9010A

Micro-System Troubleshooter

Evaluation Manual

K0033A-01U8107/SE EN
Rev. A 7/81

©1981, John Fluke Mfg. Co., Inc., all rights reserved Litho in U.S.A.

FLUKE®

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

WHAT THIS MANUAL IS ABOUT

Philosophy of the 9010A

The 9010A Micro-system Troubleshooter was designed to fill a sizeable gap in the availability of instruments for troubleshooting microprocessor based products (micro-systems). It gives you a window through which to control and view the activities of a micro-system, without the complicated and time-consuming interconnecting, programming, and analyzing required by most testers. If you have the 9010A for evaluation, however, you will want to verify this for yourself.

Purpose of the Manual

The purpose of this manual is to take you step-by-step through the process of learning how to operate and program the 9010A in a real-life situation. The manual contains instructions for using the 9010A to test and troubleshoot the 8080-based single-board computer included with this kit. It also describes how to run the programs on the demonstration cassette tape we have included. Finally, it presents considerations on how to get started testing and troubleshooting your own microprocessorbased products.

This manual assumes you know little about the 9010A. Additionally, the evaluation kit contains the 9010A Operator, Programming, and Pod manuals. Feel free to refer to them as needed to enhance your understanding of the more technical aspects of the 9010A.

How to Use the Manual

The manual contains lab-type instructions. The idea is to read about a capability or feature, then prove to yourself that it works on the kit's single-board computer, the TK-80A. Be sure to progress in sequence through the exercises, as they are presented in the order most likely needed for easy understanding.

The lab exercises ask questions to highlight salient material. Please answer them on a piece of scratch paper rather than in this manual.

Our answers are at the bottom of the page on which the question is asked. Each question is followed by a superscripted number. The answers have corresponding numbers, like footnotes. For example, look at the bottom of the page for the answer to this question: What should you do if your answer disagrees with ours?[1]

The TK-80A's component locator and schematic diagram, along with the demo tape program listing, are in the appendices.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

1. Should your answer disagree with ours, check to be sure you performed the exercise as written. If it still disagrees, then there might be a fault in the 9010A or TK-80A board. Note the problem symptoms on a sheet of paper and notify your Fluke representative. Your rep will clear up the problem as quickly as possible.

Specific Learning Objectives

When you have finished reading and performing the exercises in the manual, you should be able use the 9010A to:

1. Connect the 9010A to the Unit Under Test (UUT) properly;
2. Locate ROM, RAM, and I/O on the UUT;
3. Conduct built-in tests of the UUT ROM, RAM, and processor bus;
4. Use the built-in troubleshooting function to test and isolate failures;
5. Use the probe to stimulate circuits, take signatures, and count events;
6. Write and run a 9010A program to test a portion of the UUT;
7. Isolate faults on the UUT with programs from the demonstration tape;
8. Plan an approach to testing your own UUT with the 9010A.

HOW TO GET SET UP

Items Needed

Before configuring the system, check your kit to ensure it contains the items in Table 1. If any are missing, contact your Fluke representative.

Table 1. Evaluation Kit Contents

9010A Micro-system Troubleshooter
9000A-8080 Pod
9000A Probe
9010A Operators Manual
9010A Programming Manual
8080A Pod Manual
9010A Demonstration Cassette Tape
TK-80A Single Board Computer
TK-80A Power Cord
2 Micro-clip Jumpers
Spare .5-Amp and 2-Amp Fuses
This Evaluation Manual

CAUTION:

The TK-80A board contains static-sensitive components. Be careful in its handling so that it is kept in its conductive pouch until ready for use. Do not subject it to static discharge or it will be damaged.

Always perform configuration and power switching in accordance with the following rules:

1. Always ensure 9010A power is off BEFORE connecting/disconnecting the pod to/from the 9010A;

2. Always ensure the pod is connected to the 9010A and 9010A power is on BEFORE connecting/disconnecting the pod to/from the UUT;

3. Always ensure 9010A power is OFF and the pod is connected to the UUT BEFORE applying/removing power to/from the UUT.

Configuration and Power-Up Sequence

Figure 1 shows how the system looks when properly configured. Connect the system components together and apply power to them in the sequence which follows. Notice during the sequence that the 9010A performs not only a power-up self-test, but also a pod self-test. These tests offer reasonable confidence in the integrity of the 9010A system before you begin testing and troubleshooting.

