights off	Indicates the line remains in the invalid logic continuously.
Both red and green on continuously	Indicates that a line is toggling* between high and low, but is staying in the invalid area for less than 100 ns. An example of this would be a clearly defined square wave. 
Green flashing, red off	Indicates the line is toggling* in between logic low and invalid. 
Red flashing, green off	Indicates the line is toggling* in between logic high and invalid. 
Both lights flashing	Indicates the line is toggling* between all three logic states. 

*The toggling rate is not related to the flashing rate.

INSTRUCTOR-LED EXERCISE

Using the Probe in the Free Run Mode to Verify Logic Levels

1. Review TK-80 schematic.
2. Check the sync mode for free run.
 - a. Press SYNC Key.
 - b. Press ENTER Key.
3. Probe a GND line:

Probe's red __ , green __ light on.
4. Probe a VCC line:

Probe's red __ , green __ light on.
5. Execute a Bus Test. Loop. Probe DBO (U34-15).
 - a. Press BUS Key.
 - b. Press LOOP Key.
 - c. Probe DBO (U34-15):

Probe's red __ , green __ light on.

9010A HANDS-ON TRAINING

Exercise 4A

FAULT:	DATA BUS	FS #1
IDENTIFY USING:	BUS TEST	
VERIFY USING:	LOOP ON ERROR (STIMULUS)	
	PROBE, FREE-RUN (MEASUREMENT)	

1. Set fault switch #1 to its fault position.
2. Perform a BUS TEST and record the 9010A display message.
3. What is the meaning of this message?
4. Respond to this message with the YES key and record what happens.
5. Locate on the UUT schematic possible locations for this fault and verify it using the PROBE.
6. Press BUS TEST, followed by the LOOP key. The 9010A is now looping on the Bus fault.
While LOOPING, switch fault #1 in and out. What happens?
7. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How do you initiate the "loop on error" mode for stimulating a faulty circuit?
 - b. How do you know this fault is located on the uP side of the bus buffer?
 - c. What methods of troubleshooting, other than the 9010A, would you use to find the source of this fault?

Exercise 4B

FAULT:	CONTROL BUS	FS #15
IDENTIFY USING:	BUS TEST	
VERIFY USING:	LOOP ON ERROR (STIMULUS)	
	PROBE, FREE-RUN (MEASUREMENT)	

1. Set fault switch #15 to its fault position.
2. Perform a BUS TEST and record the 9010A display message.
3. What is the meaning of this message? Hint: Use the POD label.
4. Respond with LOOP to this message and record what happens.
5. Toggle fault switch #15 while LOOPING. What happens?
6. Locate on the UUT schematic possible locations for this fault and verify it using the PROBE.

U__ Pin__	Logic Level ____
U__ Pin__	Logic Level ____
7. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. Was the ____ line tied high or low ?
 - b. What other methods of troubleshooting would you use to find the source of this fault?

THE RAM SHORT TEST

RAM Short uses three test algorithms in the following order to check out RAM:

1. Basic read/write test
2. Data Bits tied test
3. Address Decode test

A description of each of these tests along with their associated error messages are listed below.

1. Read/Write Test

RAM Short does a basic READ/WRITE test which checks for the RAM's ability to store data. All ones are written to and then read from RAM, then all zeros are written to and read from RAM. If this R/W test fails then either the RAM chip(s) is bad, there are inactive control lines, or there is an open or shorted data line(s). The display message for this type of error is listed below.

R/W ERR @ aaaa BTS hh - LOOP?

where aaaa is the address of the Read/Write failure and hh is the bit mask indicating the data bit(s) not Read/Writeable.

The easiest way to determine which bits are not Read/Writeable is to convert the hexadecimal bit mask to binary as in the example below. To make converting hex to binary a little easier look at the hex keypad on the Micro-System Troubleshooter to find the binary equivalent.

B7	B6	B5	B4	B3	B2	B1	B0
0	1	0	0	1	1	0	0

If the R/W error message indicated a bit mask of 4C, then that would translate into an 8 bit binary word as shown above. The bit positions with a value of one indicate the bits not read/writeable. Using our example, this means bits 2, 3, and 6 are not read/writeable.

The RAM SHORT Test

2. Data Bits Tied Test

Bus Test is not able to detect whether buffered data lines are tied together. By storing data patterns in RAM and reading them back, RAM Short is able to identify data bits tied together beyond the data buffer. The display message for this type of error is listed below.

RAM BITS d1 AND d2 TIED - LOOP?

where d1 and d2 are the decimal numbers of the data bits tied together.

3. Address Decode Test

Another portion of the RAM Short Test will detect decode errors, thereby identifying problems with the address lines. For each address location to be tested, a unique data word is calculated from the address at which it will be stored. After the data has been calculated and stored, it is then read back. If one of these unique data words is found in the wrong location then there is a decoding problem. This means an address line is shorted, open, or tied. The display message for this type of error is listed below.

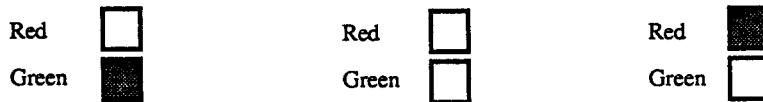
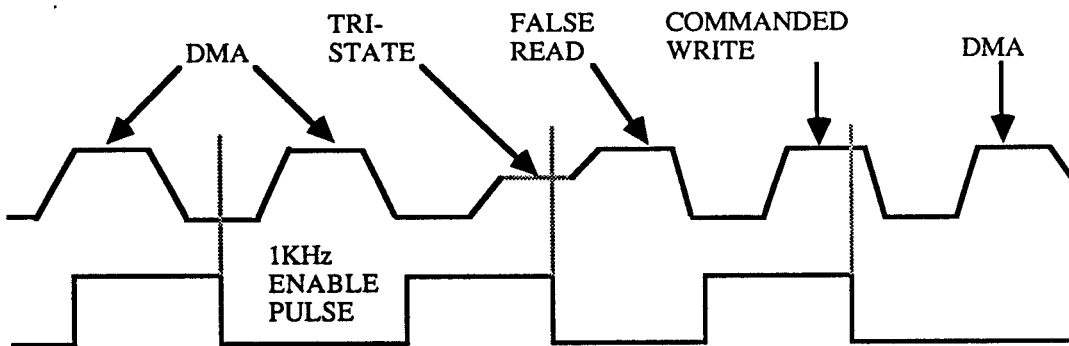
RAM DCD ERR @ aaaa BIT dd - LOOP?

where aaaa is the address of the failure and decimal number dd is the address bit that is not decoding properly.

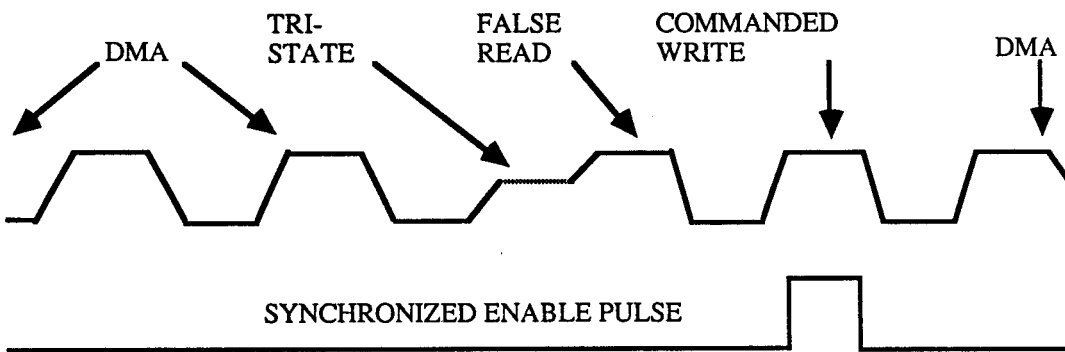
SYNCHRONIZING THE PROBE WITH UUT READ AND WRITE COMMANDS

In the previous section you used the probe to detect logic levels in the free run mode. The probe lights were being updated with new levels every 1 ms by the free running clock. To detect logic levels only during data valid periods, you must synchronize the probe to data. Do this by pressing the SYNC and the D keys. Now the probe will be updated with new information during a UUT read or write cycle only, rather than every 1 ms as in the free run mode.

The difference between the free run mode and the sync data mode are shown in the timing diagrams below. The enable pulses latch probe data on the negative edge.



Free Run



SYNC Data

Figure 2. Probe Timing Diagrams

INSTRUCTOR-LED EXERCISE

Ram Short, R/W Functions, and the Probe

1. Sync the probe to data.
 - a. Press SYNC Key.
 - b. Press D Key.
2. Perform a RAM Short Test.
 - a. Press RAM SHORT Key.
 - b. Press ENTER Key.
3. Write and read data at UUT RAM.
 - a. Press WRITE 8COO (address) ENTER FF(data for all bits high) ENTER.
 - b. Press READ 8COO ENTER.
 - c. Repeat for the following troubleshooting bit patterns:
 - o All low.
 - o Alternating high and low.
 - o One bit high, the rest low.
 - o One bit low, the rest high.
4. Use INC and DEC keys to change Address and Data.
 - a. Press WRITE INC (increments last address) ENTER FF ENTER.
 - b. Press WRITE ENTER (enters last address) INC (increments last data) ENTER.
 - c. Press WRITE INC INC DEC ENTER (address) INC DEC ENTER (data).
5. Probe RAM chips while looping on reads and writes.
 - a. Address and Data lines.
 - b. Chip Select.
 - c. Write Enable.
6. Repeat Step 5 in the free run mode.

9010A HANDS-ON TRAINING

Exercise 5A

FAULT:	DATA BUS	FS #7
IDENTIFY USING:	RAM SHORT, WRITE, READ	
ISOLATE USING:	WRITE (STIMULUS)	
	PROBE, SYNC DATA (MEASUREMENT)	

1. Set fault switch #7 to its fault position.
2. Press BUS TEST. What happens?
3. Press RAM SHORT, then ENTER. What happens?

From this error message, identify which line(s) is bad.

4. WRITE and READ data to and from RAM to determine further what is wrong with the line(s) identified in the previous step.

Can you WRITE and READ the bad bit(s), High?

Can you WRITE and READ the bad bit(s), Low?

5. To stimulate the faulty circuit, use the failure address from the error message in Step 3, and WRITE the data that you were unable to WRITE/READ in Step 4:

WRITE @ _____ = _____

6. While LOOPING on the above stimulus, use the PROBE to isolate this fault by tracing the bad bits to or from RAM.

Where is the fault and what type is it?

7. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How is the value for the stimulus data determined?
 - b. If the WRITE portion of the failed WRITE/READ is OK, what would be the next step?
 - c. Why didn't BUS TEST identify this fault?

USING THE RAM LONG TEST

RAM Long is most often used for finding intermittent and elusive RAM failures. It does everything RAM Short does, plus a pattern sensitivity check, and a more thorough decode. Due to the additional time required to do the pattern sensitivity test, RAM Long is usually done only when necessary. However, there are a couple of ways to deal with the length of RAM Long as explained below.

Address Specification - RAM Long can be shortened considerably by specifying the test to be done on the suspected bad RAM addresses, rather than testing all of UUT RAM.

Recording error messages with a 9010A and printer - When a test error is encountered during a built-in test, the test stops and the error message is reported to the display. This works well for consistent failures, but can be time consuming when trying to loop on a lengthy test while recording intermittent test errors. By changing Setup selections using the SETUP Key, error messages can be sent to the RS232 port rather than the 9010A display. Now you can record error messages on a printer while looping on a test such as RAM Long, without stopping the test. This allows you to record error messages for a long test or intermittent problem without being present.

INSTRUCTOR-LED EXERCISE

Testing with RAM Long

1. Execute RAM LONG with a shortened address range.
 - a. Press RAM LONG Key.
 - b. Enter addresses 8C00 - 8C10.
 - c. Press ENTER Key.

2. Setup 9010A as a Go/No-Go tester.
 - a. Use SETUP and MORE Keys to scroll setup messages.
 - b. Change setup messages:
SET-EXERCISE ERRORS? (answer NO)
SET-BEEP ON ERR TRANSITION? (answer NO)

3. Execute RAM Long while flipping FS #1 in and out.
 - a. Repeat Step 1.
 - b. Press REPEAT Key and flip FS #1.
Errors are sent out the RS232 port.

4. Re-enable the Setup selections.
SET-EXERCISE ERRORS? (answer YES)
SET-BEEP ON ERR TRANSITION? (answer YES)

5. Repeat Step 3.

RAM Chip Failures and Control Line Problems

RAM CHIP FAILURES AND CONTROL LINE PROBLEMS

In previous sections you used BUS Test, RAM Short, and the probe to identify and locate problems with individual address and data lines. If RAM Short fails the Read/Write portion of the test with all data bits not Read/Writeable, then it is unlikely the error is caused by individually shorted or open data lines. You may save time by checking for inactive or absent control lines on the RAM and Bus Buffer chips, as well as the RAM and Bus Buffer chips themselves.

To verify the RAM control lines, you do a looping read and/or write at the failure address, synchronize the probe to data, and trace out the write enable and chip select lines. If RAM control lines are showing activity, then the failure is probably with the RAM or Bus Buffer chip(s).

INSTRUCTOR-LED EXERCISE

RAM Chip Control Lines

1. Execute a looping Write at a RAM address.
 - a. WRITE 8C00 = FF ENTER.
 - b. Press LOOP Key.
2. Determine the addresses used for chip select.
3. Probe chip select circuitry.
4. Truth table for 74LS139 decoders.

INPUTS			OUTPUTS			
EN	B	A	0	1	2	3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

9010A HANDS-ON TRAINING

Exercise 6A

FAULT:	RAM CONTROL LINE	FS #5
IDENTIFY USING:	RAM SHORT	
ISOLATE USING:	WRITE (STIMULUS)	
	PROBE, SYNC DATA (MEASUREMENT)	

1. Set fault switch #5 to its fault position.
2. Troubleshoot the UUT using the appropriate 9010A built-in tests. Record which tests you used and the fault(s) indicated on the 9010A display.
3. The error message in Step 2 indicates there is a problem with all RAM data bits rather than individual RAM data bits. Which UUT circuits affect all RAM data bits?
4. WRITE and READ data to and from RAM to determine further what is wrong with the line(s) identified in the previous step.

Can you WRITE and READ the bad bit(s), High?

Can you WRITE and READ the bad bit(s), Low?
5. To stimulate the faulty circuit, use the failure address from the error message in Step 2, and WRITE the data that you were unable to WRITE/READ in Step 4:

WRITE @ _____ = _____

6. While LOOPING on the above stimulus, troubleshoot this circuit using the PROBE. Should the PROBE be synchronized?

Record the nodes you check and the logic levels you observe:

7. Where is the defect and what type did you discover?

8. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How does troubleshooting a R/W error message with all bits not read/writeable differ from troubleshooting a R/W error message with some bits not read/writeable?

 - b. From troubleshooting the circuit, what gave you the best clue as to the type of defect?