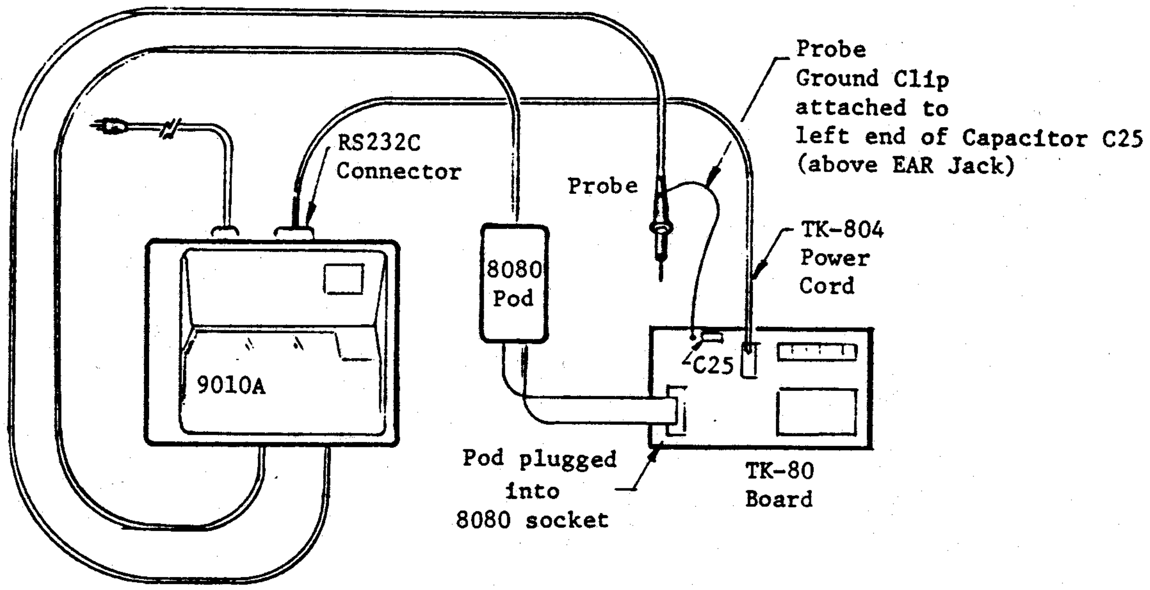


Figure 1. Evaluation System Configuration Diagram

1. With 9010A power off, plug the pod and probe into the front (underside) of the 9010A. Slide the connector locks closed.

2. The 9010A should already be configured to accept AC power which is normal for your area. If you desire to change it or think it may be configured incorrectly, configure it properly in accordance with the instructions in section 2 of the 9010A Operator's Manual. Plug one end of the 9010A power cord into the 9010A and the other end into a standard AC power receptacle.

3. Switch power on to the 9010A (left front corner).

4. Verify that the 9010A displays POWER UP OK. If it does not, then contact your Fluke representative.

5. Plug the pod flat cable into the self-test socket on the pod and clamp it in place with the knurled knob beside the socket.

6. Press the BUS key on the left side of the 9010A keyboard.

7. Verify that the 9010A displays POD SELF-TEST 8080 OK. If it does not, then ensure the pod connector is properly installed on the 9010A, and that the flat cable plug is firmly seated in the self-test socket (use the knurled knob beside the socket) and repeat step 6. If it still does not display properly, contact your Fluke representative.

8. Remove the pod flat cable from the self-test socket and plug into the TK-80A board's 8080 microprocessor socket. Ensure that the diagonal corner of the flat cable plug is at the lower right corner of the socket. The 9010A should display UUT POWER FAIL because UUT power is off.

9. Attach the probe ground clip to the LEFT end of one of the 15 uf capacitor U25 at the extreme top edge of the board (above the EAR jack). This point is chassis ground. The other side of the capacitor is +.5 Volts.