THE ROM SIGNATURE

Before ROM Test can be run a signature must be calculated from the ROM on a known good board. To develop a ROM signature, the data is "compressed" into a 4 digit hexadecimal number by passing all the ROM data through a two-stage CRC type of signature algorithm. You have the option of developing a single signature for an entire block of ROM, or you can develop signatures for each individual ROM chip.

When ROM Test is run the known good signature is compared to the one just measured. If the two don't agree then a two line message will be displayed as shown below. Press the MORE key to examine the second line.

```
ROM ERR @ aaaa - aaaa - LOOP?
```

```
SIG WAS xxxx NOT yyyy _ LOOP?
```

where aaaa - aaaa is the specified ROM address block that failed, xxxx is the signature just measured, and yyyy is the expected signature.

USING ROM TEST

If you decide to run the built-in tests individually instead of running Auto Test, then it is recommended that Bus Test and RAM Short are run before ROM Test. This will give you a preliminary check of the address and data lines complete with diagnostic messages that tell you if there are any bus problems. For example, if there was a shorted address or data line and you ran ROM Test first, then it would fail without giving any clues concerning which address or data line was causing the problem.

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

When to Use ROM Test

1. Activate FS #3.

2. Execute ROM Test.
 - a. Press ROM Key.
 - b. Press ENTER Key.
 - c. Observe both lines of error message using MORE key.

3. Execute Bus Test followed by RAM Short Test.
 - a. Press BUS key.
 - b. Press RAM SHORT Key.
 - c. Press ENTER Key.
 - d. Observe error message.

4. Which test gives better diagnostics?

9010A HANDS-ON TRAINING

Exercise 7A

FAULT:	ROM DECODE CIRCUIT	FS #6
IDENTIFY USING:	ROM TEST	
ISOLATE USING:	READ (STIMULUS)	
	PROBE, SYNC DATA (MEASUREMENT)	

1. Set fault switch #6 to its fault position.
2. Using the Microsystem Troubleshooting Strategy, determine which 9010A built-in tests you would perform first, second, and third. List these tests in their correct order:
 - a.
 - b.
 - c.
3. Perform each of these tests in the above order and record the resulting 9010A display messages. Hint: After the ROM Test, press the MORE key to obtain the complete display message.
 - a.
 - b.
 - c.
4. Which ROM circuits are easiest to check?

Starting with the easiest, list the suspect ROM circuits in the order you would check them.

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

Troubleshooting Functions

1. TOGGL DATA Key.
 - a. Address 8C00.
 - b. Data 00.
 - c. Bit 3.
 - d. Loop.
 - e. Monitor Bit 3 with Probe.
 - f. Monitor Bit 2 with Probe.
 - g. Repeat with Data FF.

2. TOGGL ADDR Key.
 - a. Address 0000.
 - b. Bit 13.
 - c. Loop.
 - d. Monitor Address Bit 13 with Probe.
 - e. Monitor Address Bit 12 with Probe.
 - f. Repeat with Address FFFF.

3. WALK Key.
 - a. Address 10000.
 - b. Data 01.
 - c. Press ENTER and then the REPEAT Key.
 - d. Watch LEDs at Port 10000.

4. RAMP Key.
 - a. Address 10000.
 - b. Loop.
 - c. Watch LEDs at Port 10000.

Section VIII
Troubleshooting I/O

TESTING AND OPERATING THE TK-80 BASIC I/O

The material in this section explains how to access and operate the TK-80 basic I/O. You will need to operate and familiarize yourself with the I/O interfaces and devices before doing the Hands-On Training Exercises.

All of the I/O on the TK-80 is I/O mapped. Because the 256 UUT I/O addresses (0000-00FF) are the same as some of the UUT memory addresses (0000-FFFF), there is one set of read and write lines (I/O RD & I/O WR) for the I/O addresses, and another set (MEM RD & MEM WR) for the memory addresses. To enable the I/O RD and I/O WR lines, the READ or WRITE key is pressed followed by an address between 10000 and 100FF. By prefixing the I/O address with a "1", the Micro-System Troubleshooter is able to differentiate between a memory read or write and an I/O read or write.

I/O Addresses	10000 - 100FF	I/O RD & I/O WR
Memory Addresses	0000 - FFFF	MEM RD & MEM WR

The basic I/O has an output port that controls 8 LEDs, an input port controlled by 8 switches, and 2 seven segment displays. Refer to the chart below for information on how to access the TK-80 basic I/O.

DEVICE TYPE	REF. DESIGNATOR	READ OR WRITE	ADDRESS	DATA
LEDS	U101	WRITE @	10000	XX
SWITCHES	S102	READ @	10000	XX
7-SEG DPY	U100	WRITE @	10001	OY(Rt.DPY) 1Y(Lft.DPY)

Where X is hexadecimal data, and Y is decimal data.

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

Exercising the TK-80 Basic I/O

1. WRITE a FF to Port 10000. Observe the LEDs.
2. WRITE a AA to Port 10000. Observe the LEDs.
3. Perform a looping READ from Port 10000.
 - a. Change the positions of the switches.
 - b. Observe the 9010 display.
4. WRITE a 15 to Port 10001. Observe the 7-SEG DPY.
5. WRITE a 5 to Port 10001. Observe the 7-SEG DPY.
6. Repeat the above exercise with other data.

Using the Probe in the Pulser Mode to Generate Signals

USING THE PROBE IN THE PULSER MODE TO GENERATE SIGNALS

To use the probe as a pulser press the pulser HIGH and/or LOW keys on the bottom right hand side of the mainframe keyboard. If neither of these keys are pressed then the probe tip is maintained at a potential of + 1.9v. If the pulser HIGH key is pressed then a continuous series of TTL-high pulses are generated at the tip. If the pulser LOW key is pressed then a series of TTL-low pulses are generated. Pressing both keys will generate alternating HIGH and LOW pulses.

In Figure 3 you can see that pulses generated in the free run mode occur every 1 ms and are 2 us wide. When the pulses are generated in the sync data mode, they occur during the entire data valid period of the UUT read or write cycle. When the UUT is idle and not doing a read or write then no pulse is being generated. This allows you to drive a node high or low during the data valid period only.

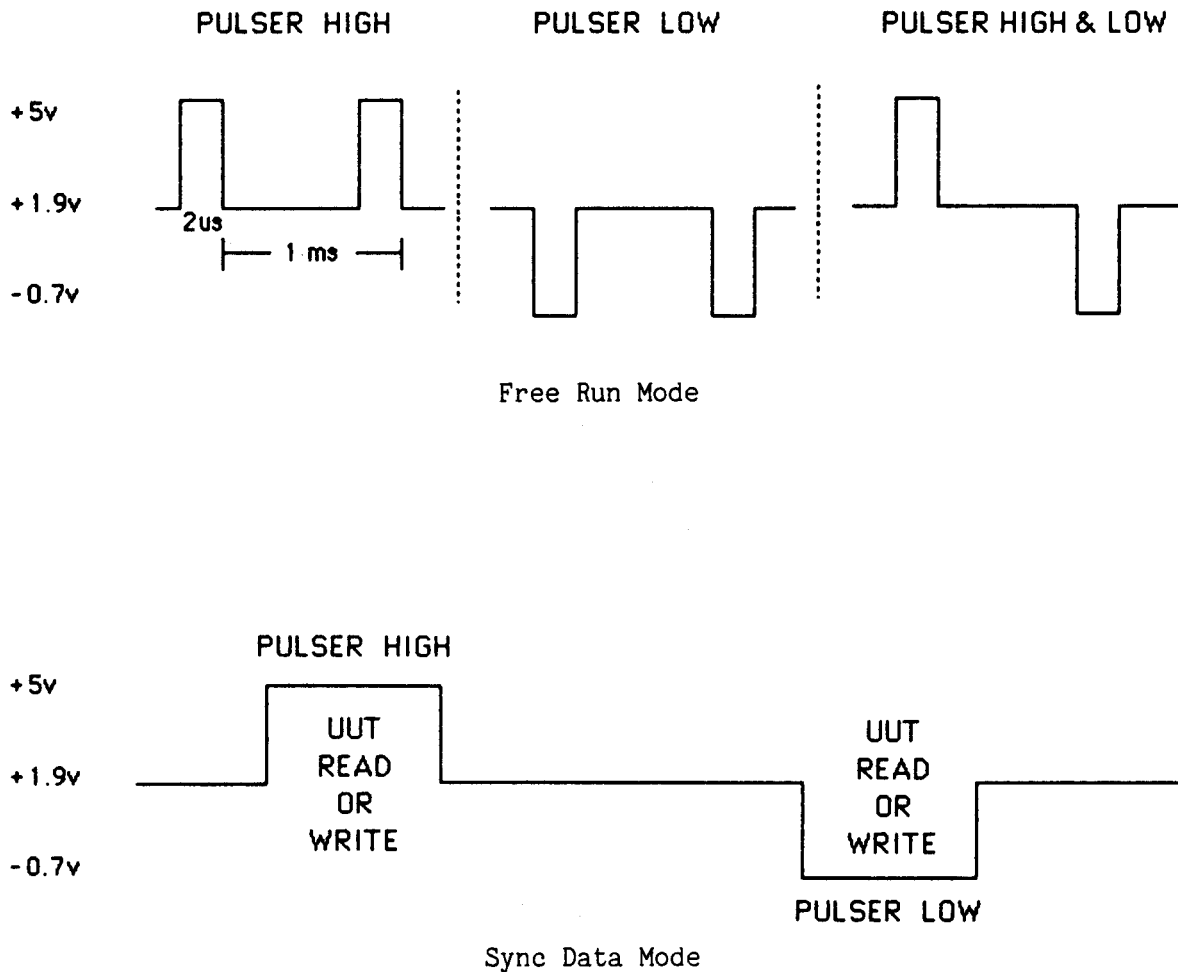


Figure 3. Pulser Timing Diagram

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

Using the Probe to Measure and Stimulate the I/O Ports

1. Synchronize the probe.
2. Stimulate Port 10000 LEDs with a bit pattern.
3. Loop on Step 2 and measure the Port's off-bus side (Q0-Q7) with the probe. Measure the chip enable (CLK) also.
4. Do a looping Read at the Port 10000 switches.
5. Set Port 10000 switches to their open position and pulse the bits on the off-bus side (I0-I7) low.
6. Do Step 5 in the free run mode.

Using the 9010A Registers

1. WRITE 8C00 = 55 ENTER
2. REG E ENTER
 - a. What is in Reg E?
 - b. Look at the reference card for definition of Reg E.
3. REG F ENTER
 - a. What is in Reg F?
 - b. Look at the reference card for definition of Reg F.
4. Use reference card to determine functions of other registers.

Instructor-Led Exercise

Programming

1. Press PROGM key. Enter #0.
2. Press DISPL key.
 - a. Type your name.
 - b. Press PRIOR key to delete one letter.
 - c. Press ENTER to enter this display message.
3. Press PROGM Key. This will close the program.
4. Press EXEC Key. Enter #0.
5. Press STOP Key.
6. Press PROGM key followed by ENTER key to re-open Program #0.
 - a. Press MORE key until your name is displayed.
 - b. Press CLEAR key.
 - c. Press DISPL key and type another message.
 - d. Press ENTER to enter new display message.
7. Close program and execute.

Loading the Async Option Programs

1. Insert Async Option cassette tape into tape drive.
2. Press READ TAPE.
3. Press YES.
4. After READ TAPE OK prompt, press PROG key, then " = " key.
5. Verify that programs 1-9, and 11-20 have been loaded.

FUNCTIONAL TESTS FOR THE BASIC I/O

Program 21

```
DPY-I/O PORT 10000 TEST (display the test header)
LABEL 1 (reference label for the "goto" statement)
READ @ 10000 (read the switch data)
WRITE @ 10000 = REGE (write the switch data to the LEDs)
GOTO 1 (loop back to label 1)
```

Program 21 will verify that I/O Port 10000 is functioning properly. This is accomplished by reading the data word determined by the switch positions, then writing this data word to the LEDs. After doing the Read @ 10000, the data is stored in Register E which is the Data Register. When the Write @ 10000 = REGE is done, Register E will still contain the same data gathered during the Read @ 10000. This allows you to flip one of the switches and see the corresponding LED turn on.

Program 22

```
LABEL 1 (reference label for "goto" statement)
DPY-ENTER 7 SEG DPY NO.<0-19>/E (prompts operator to enter no.)
IF REGE > 19 GOTO 1 (goto label 1 if no. entered is too big)
WRITE @ 10001 = REGE (write entered data to 7-seg. display)
GOTO 1 (return to label 1)
```

Program 22 will test the two 7-segment displays of the TK-80 basic I/O. This is accomplished by using the / character at the end of the display message which prompts the operator for the data word that is to be written out to the 7-segment display. The E following the / designates which of the 16 registers (0-F) will be used to store the entered value. The "If" statement (conditional branch) will cause the program to branch back to Label 1 if the operator entered a number greater than 19. The WRITE @ 10000 = REGE will write the data word stored in Register E to the 7-segment display.

Remember, entering a data word between 00 and 09 will put a decimal number on the right display and a data word between 10 and 19 will put a decimal number on the left display. Hex digits A-F will be ignored by the display's BCD Decoder/Driver and data words greater than 19 will not be accepted by the program.

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

Writing and Reading a Cassette Tape

1. Insert Cassette Tape.
2. Press WRITE TAPE.
3. Press YES.
4. After completion of Writing a Tape:
 - a. Cycle 9010A power.
 - b. What is in the memory map?
 - c. Is your program still in the 9010A?
 - d. Has the setup changed?
5. Press READ TAPE.
6. Press YES.
7. After completion of Reading a Tape:
 - a. Verify that the memory map is correct.
 - b. Press PROGM key followed by = key to display which programs have been loaded.
 - c. Verify that your program will execute.

PROGRAMMING THE MICRO-SYSTEM TROUBLESHOOTER

So far you've been executing Troubleshooter commands in the immediate mode, i.e. the command is executed immediately after it is entered. In this section you will be recording what are mostly immediate mode steps, such as reads and writes, into programs. In addition to entering immediate steps, there are only five keywords you must know in order to; write display statements (DISPL key), do unconditional branching (GOTO & EXEC keys), and do conditional branching (IF & GOTO keys). There are also some arithmetic keys that allow data in the sixteen available registers to be operated on logically and arithmetically. The keywords are explained below.

Display statements can be used to display any message up to 32 characters. Special characters within a Display statement allow you to display the contents of a register in decimal or hex. Special characters will also allow the operator to place a decimal or hex number into a register. When the DISPL key is pressed then the ASCII characters to the bottom right of the keys will be the characters seen on the display. For example, pressing the READ key in a Display statement will put a "G" character on the display.

Unconditional Branching is accomplished using either the GOTO or EXECUTE statement. Their formats are listed below.

EXECUTE XX where XX is a program number 0-99 decimal.

GOTO X where x is a reference label numbered 0-F hex.

LABEL X is used with the GOTO statement and can be placed anywhere in a program.

Conditional Branching occurs when the IF statement is true, causing the program to branch to a Label number. Otherwise, it will execute the next step in the program. The format of the IF statement is shown below.

IF (an expression) =, or >, or >= (an expression) GOTO (a label no.)

where the "expression" can be a hex number, a register number, an arithmetic function, or a combination of the three.

Programming the Micro-System Troubleshooter

The following steps explain how to create, edit, close, and execute a program. You will need to know these steps for the programs in the later part of this section.

To create a program:

1. Press the PROGM key.
2. Enter a number 0-99 decimal.
3. Press the ENTER key.

You are now ready to enter program statements. You will know if you're in the program mode by the flashing LED next to the Programming annunciator at the right of the display.

Editing a program:

1. The procedure for opening a program for editing is the same as creating a program.
2. Use the MORE and PRIOR keys to move between program lines. Pressing the number of a label (the number only) will take you to where that label is placed in the program.
3. To delete a line, move that line to the display then press the CLEAR/NO key.
4. To insert a line, move the line prior to where the inserted line will go onto the display. Now enter the new line.
5. Use the PRIOR key to delete a character in a display statement.

Closing a program:

1. Press the PROGM key.

Don't forget this step. If you try to execute the program before closing it then that command becomes part of your program.

Executing a program:

1. Press the EXEC key.
2. Enter the program number.
3. Press the ENTER key.

If you want a program to repeat itself continuously, press the LOOP key. Press the STOP key to stop the program.

9010A HANDS-ON TRAINING

Exercise 8A

FAULT: I/O PORT BUS FS #8
IDENTIFY USING: FUNCTIONAL TEST, PROGRAM 21
ISOLATE USING: WRITE (STIMULUS)
READ, PROBE (MEASUREMENT)

1. Set fault switch #8 to its fault position.
2. Perform an AUTO TEST. What is being tested during this Test?
3. Identify the fault using the Functional Test in PROGRAM 21.
4. To further identify the fault, perform a READ and WRITE at this Port. List your results:
5. To isolate this fault, LOOP on a function in Step 4 that identified the fault and troubleshoot the defective circuit using the PROBE. Should the PROBE be synchronized? Does it matter?

Record the nodes you check and the logic levels you observe:

6. Where is the defect and what type did you discover?
7. Verify your answer to Step 5 by setting fault switch #8 to its no-fault position and repeating Step 4.

8. Reset fault switch #8 to its fault position. Instead of using the Port input switches to stimulate this I/O circuit for troubleshooting purposes, use the PROBE as a PULSER while doing a LOOPING READ at Port 10000.

Should the PROBE be synchronized? Does it matter?

Try locating this I/O circuit fault using the PULSER method.

9. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. Why was it necessary to perform the READs and WRITEs at the input port and the output port after performing the functional test?
 - b. Does it matter if you use a READ or WRITE as the stimulus for isolating this fault?
 - c. How does using the WRITE function and the PROBE give you a clue as to the type of fault?
 - d. How does using the READ function and the PROBE give you a clue as to the type of fault?

STATUS LINES

Some status lines are also referred to as forcing lines because they "force" the uP into an idle state. The status lines that are considered forcing lines are marked with an asterisk on the pod decal. Some of these forcing lines are "enableable" meaning that they can be either enabled or disabled using the Setup selections.

If a forcing line becomes active or the system clock fails, then the pod uP is forced into an idle state. When this happens the pod is no longer able to report back to the mainframe and the following message appears:

POD TIMEOUT - ATTEMPTING RESET

This message remains until either the system clock is restored, or the forcing line is no longer active, depending on which condition caused the pod timeout. Presence of the system clock can be verified with the probe in the free run mode. This will give you a quick check to see whether the clock is completely gone or not. If the system clock is present you would then check for active forcing lines using the following procedure:

1. Check the pod decal to see which status lines are also forcing lines.
2. Use the SETUP key then the MORE and PRIOR keys to scroll through the Setup selections to locate the SET-ENABLE (forcing line) ? messages.
3. Answer no using the CLEAR/NO key to disable the forcing lines.
4. Repeat the operation that produced the Pod Timeout. ACTIVE FORCE LINE - LOOP? should be displayed.
5. Press the MORE key to find out which forcing line is active. A "1" in a bit position of the message: STS BITS bbbb bbbb indicate an active forcing line. The bits in the message correspond directly with Status Bit definitions on the Pod Decal.
6. Go back to the Setup selections and re-enable the forcing lines that aren't active (optional).
7. Scroll through the Setup selections and find the SET-TRAP ACTIVE FORCE LINE ? message. Answer no. This prevents the ACTIVE FORCE LINE - LOOP ? message encountered in Step 4 from reappearing.

9010A HANDS-ON TRAINING

Exercise 9A

FAULT:	STATUS LINE	FS #16
IDENTIFY USING:	BUS TEST	
VERIFY USING:	PROBE, FREE RUN	

1. Set fault switch #16 to its fault position.
2. Press BUS TEST. What is the meaning of the error message?
3. Using the POD Label, determine the names of the STATUS LINES that are FORCING LINES.
4. Using the SETUP key, disable those FORCING LINES identified in the previous step which can be disabled.
5. Press BUS TEST. Record the 9010A display message. Use the MORE key to get the second line of the message.
6. From the message in Step 5, identify which STATUS LINE is active and locate on the UUT schematic possible locations for this fault.
7. Use the PROBE in the FREE-RUN mode to verify this fault on the UUT.
8. Using the SETUP key, answer NO to SET-TRAP ACTIVE FORCE LINE?
9. Repeat BUS TEST. What happens this time? Why?

10. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How is a forcing line defined?

 - b. Which step prevented the active forcing line from being seen by the uP in the Pod?

 - c. The step identified in question "10.b" prevents the active force line from being seen by the uP in the Pod, but the Active Force Line message is still reported to the 9010A display. Which step prevented the Active Force Line message from being reported to the 9010A display and allowed you to continue testing?

USING PROBE DATA IN A PROGRAM TO TROUBLESHOOT I/O

In a previous section you used functional tests to see whether the I/O Ports were operating properly. If the tests detected a failure then you used the probe with Troubleshooter Read and Write commands to stimulate and measure the circuit while in the immediate mode. Up to now the data being measured has been in the form of logic levels. You can even synchronize the probe to examine logic levels only when data is valid. But sometimes it is necessary to examine a "stream" of data to diagnose intermittent or pattern sensitive errors. To do this requires more complex forms of stimulus and ways of examining the data other than observing the probe lights.

CRC signatures and transition counts are the probe data normally used when measuring data streams at a node. Reading logic levels will only tell you if the node went high or low - they won't tell you if the data stream is correct or not. The signature is capable of detecting one bit errors 99.9999% of the time. You would follow the steps listed below to gather signatures in the immediate mode.

1. Sync Probe to Data.
2. Press the READ PROBE key.
3. Stimulate the node (i.e. RAMP, BUS Test, etc).
4. Press the READ PROBE key.

In the second step READ PROBE will clear out the register where the probe data is stored. In Step 3, a stimulus is needed that will provide a series of bit patterns that can be repeated later on. The Ramp function makes a good stimulus for data lines because it provides a unique signature for each line. The Ramp function does 256 writes starting at 00 and incrementing to FF. The fourth step ends the data gathering cycle and stores the probe data into Register 0 (see 9010A Reference Card).

If you are going to collect a lot of signatures, say all eight data bits at five different ICs, then repeating Steps 2-4 above each time becomes rather tedious. Writing these steps into a program will save a considerable amount of time.

Using Probe Data in a Program to Troubleshoot I/O

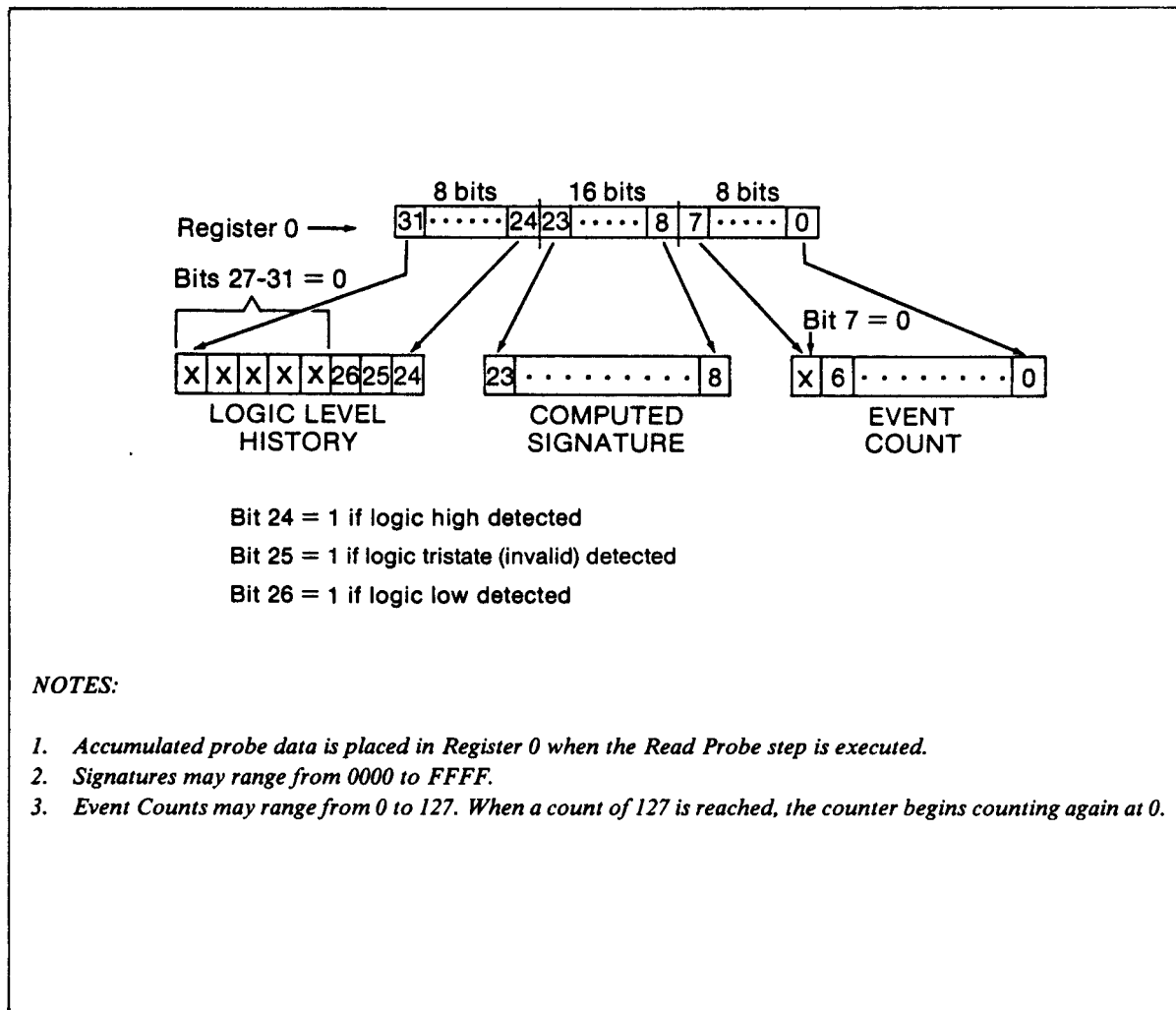


Figure 4. Probe Response Data in Register 0

INSTRUCTOR-LED EXERCISE

Gathering Signatures

1. Sync probe to data.
2. Probe buffered data bit 0.
3. Clear the probe register (Reg 0). Press READ PROBE Key.
4. Stimulate the node. RAMP @ 0
5. End data gathering cycle. Press READ PROBE Key.
6. Mask out the Signature using the Immediate Mode.
 - a. Press REG 0. (Do not press ENTER key here.)
 - b. Press SHR Key eight times.
 - c. Press AND FFFF ENTER.

PROGRAM 23

```

DPY-ENTER STIMULUS ADDRESS /F (operator designates stimulus address)
SYNC D (synchronize the probe)
LABEL 1 (reference label for goto statement)
READ PROBE (clear the probe register)
RAMP @ REG F (stimulate the node)
READ PROBE (latch the data)
REG 1 = REG 0 SHR SHR SHR SHR SHR SHR SHR SHR AND FFFF (mask out signature)
DPY- SIGNATURE = $1 (display the signature)
GOTO 1 (return to label 1)

```

In Figure 4 you can see that after a Read Probe operation Register 0 contains the count in bits 0-6, the signature in bits 8-23, and the level history in bits 24-26. Bits 7 & 27-31 are always 0. The program above will perform the steps outlined earlier for getting probe data into register 0. After probe data is in the register, the program strips out the signature and displays it.

INSTRUCTOR-LED EXERCISE

Using Program 23

1. Execute program 23.
2. Input stimulus address.
3. Monitor the following nodes with the Probe.
 - a. Buffered data bit 0 signature = _____
 - b. " " " 3 " = _____
 - c. " " " 7 " = _____
4. Compare results.

9010A HANDS-ON TRAINING

Exercise 10A

FAULT: OUTPUT PORT BUS FS #14
 IDENTIFY USING: RAMP (STIMULUS)
 PROBE SIGNATURES (MEASUREMENT)

1. Use PROGRAM 23 to gather signatures for Q0 through Q7 Output Port 10000 data lines. Record your results in the table below in the column for signatures without a fault.

DATA LINE	SIGNATURE WITHOUT FAULT	SIGNATURE WITH FAULT
Q0		
Q1		
Q2		
Q3		
Q4		
Q5		
Q6		
Q7		

2. Set fault switch #14 to its fault position.
3. Repeat Step 1 and record results. (Signature with Fault)
4. Compare the signatures. What does this indicate?

To verify your answer, WRITE a data pattern to Port 10000 where:

- a. One of the tied bits is low and the rest high. Observe the LEDs.
- b. One of the tied bits is high and the rest low. Observe the LEDs.

5. To review what you have learned from this exercise about using the 9010A PROBE, answer the following questions:
 - a. In Step 4, would WRITING a 55 or a AA have verified the fault if D7 and D5 were shorted together?
 - b. Why do signatures off the bus differ from signatures on the bus when using identical stimulus?

Section XI
Peripheral Interface Adapter (PIA)

THE PERIPHERAL INTERFACE ADAPTER (PIA)

The Peripheral Interface Adapter (PIA) is located on the TK-80 board and is designated as U30. It has three ports and each port can be configured as either for input or output. The direction of each port, as well as its mode of operation (basic, strobed, or bi-directional), is determined by writing a control word to the control register of the PIA. Listed below are the port addresses, direction, mode of operation, and the control word that configures them for the TK-80 board application.

	ADDRESS	DIRECTION	MODE
Port A	100F8	Input	Basic
Port B	100F9	Input	Basic
Port C	100FA	Output	Basic
Control	100FB = 92 (for directions and mode listed above)		

As you can see in the table above, all three I/O ports are used in the basic mode of operation. When used in the basic mode, the data remains latched in the port buffer until new data is written to, or read from, the port. The other two modes, strobed and bi-directional, will not be used in this course.