10. Switch OFF 9010A power and connect the TK-80A power cable to the connector provided on the TK-80A board at the left end of the display. Make sure it is fully seated. Connect the other end of the cable to the connector on the rear of the 9010A.

11. Switch ON 9010A power and verify that the 9010A displays POWER UP OK and the TK-80A display lights up. If the displays are not correct, then ensure that the power cord to the TK-80A is firmly connected and fully seated on both ends. If the displays are still not correct, then check the fuses inside the box in line with the TK-80A power cord. If a fuse is blown, replace it with the correct one from the spares. If there is still no display, then contact your Fluke representative.

ABOUT THE TK-80A SINGLE-BOARD COMPUTER AS A UUT

Introduction

You will be using the TK-80A Single Board Computer to do most of the exercises in this manual. We selected the board because it typifies a microprocessor-based system, and because it is based on the widely-used 8080A microprocessor. However, since all computer boards have their unique differences, it is appropriate briefly to study the TK-80A.

Figure 2 is the TK-80A block diagram. Appendix B at the rear of the manual contains the schematic diagram and component locator. Refer to them as necessary in the following discussion.

The board contains five major circuits:

1. Microprocessor, Timing, and Bus System Control
2. Memory and Read/Write Control
3. 8-Digit, 7-Segment Display and Control
4. Programmable Interface Adapter (PIA) and Keyboard
5. Tape Input/Output

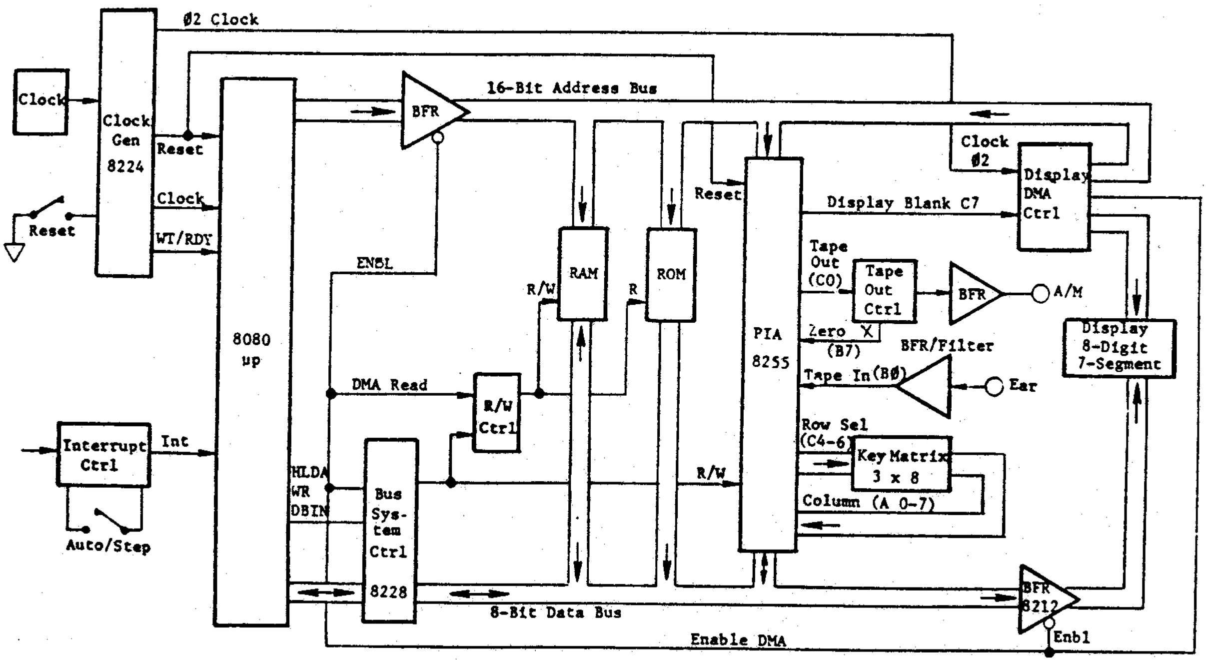


Figure 2. TK-80A Single-Board Computer Block Diagram

Microprocessor, Timing, and Bus System Control

Notice from the block diagram that the clock generator, bus system controller, display buffer, and PIA circuits each consist of a single integrated circuit. These are described fully in the manufacturer's data book. We will give only cursory attention to them here.