The Peripheral Interface Adapter (PIA)

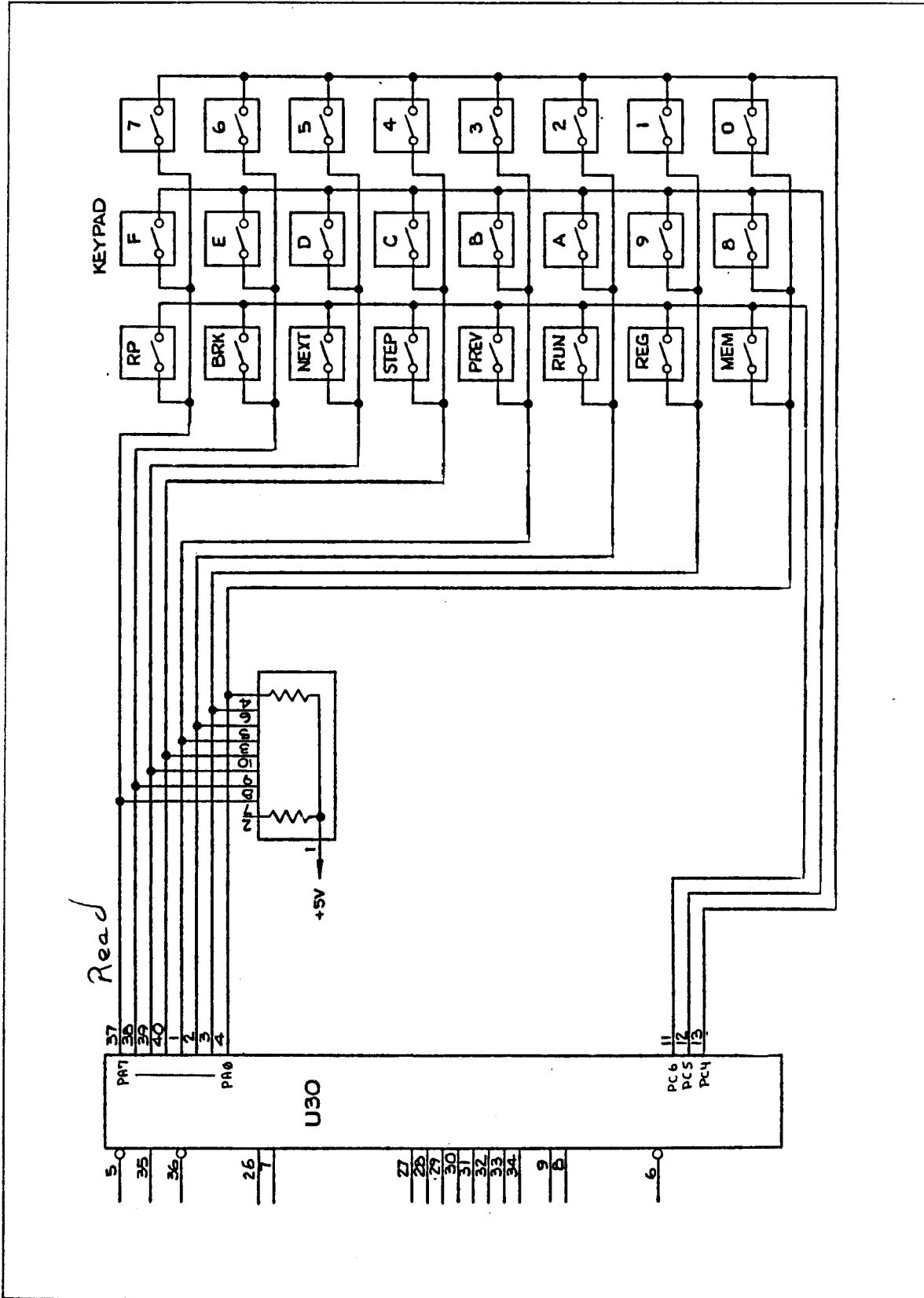


Figure 5. TK-80 PIA

The Peripheral Interface Adapter (PIA)

Port A is used for reading the keys on the TK-80 board (see Figure 5). The PA0-PA7 bits are connected to the keys and have an I/O address of 100F8. The following procedure explains how the keys are read.

1. Configure the PIA with the control word. WRITE @ 100FB=92.
2. Enable the column of keys you wish to read by writing a "0" to the Port C bit connected to that column.
3. Do a looping read at Port A.
4. Press a key in the enabled column.

Port B is an input port used for reading a set of switches on the I/O board. The switches are labeled with their I/O address of 100F9. The switch position to the far right is the least significant bit (LSB), incrementing to the most significant bit (MSB) to the far left. The switches are read by doing a READ @ 100F9.

Port C is an output port that is used for enabling the key columns of the keyboard that are connected to Port A. It is also used for turning on a set of LEDs on the I/O board labeled with the I/O address of 100FA. The LSB is connected to the LED to the far right, while the MSB is connected to LED to the far left. To turn on the Port C LEDs, do a WRITE @ 100FA with a "1" in the bit positions you wish to light.

Instructor-Led Exercise

INSTRUCTOR-LED EXERCISE

Exercising the TK-80 PIA Ports

1. WRITE a 92 to Control Register 100FB.
2. WRITE a FF to Port 100FA. Observe the LEDs.
3. WRITE a 00 to Port 100FA. Observe the LEDs.
4. WRITE a 55 to Port 100FA. Observe the LEDs.
5. Perform a looping READ from Port 100F9.
 - a. Change positions of the switches.
 - b. Observe the 9010 display.
6. Inputing from the Keyboard.
 - a. WRITE a EF to Port 100FA.
 - b. Perform a looping READ from Port 100F8.
 - c. Test keys 0 through 7.
 - d. WRITE a DF to Port 100FA.
 - e. Perform a looping READ from Port 100F8.
 - f. Test keys 8 through F.
 - g. WRITE a BF to Port 100FA.
 - h. Perform a looping READ from Port 100F8.
 - i. Test keys: MEM, REG, RUN, PREV, STEP, NEXT, BRK, and RP.

NOTE

The UUT RESET key cannot be read through the PIA. If the UUT RESET key is pressed, you must re-write the control word to the PIA.

FUNCTIONAL TESTS FOR THE PIA

The program listed below is similar to Program 21 for Port 10000 of the TK-80 basic I/O. The difference is the test header, the control word written to the PIA that configures it, and the port numbers.

PROGRAM 24

```
DPY-PIA PORTS B+C TEST      (test header for Ports B and C)
WRITE @ 100FB = 92          (configure PIA for mode and port direction)
LABEL 1                    (reference label for goto statement)
READ @ 100F9                (read Port B switches)
WRITE @ 100FA = REGE        (write switch data to Port C LEDs)
GOTO 1                      (goto label 1)
```

This program is implemented the same as Program 21. After Program 24 has been executed flip one of the Port B switches and look for the corresponding Port C LED to turn on.

9010A HANDS-ON TRAINING

Exercise 11A

FAULT: I/O PORT FS #11
IDENTIFY USING: FUNCTIONAL TEST, PROGRAM 24
ISOLATE USING: READ (STIMULUS)
PROBE (MEASUREMENT)

1. Set fault switch #11 to its fault position.
2. Identify this fault using AUTO TEST and the Functional Test in PROGRAM 24.
3. To further identify the fault, perform a WRITE and READ at these Ports. List your results:
4. While using a LOOPING WRITE or a LOOPING READ stimulus (choose the correct one), isolate the fault using the PROBE. Record the nodes you check and the logic levels you observe:
5. Where is the defect and what type did you discover?
6. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. When using the Functional Test, how could you tell which Port bit was defective?
 - b. Why didn't the Functional Test identify whether the defect was an input or output port bit?
 - c. Why did you use the particular stimulus you chose in Step 4 for isolating the fault?

Exercise 11B

FAULT: UUT KEYBOARD FS #12
IDENTIFY USING: FUNCTIONAL TEST, IMMEDIATE MODE

1. Set fault switch #12 to its fault position.
2. To test the UUT keyboard, use the immediate mode. Configure the PIA chip by writing a 92 to its Control Register. Enable one group of keys at a time by WRITING to Port C of the PIA with a BF, DF, and EF while performing a LOOPING READ at Port A. Test the Keys.
3. Which key or keys is defective?
4. Verify your answer to Step 3 by setting fault switch #12 to its no-fault position and repeating the above functional test for the column containing the defective key.
5. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. Does it matter which logic level is sent out for bit seven of Port C in order to enable the desired keys on the keyboard? Why?
 - b. From your functional testing, what are the possible types of defects for the faulty key(s)?

WHY THE ASYNCHRONOUS SIGNATURE PROBE OPTION?

The Fluke 9000A Series Micro-System Troubleshooter is the best piece of equipment available for testing synchronous activity generated by a microprocessor. The ability to synchronize the probe and measure signals in time with Address or Data cycles is one of the most powerful features of the Micro-System Troubleshooter. But we have been limited in our testing ability to measure activity that was asynchronous to the microprocessor. Signature analysis has long been recognized as a viable method of testing digital circuits and it is the best method available for testing asynchronous activity. The Asynchronous Signature Probe option allows us to test any digital circuit using signature analysis.

WHAT IS SIGNATURE ANALYSIS?

In its simplest form, signature analysis is a method for testing a digital stream of data at a particular test node. If the data is consistent, if our viewing window remains constant, and if the method of clocking in data is stable, then we will find a repetitive number or signature formed for that test node. It is referred to as a signature because it is characteristic of that node, just like your written signature is a characteristic of you. The repeatability of the data stream at the node is what makes signature analysis so valuable. There is a confidence level of 99.998% of finding an error if only one bit of data changes in the stream of data that crosses the node.

HOW IS THE SIGNATURE CREATED?

Data is captured with the probe and sent to a sixteen-bit shift register. As the data is shifted into the register it is also shifted back upon itself through a series of exclusive OR gates (Figure 1). It is not so important that you understand the movement of each data bit through this circuit but you should be aware of what type of circuit is being used. This procedure is called a cyclic redundancy check (CRC) and is a universally accepted technique for creating signatures. Each unique stream of data creates an equally unique number in the sixteen-bit register. This number is converted to a four-digit hex number which is called the signature.

When is the Signature Created?

WHEN IS THE SIGNATURE CREATED?

The critical factor in signature analysis is the timing involved when taking the signature. The data must be captured during a known and repeatable time frame. To accomplish this we reference our data gathering to some timing activity, usually determined by the circuit under test. The first thing that must be done is to establish a window or gate period during which we will sample data.

This is done by selecting a Start and a Stop signal on the circuit. The Start determines when the window will open and the Stop determines when it will close. Usually the same signal is used for both Start and Stop, with the operator having the capability of selecting either the rising edge of the signal or the falling edge of the signal as the trigger. In the drawing in Figure 6 we have selected the falling edge for the trigger on both Start and Stop. This gives us a window period from time B, the falling edge of the Start signal, to time D, the falling edge of the Stop signal.

Remember that the strength of signature analysis is its repeatability. But this requires that we use the same reference points each time. If we trigger the Start on the rising edge, point A, we would now have a window from time A to time C. This would result in a different signature than the previous example because we no longer see the data bit that occurred between point A and B.

If we had selected our Start as the rising edge and our Stop as the falling edge of the signal our window would have occurred between time A and time B. The Stop occurs on the next selected transition that happens and in this case would stop at time B rather than time D. This would have created too small a window to get a good look at the data. So it is important that you understand your circuit and trigger your start and stop properly. And keep your reference points consistent between tests to maintain repeatability.

We can further qualify the window or gate period using an Enable signal. If the Enable signal is set to always active, our window consists of the time period between the Start and Stop signal as discussed above. If we trigger on a high Enable, then our window is valid whenever the Enable signal goes high. If we trigger on a low Enable, the window is valid whenever the Enable signal is low. In most examples the Enable signal will always be active, that is, not showing any preference for a high or a low.

We must next select a signal that will clock our data through this window. The system or UUT clock is usually the source for this signal. Each time a clock signal occurs, during the window we have created with the Start and Stop signal, a data sample is taken by the probe. As with the Start and Stop signal, we can trigger the sampling clock on the rising or falling edge of the signal. For increased sampling rates we can trigger on both the rising and falling edge. In Figure 6 we are clocking on the falling edge of the clock signal and will take five data samples during the window we have created between points A and D.

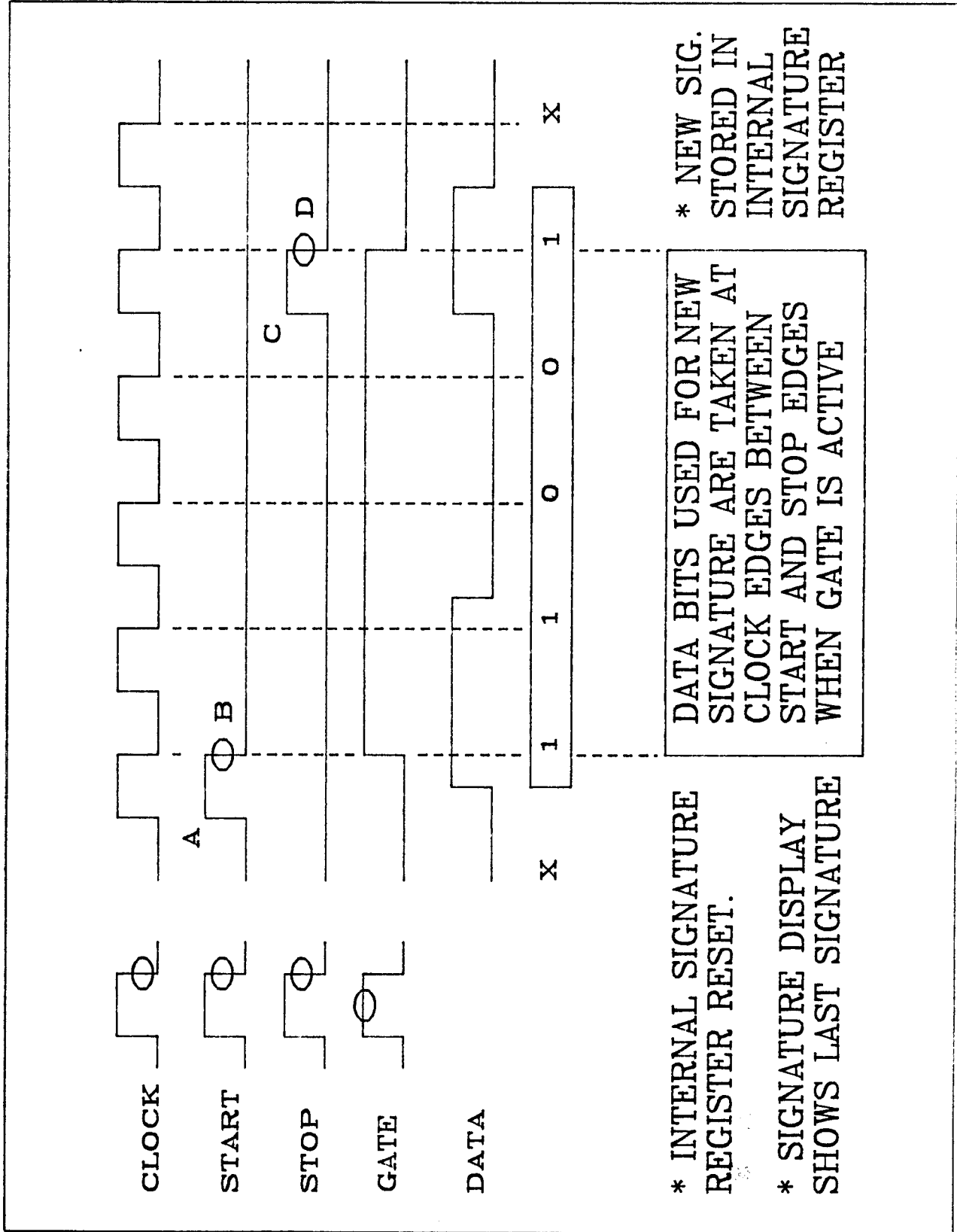


Figure 6. Basic Gate Operation

How to Use the Asynchronous Signature Probe Option

HOW TO USE THE ASYNCHRONOUS SIGNATURE PROBE OPTION

The first step is to attach the clock module to the 9000A series mainframe. The connector is the same type that is used with the interface pods and is connected the same way. Next load the cassette tape that contains the software to drive the Asynchronous Signature programs. Then execute program 0, a continuously running program that measures signatures, take counts, and displays waveforms.

SETUP CONDITIONS

There are several setup instructions that must be implemented before taking signatures.

Start

Pressing the SETUP key brings up the first setup command which is the Start function. This determines the characteristics of the Start signal you will use for the signature. The number one (1) key can be used to select whether you trigger the Start on the rising or falling edge of the signal you have picked. The number two (2) key is used to select between an External Start, one that comes from the circuit where you have the green start lead connected, or Pod-sync where the Start is triggered from the sync signal generated by the Pod. Pod Sync is seldom used with the clock module because if you are relying on Pod Sync it is usually easier to use the 9000A series original built-in signature capability.

Stop

Pressing the More key now brings up the next setup command which is the Stop characteristics. Again you can press the number one (1) key to determine whether you trigger on the rising or falling edge of your selected Stop signal. And the number two (2) key selects as the source of the Stop, the External Stop signal connected to the red lead on the clock module, or Pod Sync as described above.

Clock

The More key now brings up the setup for the Clock signal characteristics. There are three choices for triggering that can all be selected by pushing the number one (1) key. You can select either to trigger the clock on the rising edge, the falling edge, or to trigger on both the rising and falling edges which effectively doubles your number of samples. The number two (2) key again selects the source of the clock signal, either the External Clock (connected to the yellow lead of the clock module) or Pod Sync (from the pod).

Enable

The next setup instruction determines the condition of the Enable line. The default condition comes up with this set to Always which means the window is always enabled, as determined by the Start and Stop signals. If you press the number one (1) key at this time you are now looking at the external enable signal connected to the blue lead of the clock module. You now have an additional choice, with the number two (2) key, of whether to open the window when the enable signal is high or when the enable signal is low.

Stop Counter

The More key now brings up the setup for a special feature called Stop Counter. This affects the Stop signal, the point at which you stop gathering data. In its default condition the Stop Counter feature is disabled. This means the data gathering stops in conjunction with your previously selected Stop signal. If the number one (1) key is pressed you now enable the Stop Counter. This feature allows you to select how many clock triggers occur after the Start signal, before you stop gathering data. You can select a Stop Count of up to 4095 (decimal) by selecting that number and pressing the enter key. To select a new Stop Count you press the number two (2) key. To disable this feature, press the number one (1) key again.

Events Count

The last setup instruction determines when events are counted. The Asynchronous Signature option measures upward transitions which are opposite of the transitions measured by the mainframe. The default value of this setup selects Gated Counts which means all the counts that occur between the Start and Stop signals are recorded. If the number one (1) key is pressed you will measure Free Running Counts which means that all the counts that occurred since the last Read Probe will be recorded. This second condition is very much like the existing counter in the 9000A series mainframe.

SAVING SETUP CONDITIONS

After you have selected the values for the setup conditions for your particular UUT, they are automatically stored in Register 8. You can read Register 8 at this time and record the code for these particular setup conditions. Now if you wish to return to the signature gathering mode you would execute Program 1 which contains the last setup commands you created. If you were to execute Program 0 instead, the setup commands would go back to their default values.

Signature Gathering

SIGNATURE GATHERING

Now you would connect the appropriate leads from the clock module to the selected test points on the circuit board. It is important that the ground lead on the clock module be connected to an appropriate ground point on the circuit being tested.

To gather the signature and count you press the Read Probe key on the mainframe. The display will show "WAIT" for about three seconds then display the signature and count for the selected test point. To gather another signature, just press the Read Probe key again.

WAVEFORM CAPTURE

Another feature of this option is the capture of digital waveforms. The last 640 nanoseconds of data that occurred immediately prior to the Stop signal can be displayed. To see this signal, after the signature has been displayed, press the More key on the mainframe. The (-) sign represents a high, the (=) represents tristate, and the (.) represents a low. Each character represents the level present at the end of a twenty nanosecond window. Taken together, the thirty two display bits show 640 nanoseconds of time.

This feature is very useful for showing intricate timing relationships. You would select for your Stop signal the point you want to use for a reference and then probe the signal you wish to compare to this reference. The reference signal would exist 20 nanoseconds after the display has ended and the measured signal would appear on the display if it occurred within 640 nanoseconds of the selected Stop signal.

ASYNCHRONOUS SIGNATURE PROGRAMS

PROGRAM NAME	PROGRAM NUMBER/ TAPE SIDE	DESCRIPTION
Initialize	0/A	A program that loads the initialization register (register 8) with the default hardware (module) setup parameters then executes the Interactive Operation Program.
Interactive Operation	1/A	A program that executes other programs in response to front panel keys, to allow interactive user operation of the Troubleshooter/Signature Module.
Service Gate Keys	2/A	A program that executes other programs in response to the front panel READ, READ PROBE, and CONT keys.
Service SETUP Key	3/A	A program that executes other programs to initialize the module when the Setup Hardware Program is executed.
Display Gate	4/A	A program that monitors the occurrence of the start, stop, and clock; and indicates the condition of these signals on the Troubleshooter display.
Start Setup	5/A	A program called upon by the Service SETUP Key Program to allow operator selection of the START signal.
Stop Setup	6/A	A program called upon by the Service SETUP Key Program to allow operator selection of the STOP signal.
Clock Setup	7/A	A program called upon by the Service SETUP Key Program to allow operator selection of the CLOCK signal.
Enable Setup	8/A	A program called upon by the Service SETUP Key Program to allow operator selection of the ENABLE signal.
Stop Count Setup	9/A	A program called upon by the Service SETUP Key Program to allow operator selection of gated CLOCK pulses for stop.

ASYNCHRONOUS SIGNATURE PROGRAMS (cont)

PROGRAM NAME	PROGRAM NUMBER/ TAPE SIDE	DESCRIPTION
Delay Setup	10/A	A program called upon by the Service SETUP Key Program to allow operator selection of the delay compensation.
Event Setup	11/A	A program called upon by the Service Setup Program to allow operator selection of the event count parameters.
Setup Hardware	12/A	A binary program that initializes the module in accordance with the contents of register 8 (initialization register) by sending a series of op codes to the module.
Arm Gate	13/A	A binary program that resets all registers within the module and then arms it to receive the control signals; START, STOP, and CLOCK.
Get Signature	14/A	A binary program that retrieves the signature data from the module and places it in the B register.
Get Events	15/A	A binary program that retrieves the events data from the module and places it in the B register.
Get Waveform	16/A	A binary program that retrieves the waveform data from the module and places it in the B register.
Read Data and Status	17/A	A binary program that directs the Signature Module to read one nibble of data and one nibble of status.
Send Op Code	18/A	A binary program that sends one byte over the pod bus to the Signature Module.
Display Waveform	19/A	A program that presents the contents of the two waveform data registers on the Troubleshooter display.

ASYNCHRONOUS SIGNATURE PROGRAMS (cont)

PROGRAM NAME	PROGRAM NUMBER/ TAPE SIDE	DESCRIPTION
Append Signature	20/A	A utility program that appends the value in user register B to the display in signature format.
Merge Tape Program	0/B	A utility program that allows you to combine programs from different 9010A tapes onto one.
Stimulus Program A	1/B	A program used to stimulate the control and gate sections of the Signature Module during module troubleshooting procedures outlined in the Service Manual.
Stimulus Program B	2/B	A program used to stimulate the signature, events, and waveform sections of the Signature Module during module troubleshooting procedures outline in the Service Manual.

Signatures and Counts for the TK-80 Display Circuitry

SIGNATURES AND COUNTS FOR THE TK-80 DISPLAY CIRCUITRY

In order to collect consistent and repeatable signatures in the TK-80 Display Circuitry, certain bits of the PIA must be set to known levels. A logic high on PC7 is needed to turn on the TK-80 display, and a logic high on PC0 will select one of two available clock frequencies that is divided down to generate Display Request (DISREQ).

Also, a known pattern should be placed in the Display RAM so repeatable Signatures can be taken on U21. This is accomplished by lines five through eleven in the program below. Shifting the "1" in the Data Register (REG E) to the left each time through the loop will cause each Data line to toggle, therefore creating a unique Signature for each line.

As you will probably be doing this step more than once, a program has been outlined below that will perform the necessary setups.

PROGRAM 25

```
WRITE @ 100FB = 80      (configure the PIA)
WRITE @ 100FA = 81      (set the Port C bits)
WRITE @ 100F9 = 00      (toggle Port B)
WRITE @ 100F9 = 80      (toggle Port B)
REGF = 8FF8             (load address reg. with 1st display address)
REGE = 1                (set bit 0 in data reg)
LABEL 1                (reference label for goto statement)
WRITE @ REGF = REGE     (write data to 1st display address)
INC REGF                (increment address reg. to next dpy. address)
SHL REGE                (set the next bit in the data reg.)
IF 9000 > REGF GOTO 1   (is this the last display address?)
```

Listed below are the Clock Module timing setups for collecting Signatures and Counts for this circuit.

Start	Falling edge	Ext. Start	U54-11
Stop	Falling edge	Ext. Stop	U54-11
Clock	Falling edge	Ext. Clock	U52-3
Enable	On Ext. Enable	High	U52-6
Stop Count	Disabled		
Events Count	Gated		

You can also select the proper setups by adding the following lines to the end of Program 25.

```
REG 8 = 5000128        (fill reg. 8 with setups listed above)
EXECUTE PROGRAM 12     (setup the hardware according to reg. 8)
EXECUTE PROGRAM 1      (initiate the interactive mode)
```

TEST POINT	SIGNATURE	COUNT	TEST POINT	SIGNATURE	COUNT
U54-2	CFE7	16	U53-1	5D46	16
-11	924C	1	-2	5D46	0
-12	F763	2	-3	CFE7	16
-13	5581	4	-4	5D46	0
-14	E931	8	-5	XXXX *	16
-15	BC11	1	-6	XXXX *	16
			-8	5D46	16
U26-1	5D46	16	-9	0000	16
-2	F763	2	-10	5D46	0
-3	0000	0	-11	XXXX *	8003
-4	5581	4	-12	XXXX *	16
-5	0000	0	-13	XXXX *	2662
-6	E931	8			
-7	0000	0	U20-1	0000	0
-8	0000	0	-2	CFE7	16
			-3	92A1	16
U55-1	19F3	4			
-2	BOA7	4	U27-4	92A1	16
-3	49E0	4	-5	5D46	0
-4	F1B1	4	-6	CFE7	16
-5	2CDD	4			
-6	7AF0	4	U52-1	0000	16
-7	85CE	4	-2	5D46	16
-9	5047	4	-3	0000	26608
-10	DE1E	2	-4	5D46	0
-11	D1A2	2	-5	0000	0
-12	CFE7	16	-6	5D46	16
-13	F763	2			
-14	5581	4			
-15	E931	8			
-16	5D46	0			
For U21 change the Setup for Enable to: On Ext. Enable Low U52-6					
U21-3	81CB	18	U21-4	81CB	14
-5	40E5	0	-6	40E5	14
-7	2072	18	-8	2072	14
-9	1039	16	-10	1039	14
-11	FE73	16			
-15	081C	14	-16	081C	18
-17	040E	14	-18	040E	0
-19	0207	2	-20	0207	0
-21	0103	14	-22	0103	0
* unstable data					

9010A HANDS-ON TRAINING

Exercise 4C

FAULT: ADDRESS BUS FS #2
IDENTIFY USING: BUS TEST

1. Set fault switch #2 to its fault position.
2. Perform BUS TEST and record the 9010A display message.
3. What is the meaning of this message?
4. LOOP on this error and toggle fault switch #2 in and out. Observe what happens.
5. Locate on the UUT schematic possible locations for this fault.
6. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How do you know this fault is located on the uP side of the bus buffer?
 - b. What other methods of troubleshooting would you use to find the source of this fault?

Exercise 5B

FAULT:	DATA BUS	FS #3
IDENTIFY USING:	RAM SHORT	
VERIFY USING:	WRITE (STIMULUS)	
	READ (MEASUREMENT)	

1. Set fault switch #3 to its fault position.
2. Perform a BUS TEST. What happens?
3. Now perform a RAM SHORT test and record the display message.
4. What is the meaning of this message? Hint: Use the reference material describing the 9010A error messages.
5. To verify this fault, perform the following WRITES and READS. WRITE a data pattern where one of the tied bits is low and the rest high. Then write that bit high and the rest low.

WRITE @ ENTER = ____

READ @ ENTER = ____

WRITE @ ENTER = ____

READ @ ENTER = ____
6. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. How do you know this fault is located on the buffered side of the data bus?
 - b. Why wouldn't WRITING all zeros or all ones in Step 5 verify this data bus fault?
 - c. What functions do the ENTER key perform in Step 5?

Exercise 8B

FAULT:	I/O CONTROL LINE	FS #10
IDENTIFY USING:	FUNCTIONAL TEST, PROGRAM 22	
ISOLATE USING:	WRITE (STIMULUS)	
	PROBE (MEASUREMENT)	

1. Set fault switch #10 to its fault position.
2. Identify the fault using AUTO TEST and the Functional Test in PROGRAM 22. List this fault below:
3. Should you use a LOOPING READ, or a LOOPING WRITE as a stimulus for isolating this fault? Why?
4. While LOOPING on the function chosen in Step 3 at the appropriate address, check this circuit using the PROBE. On what should the PROBE be synchronized?

Record the nodes you check and the logic levels you observe:

5. Where is the defect and what type did you discover?
6. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. From the way the Functional Test failed, which part of this circuit would you suspect is bad before taking measurements?
 - b. Why didn't the PROBE indicate a tristate level at the open circuit?