The 8080A pod actually contains an 8080A microprocessor. For this reason, it can operate the UUT at normal UUT timing rates. The 8080A has eight bi-directional data lines for transferring data between itself and external circuits. It has 16 address lines for selecting memory and input/output devices as the source or destination of data being transferred.

In using the 9010A to test and troubleshoot the UUT, we are normally unconcerned about the unique microprocessor instruction set because we are testing circuitry external to the processor. We are mostly concerned about its interface to external devices, not how it operates internally.

The 8080A synchronizes data transfers by pulsing the SYNC line when the address and data are valid. However, it puts data on its data lines at other times to notify the bus system controller chip of the type of operation it is about to perform: read, write, memory, or I/O transfer.

The clock generator circuit divides the system crystal frequency to a usable value (2 MHz) and distributes its phases to the processor and display control circuit. It also interfaces the reset pushbutton to the processor. Although RESET normally causes the processor restart at address zero, you can set up the 9010A to ignore it.

The bus system control chip, when commanded by the processor, issues commands to read or write data between the processor and either memory or I/O (the PIA). It also buffers the data lines between the processor and the rest of the board. It uses the HLDA, WR, and DBIN (refer to the 8080 Pod Manual) signals to control direction of data flow on the data bus.

The TK-80A has an interrupt control circuit. However, it is set up to let interrupts be generated by an external circuit and is not part of this discussion.

Memory and Read/Write Control

The TK-80A board contains 1 kilobyte (1024 bytes) of random access memory (RAM) and 2 kilobytes of read-only memory (ROM). ROM addresses are 0-7FF, while RAM addresses are 8C00-8FFF. The ROM contains a complete operating system program which will run when the board's own 8080A is installed. It can also be made to run by the 9010A. The RAM is used for temporary storage of programs or data. The Read/Write control circuit selects which memory chip the processor is accessing at any one time.

The board uses a 16-bit address bus and an 8-bit data bus for transfer of data and addresses. The devices which send data to the busses have tri-state outputs: they may output high (5V), or low (0V) logic levels, or enter a high-impedance state which prevents them from loading any other circuit currently driving the bus.

Display, Control, and Direct Memory Access

The TK-80A display contains eight 7-segment digits. The board uses a direct memory access (DMA) technique to update the display and keep the displayed data illuminated. During the time the processor is not busy transferring data to other chips, the display control circuit performs memory read operations from addresses 8FF8 through 8FFF. It uses the tape circuit oscillator to circulate a counter for generating the addresses. The counter is also decoded to select which display digit is being lit.

At intervals between processor data transfers, the display control puts the address on the address bus and signals the bus system controller to enable a memory read operation. After the read, the data is gated through the 8212 Buffer to the selected display digit. Each data bit which is a 1 lights a particular segment (see lower right corner of the schematic).

The 9010A can command the Programmable Interface Adapter (I/O control chip) to send a display blank signal to the display control circuit. This will make the characters in the eight display memory addresses invisible. You can also cause the 9010A to write data to those addresses, thus displaying whatever you want.

Programmable Interface Adapter (PIA) and Keyboard

The PIA is an input/output controller. The TK-80A board uses it to enable and read the keyboard and to control a circuit which transfers data with an external tape recorder.

The PIA contains four registers: one control register and three data bi-directional registers. The 9010A treats the PIA as a memory device with addresses in the range 100F8 through 100FB. Address 100FB is the write-only control register, while 100F8-100FA are I/O registers A, B, and C, respectively. The block diagram shows uses for the bits in the registers.

When you push the RESET key, you reset all of the registers. RESET also configures registers A, B, and C to send data to the processor.

The TK-80A board does not use all of the PIA capabilities. It is designed so that register A (100F8) is input from the key matrix, register B (100F9) is input from the tape circuit, and register C (100FA) is output to enable the rows in the key matrix, tape output, and display blanking. The value the 9010A must send to the PIA (address 100FB) to set up the registers for this mode of operation is 92 (hex).

The PIA toggles a bit in the C register when you write 0X (hex) to address 100FA. The X represents four bits in the following pattern:

bit 0: 0=clear; 1=set
bits 1-3: tells which C register bit (0-7) to set.

An example of use is to write hex 01 to address 100FA to send a 1 to the tape circuit, or hex 00 to send a zero.

The key matrix consists of three rows of eight keys. The RESET key is not included because it is interfaced directly to the processor.

When the 9010A writes a zero to one of the row bit positions in address 100FA, the respective row of keys is enabled. When you press one of the keys in that row, a zero for that key is latched into the PIA register A. When the 9010A reads that register at address 100F8, it will receive a byte of all ones except for the zero bit(s) for the key(s) you pushed.

The bits of address 100FA which enable the rows of keys are as follows:

bit 4=0 - enable keys 0-7;
bit 5=0 - enable keys 8-F;
bit 6=0 - enable the colored keys.

Tape Input/Output

The processor interfaces with an external tape recorder via the PIA and tape input/output circuit. The processor program (routines contained in ROM) must work with the tape interface circuitry to save data on tape.

The tape output circuit outputs one of two frequencies, depending on the output of bit 0 of the PIA C register. If the bit is a 0, the tape output circuit outputs a 2400 Hz sine wave. If a 1, it outputs 1200 Hz.

The output circuit sends a zero-crossing signal to bit 7 of the PIA B register (address 100F9) on the rising edge of each cycle of the output frequency. This allows the processor to count the number of cycles of output and aids it in determining how long it has been sending a 1 or a 0.

The program must also determine the protocol, such as how many start and stop bits are sent with a data byte. Further, it must strip any start and stop bits from the data bytes received from the tape.

The tape input circuit reads data from the tape recorder. The signals are the same as those recorded. The circuit filters the signals. Then it turns any high voltage level into a logic 0 and any low level into a logic 1. It sends the logic levels to bit 0 of the PIA B register. It is up to the processor program to check the timing of the input signal to determine its frequency.

The tape circuits provide two jacks for connection to an ordinary cassette tape recorder. The A/M jack is for connection to the recorder's AUX or MIKE input. If MIKE is used, then R19 (51 ohms) must be installed (see schematic) to attenuate the output signal. The EAR jack is for connection to the recorders EARPHONE or EXTERNAL SPEAKER output.

HOW TO CHARACTERIZE THE UNIT UNDER TEST (UUT)

Introduction

The preceding information on the TK-80A board, in conjunction with the schematic diagram, may have given you enough understanding of it as a UUT to start testing and troubleshooting right away. However, it is not always practical to consult a schematic or block diagram description. Further, many schematics are cryptic, requiring you to consult a data book to understand the operation of the single-chip circuits.

Assume that you have just been handed the TK-80A board to test and troubleshoot. Your approach would likely be to test and troubleshoot any failures in the following circuits in the order given:

1. Power and clock
2. Processor bus
3. Memory and I/O (includes locating them)
4. Display
5. Keyboard
6. Tape I/O

The 9010A lets you test the UUT from the inner kernel (the processor) to the peripheral circuits. You will be using the 9010A to achieve the above approach in the exercises which follow. At the same time you will be learning how effective the 9010A is for testing and troubleshooting microprocessor-based systems. During the exercises, refer to the schematic and block diagrams as necessary.

Registers and the Display

The 9010A contains 16 registers, 0 through F, hexadecimal. Each register holds 32 bits or 8 hexadecimal digits. The binary equivalent of the hexadecimal digits is engraved on the 9010A's hex keys. As an operator or programmer you may enter data into or examine the contents of any of the registers.

Certain 9010A operations can automatically enter information relating to the UUT into registers 0 and A through F. Although they will be elaborated on more later, they appear in Table 2.