Exercise 5C

FAULT:	ADDRESS BUS	FS #4
IDENTIFY USING:	RAM SHORT	
ISOLATE USING:	TOGGL ADDR (STIMULUS)	
	PROBE, SYNC DATA (MEASUREMENT)	

1. Set fault switch #4 to its fault position.
2. Press BUS TEST. Record the 9010A display message.
3. Press RAM SHORT, then ENTER. Record the 9010A display message.
4. What does this message mean?
5. Make sure the PROBE is synchronized to data.
6. Use the Address Toggle Troubleshooting Function for a stimulus. This function will toggle the address bit that you designate.

Press the TOGGL ADDR key.

Enter the failure address.

Enter the suspect bit from the error message.

LOOP on this function as the stimulus.
7. Use the PROBE to check the fault. Record the nodes you check with the logic levels observed.

8. Where is the fault and what type is it?

9. Verify your answer to Step 8 by setting fault switch #4 to its no-fault position and repeating Step 7.

10. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. Why didn't BUS TEST identify this fault?

 - b. Is the TOGGL ADDR function reading or writing data?
What uP lines would you check to find out?

 - c. What other methods of troubleshooting would you use to find the source of this fault?

Exercise 8C

FAULT:	I/O PORT LINE	FS #9
IDENTIFY USING:	FUNCTIONAL TEST, PROGRAM 21	
ISOLATE USING:	WRITE (STIMULUS)	
	PROBE (MEASUREMENT)	

1. Set fault switch #9 to its fault position.
2. Identify the fault using AUTO TEST and the Functional Test in PROGRAM 21.
3. To further identify the fault, perform a READ and WRITE at these Ports. List your results:
4. While using a LOOPING WRITE or a LOOPING READ (choose the correct one) as a stimulus, isolate the fault using the PROBE. Record the nodes you check and the logic levels you observe:
5. Where is the defect and what type did you discover?
6. To review what you have learned from this exercise about troubleshooting with the 9010A, answer the following questions:
 - a. Why did you need to perform immediate mode READS and WRITES in Step 3 to identify the fault?
 - b. Why did you use the particular stimulus you chose in Step 4 for isolating the fault?

MICRO-SYSTEM TROUBLESHOOTER
ADVANCED TRAINING

Outline

- I. Introduction
 - A. Introduce Instructors
 - B. Overview of Course

- II. Review Basic Operations
 - A. Inputting "Known Good" Information
 - B. Built-in Tests
 - C. Troubleshooting Functions
 - D. Probe Operations
 - E. Registers
 - F. Arithmetic Functions
 - G. Programming

- III. Utility Programs
 - A. Probe and the Program
 - B. Identifying unknown devices on your UUT
 - 1. Address and Data Line Identification
 - 2. Mapping Identification
 - C. Display and Print Messages
 - D. Probe Placement Detector

- IV. Actual Circuit Applications
 - A. Working with I/O devices
 - 1. Basic Approach to I/O
 - a. Understand the Device
 - b. How is the Device used
 - c. How to access the device
 - d. How to Stimulate the Device
 - 2. Application Examples
 - a. Programmable I/O
 - b. OFF BUS devices
 - c. Using RUN UUT

- V. Course Review
 - A. Questions and Answers
 - B. Critique

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Laboratory Workbook

The objective of this course is to familiarize the student with the methods of applying the Micro-System Troubleshooter to microprocessor-based systems using practical examples.

During this class you will be using a 9010A with an 8080 pod on an 8080-based UUT (Unit Under Test). The UUT is a designers board made by Nippon Electric Corporation. Using this setup and this lab you will develop a number of programs that will be of use to you on your own systems.

This course is comprised of three (3) major sections. Section I is a review of the basic operations of the 9010A. At the completion of this section everyone in the class will be using the same terminology and have the same understanding about the basic operations of the 9010A. In Section II you will develop a number of utility programs that will simplify the application of the 9010A to your systems at work. Section III will teach a suggested method of deciding how the Micro-System Troubleshooter should be applied to troubleshooting problems. Two circuits on the UUT will be used as actual examples on which this method will be used.

SECTION I - REVIEW BASIC OPERATIONS

Inputting "KNOWN GOOD" information

What are the four methods of inputting "KNOWN GOOD" information into the 9010A memory ?

1. Learn
2. Tape load
3. Manual enter
4. Download (optional)

The following is provided to you as "KNOWN GOOD" information about the 8080A based UUT now attached to the 9010A.

RAM is MEMORY MAPPED from address 8C00 to 8FFF.

ROM is MEMORY MAPPED from address 0 TO 7FF.

I/O is MEMORY MAPPED at address 856C
with ONLY bits 0, 2, & 3 read-writable.

I/O is I/O MAPPED from address 20 to 23
with ONLY bits 3 & 5 read-writable.

Now use one of the four methods listed above to enter the "KNOWN GOOD" information into the 9010A memory.

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Built-in Tests

1. Place a number in the boxes below that represents in which order each of the following tests are performed when the AUTO test is run.

*20x Below
Set new
Ranks.*

2C HAPANE

1 Bus Test

2 RAM Short Test

RAM Long Test

3 ROM Test

4 I/O Test

2. On the lines provided above, place a brief description of what checks are performed for each of the built-in tests.

3. Perform an AUTO TEST on the UUT. Record the results of the test below.

OK

Troubleshooting Functions

The TROUBLESHOOTING functions are probably the most used functions of the Micro-System Troubleshooter. The next questions will review these operations.

1. At the completion of each troubleshooting command the word "OK" will be added to the display. What does the "OK" mean ?

OK
2. What SETUP selections are associated with the "OK" in question 1 ?

Disable error so ok with errors
3. What troubleshooting functions are associated with STS/CTL ?

See Pod

Probe Operation

The probe is one of the most useful tools the troubleshooter has to offer. This section will review the probe operation.

1. List the four functions of the probe available to the operator and check the appropriate box for the Valid Sync modes that will affect each probe function ?

		Probe Function	Valid Sync Modes		
			Free Run(F)	Sync'd (O-E)	No Affect
INPUT	I.	<u>Level History</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <small>new + old only</small>	<input type="checkbox"/>
	II.	<u>COUNT</u>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	III.	<u>SIGNATURE</u>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
OUTPUT	IV.	<u>PULSER</u>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2. What are the valid sync modes, for the probe, when using the Micro-System Troubleshooter with an 8080 POD ?

3. What sync modes are available from the O'Scope SYNC output, when using the Micro-System Troubleshooter with an 8080 POD ?

DATA, ADDRESS
4. When the SYNC mode is FREE RUN, at what speed does the PROBE operate ? _____
1KHz

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Registers

Registers are used extensively in the programming mode for storing and transferring data. The following questions will review the information on registers. It is not so important that you be able to remember all the details about the registers but that you know where to find the information.

1. How many registers are available for data storage in the Troubleshooter ? 0-F (16)
2. How are the registers identified ? 0-F
3. What are DEDICATED registers and how are they identified ?
0 & A-F
4. What are NON-DEDICATED registers and how are they identified ?
1-9
5. What are GLOBAL registers and how are they identified ?
8-F
6. What are LOCAL registers and how are they identified ?
0-7
7. Where was the above information found the quickest ?
REF CARD

Arithmetic Functions

This next section will review the operation of the arithmetic functions of the Troubleshooter.

1. What troubleshooter function(s) allows you to count within a program ? INC Reg Decr
2. The SHIFT LEFT and SHIFT RIGHT functions will cause the binary digits in a value to be shifted one place to the LEFT or RIGHT each time you press the appropriate key. If you took 8FFF and shifted it once to the RIGHT what would the result be in hex ?
47FF
3. If you took the answer to question 2 and shifted it LEFT one time what would the result be ? 97FE
4. What would the result be if you took 8FF8 and ANDed it with F00F ? 8008
5. What would the result be if you took 8FF8 and ORed it with F00F ?
FFFF
6. If a register contained 8FF8, what would the result be if you COMPLimented it? 7007

Programming

In this next section you will review the PROGRAMMING operations of the Troubleshooter.

1. What are the proper key strokes for creating a program ?
PROGM & NUMBER, enter
2. What is the REAL limiting factor on program storage capability and what is its maximum value ? Memory 10 K Storage
3. Once inside a program, in the PROGRAMMING mode, what method(s) are available to move from one step to another ? UP TO 100 Programs
~~GOTO LAST IF~~ Prior - next -
4. What Troubleshooter function(s) are used for UNCONDITIONAL JUMPING ? GOTO
5. What Troubleshooter function(s) are used for CONDITIONAL JUMPING ? IF
6. What Troubleshooter function can you use in a program to print messages on a printer ? Display Aux/IF
7. What Troubleshooter function can you use to write messages to the display ? Display
8. When entering a display message in the programming mode, what key(s) will delete only one character ? _____

Built-in Tests in a Program

As you may know, all the Built-in tests can be used inside a program. In the next part of the lab you will construct a very simple program that will perform an AUTO TEST and then display DONE and beep the beeper.

KEY	DISPLAY
PROGM	PROGRAM _
1	PROGRAM 1_
ENTER	PROGRAM 1 CREATED
AUTO TEST	AUTO TEST
DISPL	DPY-_
D	DPY-D_
>	DPY-DO_
IF	DPY-DON_
E	DPY-DONE_
VIEW ROM	DPY-DONE▲
ENTER	DPY-DONE▲
PROGM	PROG 1 CLOSED - 10182 BYTES LEFT

Now execute program 1

#99
Tape
Merge
32 BIT unsigned address
Pulser Set with
Prog control

SECTION II - UTILITY PROGRAMS

Since the introduction of the Micro-System Troubleshooter a number of generic type programs, hereafter referred to as UTILITY programs, have been developed by customers.

These programs are generic in the sense that they are usable with any of the pods with very little, if any, modifications. Once generated and loaded on tape they can be used on any UUT.

The Probe and the Program - B0163

The PROBE is a very useful device that not only can be used to stimulate, but also to monitor any node on a UUT. When the probe is used to monitor a node, there are three basic steps that must be performed to gather probe data at that node.

1. What are the three basic steps when gathering probe information ?
 1. Sync DATA
Read Probe - clean Reg
 2. STIMULATE
 3. Read Probe
2. How is the probe data transferred into a program ?
Reg 0
3. With the above information, make a program that:
 - A. Gets the probe data.
 - B. Separates the three pieces of probe information into three separate registers.
 - C. And can be used as a subroutine whenever probe information is desired.

House Keep
Don't call
Program.

91 Read Probe

Read Probe
 Reg 8 = Reg 0 AND 7F Event count to Reg 8
 Reg 9 = Reg 0 SHR 5HR 5HR 5HR 5LR 5LR 5LR 5LR AND FFFF
 Reg A = Reg 0 5LR (24 Times) AND 7 (8 Times)

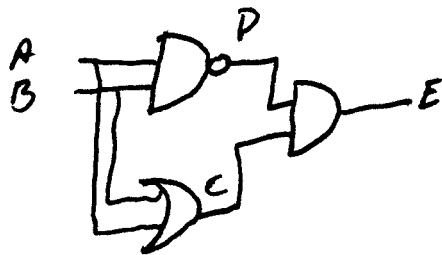
PROGRAM NOTES

EXCLUSIVE OR

Reg C = Reg A OR Reg B

Reg D = Reg A AND Reg B COMP

Reg E = Reg D AND Reg C

A & B in
E out

- 100 Descriptors
- 100 Program Numbers
- 16 Labels
- 10 Tests
- 12 K user memory
- 2 K used for Descriptors & Setup.
- 10 K used by your program
- 32 characters on screen

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

4. To test the program developed in question 3 make another program that will:
 - A. Stimulate the UUT with a RAMP function.
 - B. Separate the probe information using the program in question 3.
 - C. And display the three pieces of probe data.

Identifying Address and Data lines

There are times when nothing is known about a UUT: no schematic, no program listing, nothing more than the UUT itself. True, the LEARN feature can find the BUS connected devices, but how does that relate to each IC on the UUT. i.e. what uP signal goes to the pins on the device.

Using just the Troubleshooter and the probe, you can find out a lot about your UUT. One thing you might find useful would be which pins on a device are tied to an address line and which are tied to data lines.

Your next problem is to design a program that, after the probe is placed on an unknown point:

- A. Will tell if the probe is setting on an address or data line.
- B. If it is, then identify which line.
- C. If not, then display "CAN'T IDENTIFY".

To help in developing the program answer the following questions:

1. What STIMULUS from the Troubleshooter should you use to identify the address and data lines ? Toggle Address & Data
2. What piece of probe data should be used to identify the address and data line when using the stimulus from question 1 ?
Level
3. Using the utility program ^{at} for the probe, what register will contain the probe data for question 2 ? A
4. What value will you expect to find in register A (Ans. Q3) when the probe is on an address or data line ? 5

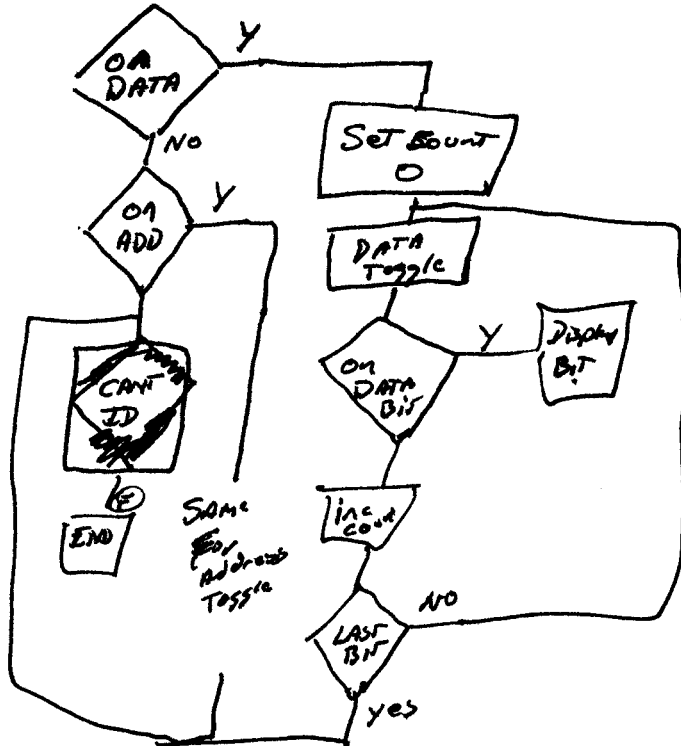
Use the space provided to write your program.

Program # 92

PROGRAM NOTES

ID DATA & Address Lines

START



used Program 91 Reg A on Read probe

Read Probe
 Sync DATA
 WALK @ 0000 = 01
 EXEC 91
 IF Reg A = 5 goto 1
 Write 0000 < 00
 Write FFFF < 00
 EXEC 91
 IF Reg A = 5 goto 4

TABLE 7
 DSP. Cant Find
 GOTO F

TABLE 1
 Reg D = 0

TABLE 2
 DATA Toggle 0 = 00 BIT Reg D
 EXEC 91
 IF A = 5 GOTO 3
 inc Reg D
 IF 8 > 7 Reg GOTO 2
 GOTO 7

TABLE 4
 Reg D = 0

TABLE 5
 ATO 6 0 BIT 0 Reg D
 EXEC 91
 IF Reg A = 5 goto 6

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Mapping Devices on your UUT

Another useful thing to know about an unknown UUT is what conditions must be met to cause a device to be selected.