Table 2. 9010A Special Uses for 9010A REGISTERS

REG	USE
A	Bit Mask for I/O tests (indicates which bits to test);
B	ROM signature (computed or entered during ROM test or LEARN);
C	Status or Control line pattern (see pod decal for bits);
D	Bit # for TOGGL ADDR or TOGGL DATA;
E	Read/write data from/to UUT;
F	Read/write address for above data;
0	Reading from probe during READ-PROBE operation, as follows: bits 0-7 = number of events counted; bits 8-23 = signature of states in sync with Data/Address Valid; bits 24-26= history of logic levels sensed at probe tip: 24 = 1 if it sensed a HIGH; 25 = 1 if it sensed a high-impedance (tri-state) state; 26 = 1 if it sensed a LOW.

You can examine and change the contents of any register. For example, press REG 1 ENTER. The 9010A displays the contents of register 1. To change it press REG 1 12345678 ENTER. Examine the contents of register 1 again. What does it contain?[2] What happens if you try to enter nine digits?[3]

You can increment, decrement, and right-or-left-shift the contents of the registers. And you can perform logical operations with them. For example, enter 12345678 in register 1, then press REG 1 INCR SHIFT-LEFT DECR SHIFT-RIGHT ENTER. What does the display do?[4]

The register contents do not change until you press ENTER. For example, press REG 1 123 REG 1. What does it contain?[5]

------------------ ANSWERS TO QUESTIONS IN TEXT ------------------

2. 12345678
3. Pressing the 9th digit causes the 9010A to enter the preceeding 8 digits and beep.

Remember this: the 9010A flashes a cursor any time it expects you to make an entry. If you fail to make entries until the cursor no longer flashes, then the operation will not be performed.

You can also set one register equal to the value of another. For example, press REG 2 REG 1 ENTER. The 9010A displays REG2 = 12345678 to indicate the contents of register 1 were duplicated in register 2. What happens to the contents of register 1 in the process?6

You can use the registers to perform AND, OR, and COMPLEMENT Boolean operations, too. For example, press REG 1 COMPL. The 9010A assumes you want to complement the register just entered. What does the display show?7 Now press AND CLEAR. You would press CLEAR if you had pressed AND by mistake. What does the display do?8

To continue the Boolean operation, press COMPL OR REG 2 ENTER. Why does the display show REG1 = FFFFFFFF?9 Press REG 1 REG 2 AND ABC ENTER. What does the 9010A display and why?10

To get a bit more familiar with the 9010A keyboard, press CLEAR. Notice that the 9010A beeps at you and does not clear the display. The 9010A will beep any time you press a key which is illegal under current conditions. It clears the display any time you press another key which tells it to do an operation.

For example, it will also beep if you press a numeric key. You can also clear any current operation by pressing STOP. Try it. What is displayed as a result?11

Setup Information

The 8080 processor usually responds to not-ready, hold, interrupt, and reset by halting the current operation. However, the SETUP feature lets you change the way the 9010A responds to these as well as other preset conditions.

For example, press SETUP on the 9010A. Then press the MORE and PRIOR keys. Notice that the display scrolls through a list of 19 question-type messages. They notify you of the current setup conditions. By pressing YES and NO, you can change the setup of the displayed item.

Take for example the RESET line. You can see from the block diagram of the TK-80A that the RESET key sends a RESET signal to the processor. If in SETUP you have enabled TRAP ON ACTIVE FORCE LINE (YES), then when you press

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

4. It displays 12345679, then 2468ACF2, then 2468ACF1, then 12345678, and ends up at 12345678.
5. 12345678.
6. Register 1 remains unchanged.
7. EDCBA987, which is the ones (bit-for-bit) complement of 12345678. It also shows a flashing cursor, indicating that register 1 has not been changed yet.
8. It deletes everything but REG1 =.
9. Because a logical OR of a number with its complement sets all bits (F's).
10. 238 because it is the result of a logical AND of 12345678 with ABC.
11. FLUKE 9000 READY

the TK-80A RESET key while the 9010A is doing a read or write operation, the 9010A suspends the operation and displays ACTIVE FORCE LINE-LOOP?.

This message notifies you that the pod has sensed one of the lines which normally forces the processor to stop what it is doing. To prove this, press the 9010A BUS key. Then press LOOP. Notice that the LOOP indicator is flashing while the 9010A continuously runs the BUS test. Now press the TK-80A RESET key. What is the displayed result?12

Now use SETUP to disable TRAP ON ACTIVE FORCE LINE (press NO). Loop again on BUS test and press TK-80A RESET. Notice now that the test continues without interruption.

It is important to note that you may be unable to test some micro-systems without first changing one or more SETUP items. For example, the 8080 suspends operation when an external circuit sends it not-READY or HOLD. SETUP allows you to enable or disable these signals. If enabled, the 9010A will display a message when they occur. If disabled, the 9010A will sense them but will not respond to them. As another example, you may know that the UUT power supply is marginal, but you still want to run some tests. In this case you set TRAP ON BAD PWR SUPPLY to NO. Otherwise, the pod will interrupt a test as soon as it senses the marginal supply voltage.Thus, it is possible that you must change SETUP conditions to get some UUTs to run at all.

In a similar fashion, the other SETUP items allow you to set the background conditions for operating the 9010A. We will discuss other SETUP items as they pertain to other exercises.

Bus Test and Error Handling

Before finding out where the UUT ROM, RAM, and I/O are located, you will want to make sure the data, address, and control busses are functional. When you press BUS test, the 9010A checks the data and address lines to make sure they are not stuck high or low, nor tied to each other. The 9010A only checks control lines for a stuck high or low condition.

While running any 9010A operation, notice the consistency of its reporting. During the test it displays WAIT to let you know the test is in progress. It notifies you when a failure occurs and asks you if you wish to loop on the failing operation.

If you press NO or CONT, it continues at the next step of the test. If you press YES or LOOP, it loops on the failure and flashes the LOOPING light. If during the loop you press STOP, it stops at the end of the current test step and flashes the STOPPED light. If you then press CONT, it continues the test at the next step.

When it reaches the end of the test, it displays OK if the test passed, and FAIL if it failed. Because of this reporting, you never have to guess what the 9010A is doing. Now let's see how the bus test works.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

12. If TRAP ON ACTIVE FORCE LINE is enabled, then ACTIVE FORCE LINE-LOOP? is displayed. If not, then BUS TEST OK is displayed if there are no errors.

CAUTION

> For some of the exercises, you will be inserting jumpers between circuits on the TK-80A board. DO NOT SHORT POWER TO GROUND. If you do, the TK-80A power cord's fuse(s) will blow.

Get a micro-clip jumper from your kit. Jumper U34 pin 4 (DBIN) to pin 14 (ground). DBIN is the signal which gates data bytes into the processor. Now press BUS test. What indication does the 9010A give you of a failure and what does it mean (hint - see pod decal)?13

Press LOOP or YES in answer to the LOOP? question. Notice that the LOOPING light flashes. This causes the 9010A continuously to exercise the step of the bus test which failed. It does not test any other part of the bus. What happens if you alternately insert and remove the jumper?14

The 9010A allows you simply to run the test without concern over whether it fails. Say you want to probe the circuit with an oscilloscope as you run the test. In such a case you could change the 9010A setup.

For example, press SETUP. Now press MORE until the 9010A displays SET-EXERCISE ERRORS? Press NO. Ensure the jumper is in place and press BUS. Notice how the display shows WAIT then FAIL, without reporting the nature of the error. Any time you disable exercise of errors, the 9010A sends the error message out the AUX I/F (RS-232) port on the rear of the unit. It allows you to print the errors rather than to display them.

Press LOOP. Notice how the 9010A beeps each time the test fails. This could become nerve-wracking.

The 9010A will let you disable the beep. Press SETUP MORE. The 9010A displays SET-BEEP ON ERROR TRANSITION? YES. Now press NO BUS LOOP. What is the 9010A doing?15 Now use SETUP to reinstate EXERCISE ERRORS and BEEP ON ERROR TRANSITION to YES.