Design a program which after placing the probe on the chip select or chip enable pin, will:

- A. Use the probe utility program.
- B. Stimulate the UUT in such a way as to identify the conditions which cause the chip to be selected.
- C. If found, display the condition.
- D. If not found, display "CAN'T IDENTIFY".

Again, some basic decisions have to be made about the method this program will use. The following questions will help in these decisions.

1. What usually determines when a device is selected ?

2. What form will the STIMULUS take to identify the chip select ?
WRITE

3. Which piece of probe data will help identify chip select ?
LVL

4. What SYNC MODE will be used for this program ? ADD

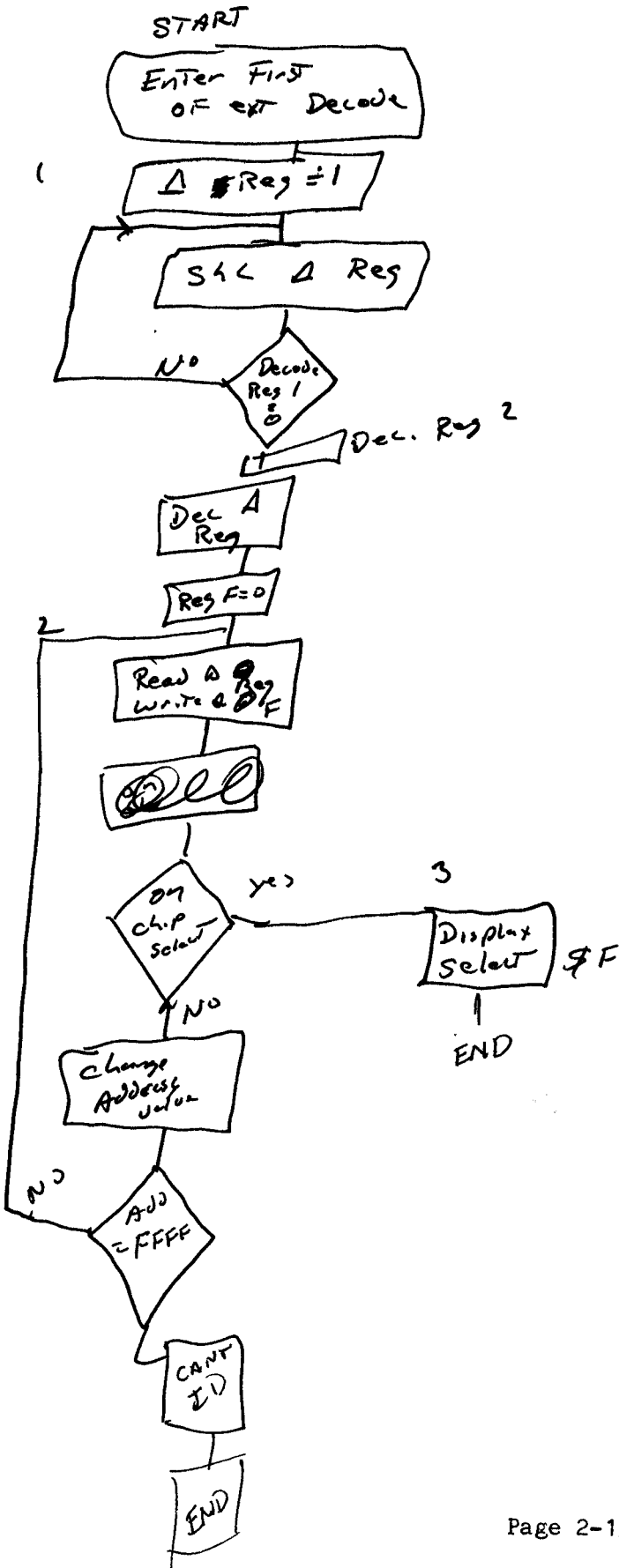
5. Using the probe utility program, which register will be used to identify chip select (ans. Q3) ? A

6. What value will you find in register 5 (Ans. Q4) when the probe senses a LOW ? 4

Using the space below design the chip select program.

Program 93

PROGRAM NOTES Find chip select



Display and Print Messages

When designing a Guided Fault Isolation (GFI) program you will probably use a lot of display, or printer messages. These messages are used to guide the operator through the circuitry of the UUT, telling him what needs to be done next. You will find that a lot of these messages are repetitive and time consuming for the programmer. In addition these messages consume a lot of memory space, especially when they are repeated throughout a GFI program.

For example take the message:

PLACE PROBE ON PIN 5 OF U30

True, in order to save memory space such a message could be abbreviated to:

PROBE U30 PIN 5

Or simply

PROBE U30-5

You can however, save a lot more space by making a subroutine that contains the repetitive parts of the message and allows any variables to be changed by the calling program.

For your next problem, design a program that will:

- A. Use a single register to receive all the message variables from the calling program.
- B. Separate each variable from the register.
- C. Display the variables in their proper place in the message.
- D. Returns to the calling program.

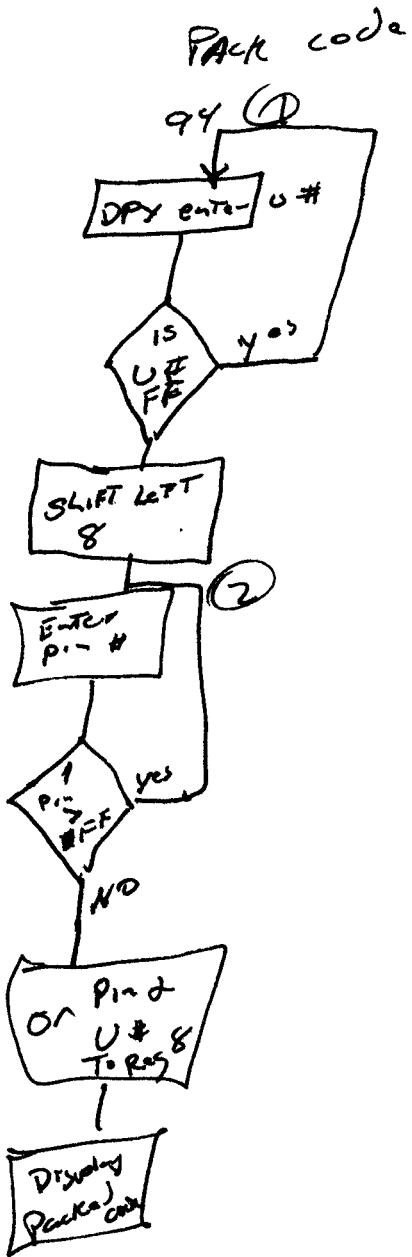
First answer the following:

1. What are the variables in the message PROBE U30 PIN 5 ?

30 5
2. What are the maximum values allowed for each variable ?

3. For each variable, how many register bits will be required ?

PROGRAM NOTES



ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Probe Placement Detector

When using display messages in the Micro-System Troubleshooter to Guide an operator through the UUT, it will be necessary to wait for the operator to place the probe properly on the node. If the program did not stop, or wait, the program would fly right by the message and take a measurement before the operator could even pick the probe up off of the table.

One solution would be to follow each message with a STOP command. When the operator has the probe placed on the node, he will press the CONTINUE key to go on. This is a perfectly viable solution to the problem, but there is a way of completely automating this whole procedure.

Your next problem is to design a program that:

- A. When called will determine when the probe is placed on a node.
- B. Will return to the calling program indicating that the probe is placed properly or is on an open circuit.

Some specific questions have to be answered before writing this program.

1. What function of the probe will be used to detect that it is placed on a node ? Level History
2. Using the function chosen in question 1, what will indicate that the probe is sitting on a node ? _____
3. How will the program detect an open node (high impedance)?
TRISTATE
4. What will be the SYNC mode for this program ? Free run
5. How will the program sense that the probe is firmly in place on a node ? _____
6. How will this program indicate to the calling program that the probe is on an active node, or on an open circuit?

Program #96

PROGRAM NOTES

Probe Placement Detector

B FLAG 1 = on NODE
0 = NOT on NODE

START

Sync Free Run

Set count = 20

ON VALID NODE?

NO YES

Dec count

count < 0?

NO YES

SET FLAG = 0

END

SET count = 0

STILL ON NODE?

YES NO

Dec count

count = 0?

NO YES

SET FLAG = 1

END

SECTION III - ACTUAL CIRCUIT APPLICATIONS

Working With I/O Devices

When developing tests for devices like Peripheral Interface Adapters (PIA), Dual Asynchronous Receiver/Transmitter (DART), Universal Asynchronous Receiver/Transmitter (UART), Programmable Interrupt Controllers (PIC), Serial Input/Output (SIO), etc. there are some basic steps that you should always follow.

1. Understand the internal operations of the device and what each input and output signal does.
2. Determine how the device is used for the UUT application.
3. Identify how the device is accessed by the microprocessor on the UUT.
4. Identify how to stimulate the device and how to evaluate the results.

Once you have completed the four steps above and manipulated the device using the immediate mode functions, its a very easy process to put the procedure into a program that guides an operator through the test.

Using the Troubleshooter with different I/O devices requires a little ingenuity on the part of the operator, but in most cases the Troubleshooter can be of significant help in troubleshooting these devices. The remainder of this LAB deals with applying the four step technique above to actual circuits on the UUT. Once you understand the technique, you can easily apply it to other I/O devices that you may run into.

In review, remember that the built-in I/O test only works on those devices that have read/writable bits, like RAM. The I/O test is also bit selective for those addresses in which not all bits are used.

It is obvious that there are many types of I/O devices that do not fall into this category. No attempt has been made to make a test that covers all types of I/O devices. Such a task would be monstrous to say the least. This is where the programming capability of the Troubleshooter comes in.

You can customize the basic functions of the Micro-System Troubleshooter into a programmed sequence that tests all portions of a particular I/O device.

Example #1

For this example you will devise some tests for the programmable I/O that passes keyboard data on the UUT. Looking at the schematic you will see that this I/O device, U30, is an 8255.

Step 1 - Understand the internal operations of the device and what each input and output signal does.

The first step in the technique is to learn the operations of the I/O device and how it performs its function. The number one source for this information is usually the data book on the device. Sometimes there are differences between chips of the same type but of different manufacture. Be sure to get the data book that is published by the same company that manufactures the device itself.

In the reference data section of this manual you will find data book information on the 8255. Read this material and answer the questions below.

1. How many ports are available through the 8255 ? 3
2. What are the different modes of operation for the 8255 ?
0 + 1 + 2
3. How are the various modes set on the 8255 ?
CONTROL WORD
4. What does bit 7 in the control word do ?
MODE ACTIVE SET FLAG
5. What function does pin 35 serve ? RESET - MODE 0 - INPUT
6. What lines are used to transfer data between the microprocessor and the PIA? D0 - D7
7. Which pin(s) and what logic level does it take to select the PIA?
CS
8. Which pin(s) and what logic level(s) will select the port(s) of the 8255 ? _____
9. Are there any other signal conditions that must be met in order to operate this device ? _____ If so, what are they ?

100FB

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Step 2 - Determine how the device is used for the UUT application.

To test U30 properly, you must create the same configuration that is used during normal UUT operations. This next section will help you identify how the 8255 is configured for the keyboard application on the UUT.

Using the reference data section on the 8255 and the UUT schematic, answer the following questions:

1. What ports are used for the keyboard application ? A & C
2. Looking at the circuit design, what gives you the clue that a port is used as input? Pullup - PORT A
3. Which direction are these ports configured?
0
4. What gives the best clue as to what mode is used on the 8255 ?

5. What mode is U30 programmed for in the UUT application ? 0

Step 3 - Identify how the device is accessed by the microprocessor on the UUT.

Using the information gleaned from the data book, as well as other information and tools at your disposal, you can determine what it takes to access the device.

You know from reading the data book information what it takes to select this device (see Q.7 of step 1). In 95% of the cases the address is what determines when this device is selected.

1. What microprocessor line(s) are tied to the port select pin(s) ?
A0-1
2. What address must the microprocessor put on the address bus to select this chip ? F8 - FB
3. What microprocessor signals are needed to satisfy the requirements of Step #1 question 9? _____

Step 4 - Identify how to stimulate the device and how to evaluate the results.

At this point it's time to start considering how the Micro-System Troubleshooter is going to fit into the test procedure. Knowing that the 8255 is programmable, you will have to first initialize and configure it. Not wanting to reprogram the device you would best be advised to clear it.

1. What is the easiest way to clear any previous programming of the PIA and how can it be done on the UUT ?

Press Reset

Now perform the steps that you have outlined in questions 1.

With the device cleared, it is now time to start programming the PIA for test.

2. What PIA control word, in hex, will set the proper mode and port direction(s) for testing the UUT keyboard (See Step 2, Q3 & Q5) ?

It becomes obvious that with the 9010A plugged into the UUT's microprocessor socket it will have to configure the 8255.

3. What signal conditions have to be met on the 8255 before the control word will configure the PIA ?

Write 100FB = 93

4. What is the complete 9010A command that will satisfy the conditions of question 3 ?

Using the immediate mode of the 9010A, configure the PIA.

5. Now that the PIA has been configured for the proper mode and port direction, what operations are needed before we can sense key 0 through the pod?

Write 100FA = DEF = PCY low
enable col # 0-7

6. What 9010A command(s) will accomplish question 5 ?

Write

7. What is the expected data word that will come back through the pod if none of the keys are pressed ?

FF

8. What is the expected data word that will come back through the pod if the 0 key is pressed ?

FE

9. What 9010A command(s) will accomplish questions 7 & 8 ?

Read ~~100FA~~ 100F5

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

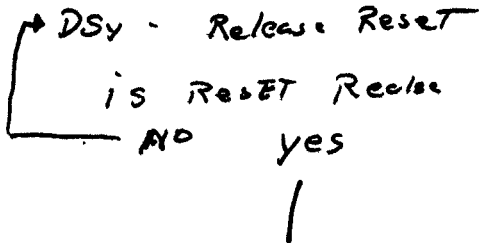
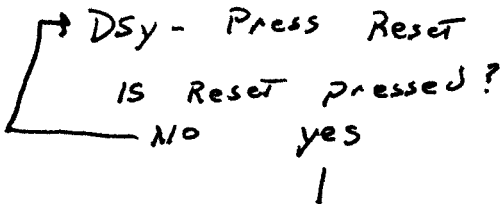
Now that you have tested your approach using the 9010A in the immediate mode, design a program that will:

- A. Instruct the operator to reset the PIA and have the 9010A sense when it has been done.
- B. Configure the PIA for testing keys 0-7 on the keyboard.
- C. Instruct the operator which key to press.
- D. Test the keyboard to verify that the proper key was pressed. If correct, test the next key. If incorrect, report the error to the operator.

Program #3

PROGRAM NOTES

Keyboard Test 0-7



CONFIGURE THE PIA

Reg 1

INIT Key count = 0

Reg 2

INIT KeyWord = FE

DSy Press Key Reg 1

ANY Key Pressed? NO

yes

Correct Key

yes

NO ——— ERROR

INC Keycount

change Key code

LAST Key?

yes

NO

Release Key?

NO

DPy - TEST Passed

END

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Before claiming that the device is good, you would test the remaining two columns of keys . For the purpose of this class however, we will assume that you would be able to complete the program on your own.

The previous program would be a COMPLETE THROUGH TEST of the keyboard circuit. This is very nice but as in real world situations, things do fail from time to time. Considering failure to be realistic it is now necessary to decide what to do if the test fails.

The usual procedure is to "DIVIDE AND CONQUER". If you divide this problem in half you would probably do so at the keyboard itself. This means you have port C as output to the keyboard and port A as input from the keyboard. To decide which way to go first is really a toss up. For this example you will test the input side first.

When testing the PIA for input you can use the probe in its pulser mode, and by probing each port pin you can sense the changes back through the pod.

To demonstrate, perform a read at port A and then loop on it.

1. Before you can probe the port, however, what logic level should be selected on the probe to stimulate port A? _____
2. Is it necessary to synchronize the probe to provide the proper stimulus? NO DATA

Now probe each input pin of port A and note the Troubleshooter display. In the table below list the data read as each input line of port A is pulsed by the probe.

PA0	<u>FE</u>
PA1	<u>FD</u>
PA2	<u>FB</u>
PA3	<u>F7</u>
PA4	<u>EF</u>
PA5	<u>DF</u>
PA6	<u>BF</u>
PA7	<u>7E</u>

Now that you have checked the 8255 for input, you need a check for port C, which is programmed for output. There a number of ways to check for output using the probe. Each of the three functions of the probe will test the port for output very nicely. However, for this next example, you will use the probes signature function on port C.

To test using signatures, you have to know what the signature is for a good data line. Assuming that the 8255 passes input data directly to the output the signatures should be the same.

MAKE SURE PULSER IS OFF

3. Using the RAMP function for a stimulus, what is the proper signatures for the following data lines:

	<u>INPUT</u>	<i>NOT SAME on chips</i>	<u>OUTPUT</u>
D ₀	<u>96EC</u>	<i>Intel</i>	PC0 _____
D ₁	<u>725B</u>		PC1 _____
D ₂	<u>E5ED</u>		PC2 _____
D ₃	<u>5BEO</u>		PC3 _____
D ₄	<u>EE8E</u>		PC4 _____
D ₅	<u>85EA</u>		PC5 _____
D ₆	<u>77C7</u>		PC6 _____
D ₇	<u>6EBE</u>		PC7 _____

This same technique can be use for all output ports.

4. If you use the same stimulus should the signatures be different for each output port ? (For example is PA0 different than PBO or PC0) NO

Another way of checking the ports would be to use a jumper that feeds one port into another port. Make one port output the other input. Then have the Troubleshooter write data to the appropriate port and read the same data from the other.

If the lines from the PIA go to an edge connector or jack then a special jumper plug could be constructed for testing. In the case of our UUT we have the I/O lines go through a buffer and off of the board through a plug. The buffers, however, are not installed so we can not use a special plug.

Another way of getting the ports tied back to back is through a 40 pin clothes pin adapter. You are provided one of these adapters already wired in the accessory kit next to your 9010A. Using the clothes pin adapter provided, slip the adapter over the 8255 with the TALL pins on the RIGHT. This ties ports A and B together.

5. What is the control word, in HEX, that programs port A for mode 0 output, port B for mode 0 input and port C as input ? _____

Configure the PIA using the control word in question 5 and then send data out port A and read it back through port B.

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Up to this point you have been working with the 8255 in mode 0. This next section will demonstrate some techniques to use for mode 1.

6. What is the significant difference between Mode 0 and Mode 1 operation ? _____
7. What control word will make port A input and port B output in Mode 1 ? _____
8. Can port C be used for I/O, while ports A and B are in mode 1 ?
_____ If so how ? _____
9. What are the pin numbers and labels for the handshake lines for both the input and output ports ? _____
10. How can you use the Troubleshooter to pass data in and out of these ports when handshaking is necessary ?

Using the clothes pin adapter, test the 8255 in mode 1 by configuring the device as port A input and port B output. Write data to port B, pass the data to port A by using the answer from question 10 then read port A.

In the space provided list the necessary immediate mode commands of the Troubleshooter that will test the 8255 in mode 1. Be sure to Reset the UUT before testing begins.

Example #2

This last example of working with I/O devices is a bit different than the first. There may be occasions when it would be desirable to test circuitry beyond the I/O device. When testing components off the bus, the difficulty lies with how to get the stimulus to the device. The trick is "how does the microprocessor get the information there?" The problem becomes twofold because you not only have to understand the I/O device that must be "written" through but also determine the best means of stimulating the device under test.

Now develop a test that will completely test the operation of U52B in the Tape Interface section of the UUT.

Step 1 - Understand the internal operations of the device and what each input and output signal does.

1. Looking at the Function Table provided in the Reference Section for the 74LS74, list the conditions that must be met to make the Q output high?

D H
 CK T
 PRE H
 CLR H

2. Are any of these inputs set by the design of the UUT? _____ If so, which lines? _____
3. Is it possible to change the configuration of U52B through program control? yes _____

Step 2 - Determine how the device is used for the UUT application.

1. What is the application of U52B?

Step 3 - Identify how the device is accessed by the microprocessor on the UUT.

With the information that we have obtained up to this point we can now make the determination as to how the microprocessor accesses U52B.

1. How does the microprocessor gain access to the flip-flop?

2. Determine the port direction and mode that U30 must be configured to test U52B. _____
3. What is the control word, in hex, that will configure U30 properly? _____

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Step 4 - Identify how to stimulate the device and how to evaluate the results.

1. What 9010A stimulus could be used to satisfy the D input requirements? _____
2. What 9010A stimulus could be used to satisfy the clock input?

3. Once the stimulus has been determined for the clock, what format should that statement take to leave the clock LOW when the stimulus is complete? _____
4. What piece of probe data could be used to determine if U52B is operating properly? _____
5. What is the complete 9010A command that will configure U30 for this test ? _____
6. What addresse(s) will be needed to test U52B? _____

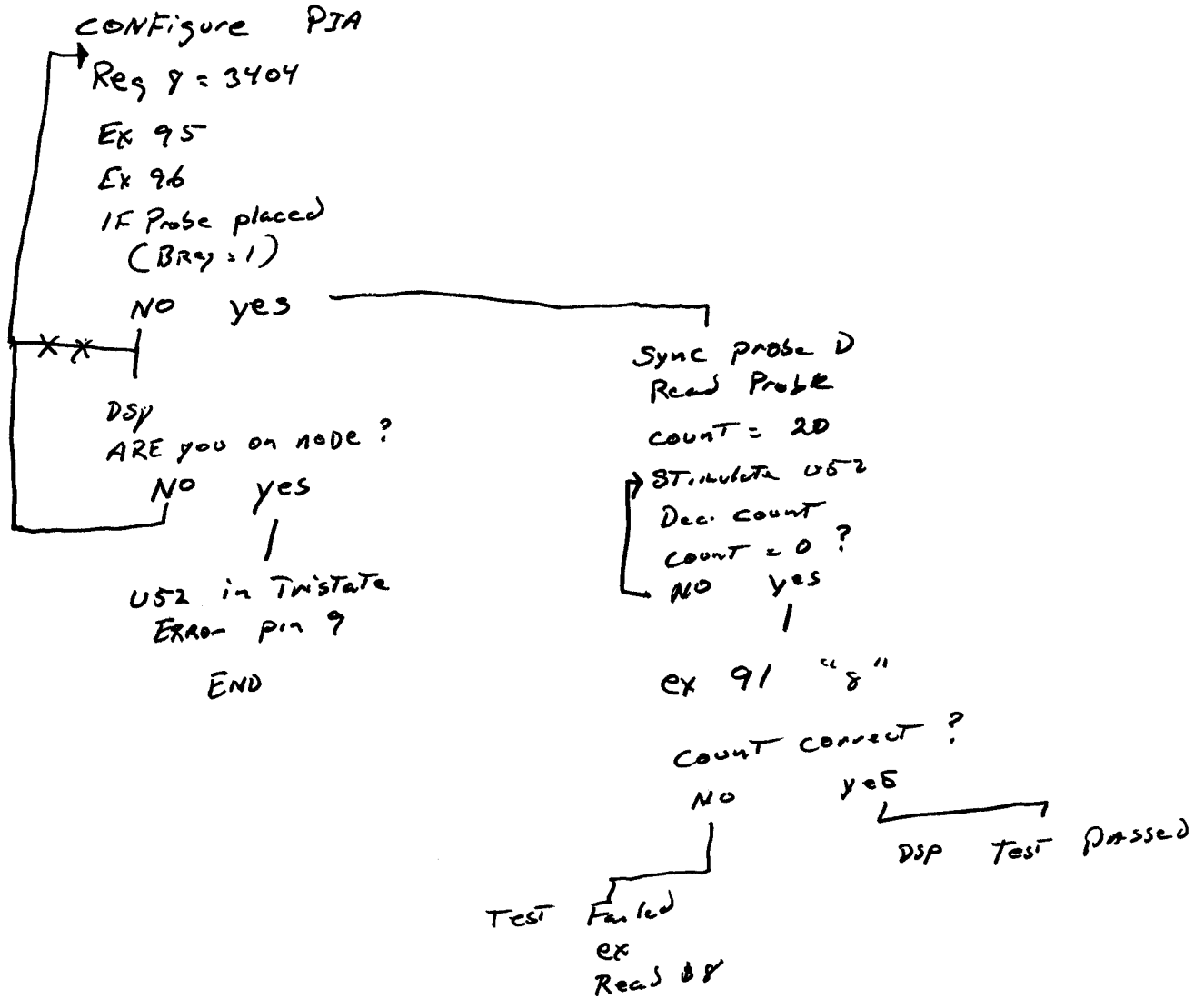
Now with all of the information gathered, design a program that will test U52B by:

- A. Configure U30 properly.
- B. Have the operator place the probe on the Q output using the following utility programs:
 - 95 1. display
 - 96 2. probe placement detector
- C. Have the program stimulate U52B so that a predetermined count will be collected by the probe.
- D. If the count is incorrect, have the 9010A display the results.
- E. If the count is correct, then display OK.

Program #4

PROGRAM NOTES

TEST FF



ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

USING RUN UUT

Not all devices can be tested using the functions of the Micro-System Troubleshooter. If absolute timing in a test is necessary, a program written in microprocessor code can be written and downloaded into UUT RAM and then exercised with RUN UUT.

Listed below is an example of an assembled 8080A/8085A microprocessor program for the TK-80 board.

8C00	21 40 8C	LXI H,8C40H	8C1B	77	MOV M,A
8C03	36 50	MVI M,50	8C1C	01 FF 0F	LXI B,0FFFH
8C05	31 00 8D	LXI SP,8D00H	8C1F	0B	DCX B
8C08	3E 01	MVI A,01	8C20	79	MOV A,C
8COA	21 F8 8F	LXI H,8FF8H	8C21	E0	ORA B
8COD	77	MOV M,A	8C22	C2 1F 8C	JNZ 8C1FH
8COE	2C	INCR L	8C25	F1	POP PSW
8COF	C2 0D 8C	JNZ 8CODH	8C26	07	RLC
8C12	F5	PUSH PSW	8C27	C3 0A 8C	JMP 8COA
8C13	21 40 8C	LXI H,8C40H	8C2A	3E 01	MVI A,01H
8C16	7E	MOV A,M	8C2C	D3 FA	OUT FAH
8C17	3D	DCR A	8C2E	76	HLT
8C18	CA 2A 8C	JZ 8C2AH			

Using the above information a 9010A program can be written that will load this program into the UUT's RAM space starting at 8C00.

WRITE @ 8C00 = 21	WRITE @ REGF INC = 2A
WRITE @ REGF INC = 40	WRITE @ REGF INC = 8C
WRITE @ REGF INC = 8C	WRITE @ REGF INC = 77
WRITE @ REGF INC = 36	WRITE @ REGF INC = 1
WRITE @ REGF INC = 50	WRITE @ REGF INC = FF
WRITE @ REGF INC = 31	WRITE @ REGF INC = F
WRITE @ REGF INC = 0	WRITE @ REGF INC = B
WRITE @ REGF INC = 8D	WRITE @ REGF INC = 79
WRITE @ REGF INC = 3E	WRITE @ REGF INC = B0
WRITE @ REGF INC = 1	WRITE @ REGF INC = C2
WRITE @ REGF INC = 21	WRITE @ REGF INC = 1F
WRITE @ REGF INC = F8	WRITE @ REGF INC = 8C
WRITE @ REGF INC = 8F	WRITE @ REGF INC = F1
WRITE @ REGF INC = 77	WRITE @ REGF INC = 7
WRITE @ REGF INC = 2C	WRITE @ REGF INC = C3
WRITE @ REGF INC = C2	WRITE @ REGF INC = A
WRITE @ REGF INC = D	WRITE @ REGF INC = 8C
WRITE @ REGF INC = 8C	WRITE @ REGF INC = 3E
WRITE @ REGF INC = F5	WRITE @ REGF INC = 1
WRITE @ REGF INC = 21	WRITE @ REGF INC = D3
WRITE @ REGF INC = 40	WRITE @ REGF INC = FA
WRITE @ REGF INC = 8C	WRITE @ REGF INC = 76
WRITE @ REGF INC = 7E	
WRITE @ REGF INC = 3D	
WRITE @ REGF INC = CA	

After this program is loaded into RAM, you start the program by pressing RUN UUT @ 8C00 and, in this case, the TK-80 LED segments will light in succession. This program was designed for illustration only and is of no practical value.

When you do run into a test that needs to be done at "real real" speed then you will need a way of determining when a microprocessor test program has finished. There are a number of ways to do this.

Once the 9010A is told to RUN UUT the POD cuts off communications with the Troubleshooter. As long as a program that is running in the Troubleshooter does not require action by the POD it will stay in RUN UUT. So, one method of terminating RUN UUT after a microprocessor has finished running, is to have the Troubleshooter go through a time delay program that runs just a little bit longer than the microprocessor program runs.

Another method is to find a part of the UUT circuit that is not used by the microprocessor test program, e.g. an unused chip select line or an output port line. You will design the microprocessor program to put a specific level, high or low, on this unused line when the program has finished running. The program listed above does this very thing. Of the last five instructions 3E, 1, D3 and FA cause the microprocessor to send a 1 out the port labeled FA. This raises bit 0 to a high when the program is done. By having the Probe synchronized to FREE RUN and continually monitoring bit 0 for a high it will cause the program to jump out when a high is sensed.

If a microprocessor test program has results that need to be tested for the proper values, then the test program will have to load the results of the test into another RAM location which, when the Troubleshooter determines that the RUN UUT program has finished, the Troubleshooter will read and make decisions based on that data.