Press BUS again so that the failure occurs, and press LOOP to make the 9010A loop on the error. Remove the jumper and connect in on U34 between pins 15 and 17 (data bits 0 and 1). What is the 9010A's response to this and why does it occur?16

-------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

13. It displays CTL ERR 00000000 00010000--LOOP? and flashes the STOPPED light. From the pod decal we see the status and control lines' bit positions in the above message. Since bit 4 = 1, we know the DBIN line is not working properly.
14. The 9010A beeps each time the jumper configuration is changed, that is, on each error transition. The display alternately displays the failure message and CTL OK 00000000 00000000.
15. It is flashing the LOOPING light, and occasionally the STOP light while it alternately displays BUS TEST WAIT and FAIL. It is repetitively running the bus test and is not stopping on failures. The STOPPED light blinks on each time the failure occurs as an indication that the 9010A is not exercising errors.
16. It displays the CTL OK message. It does not indicate a failure because the 9010A is not exercising the part of the test which caused the error.

Press CONT. When you do this, the 9010A leaves the loop condition and continues the bus test from where it left off. What indication does it give of a different failure?17

Press CONT or NO. Since the jumper is still installed, the 9010A displays BUS TEST FAIL. Remove the jumper and press RPEAT. This causes the 9010A to repeat the last action you commanded it to, the bus test. Notice now that the test passes.

One additional SETUP item relates to bus test: SET-BUS TEST @ FFFF-CHANGE? This item sets the address at which to test data lines. You could change the address so that it would affect a particular external circuit. To see the effect, for example, press SETUP, then MORE until the above message is displayed. Then press YES 8FFF ENTER BUS LOOP, and watch the TK-80A display. What is happening and why?18

Remember this: the loop feature allows the 9010A repetitively to exercise only the portion of the test that failed. What is the benefit of this type of error handling?19

Learn and View

The 9010A allows you to locate RAM, ROM, and I/O using the LEARN key. You would use this feature if you did not know what memory addresses were present, or if you wanted to verify that they were good.

During LEARN the 9010A executes a rather complicated pattern of reads and writes to the entire address range permitted by the pod. Based on the results, it identifies what type of memory it thinks is present, and at what addresses. VIEW allows you to see what information the 9010A has stored regarding ROM, RAM, and I/O. While LEARN allows the 9010A to change address descriptors, VIEW allows the operator to inspect or change them.

To see how LEARN works, you could press LEARN ENTER. The 9010A would run several minutes before displaying LEARN OK. But this is unnecessary because we know the TK-80A has ROM in addresses 0-7FF, RAM in 8C00-8FFF, and I/O in 100F8-100FB. Therefore, we will learn address spaces in sections.

SIG-F77C

Learn RAM

Let's start by learning RAM. Press LEARN 8BFF ENTER 9000 ENTER. What is the display doing after you made the last ENTER and what does it mean?20

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

17. It displays DATA BITS 0 AND 1 TIED-LOOP? and flashes the STOPPED light.
18. The right-most digit is flashing all segments periodically because the data line test is being run on the direct-memory-access address for that digit.
19. It allows the technician to do things to the UUT to try to find the error while the 9010A is looping on it. For example, the technician can blow coolant or hot air on the board's components, flex the board, wiggle socketed chips, or brush away solder splashes. If any action causes a beep from the 9010A, the technician knows the area which failed.
20. It displays LEARN @ 8BFF-9000 NOW 8C00, and increments the now part by 100. This is telling us what block of 100 hex addresses is currently being identified. It lets us know the 9010A is actually doing something. When done, it displays OK in place of NOW 8F00.

Now press the view RAM key. What is the display telling you?21

Learn ROM

Now let's verify the location of ROM. Press LEARN 0 ENTER 800 ENTER. What is different about the way the 9010A does a ROM learn?22

Now press view ROM. How is the display different from when you pressed view RAM?23 What was the 9010A doing while it was displaying WAIT and why?24

The addresses you see with the view keys are called address space descriptors because they describe where the UUT's addresses are. Your UUT could have a lot of blank ROM addresses which you do not want to test, or it could have non-contiguous blocks of ROM addresses. It could also have multiple ROM chips for which you wish to save individual signatures.

In such a case you may want to enter more than one set of ROM descriptors. The 9010A allows you to do that.

For example, assume the TK-80A ROM were in two chips, addresses 0-3FF and 400-7FF (one kilobyte each). To enter the ROM descriptors for the first block, do the following: Press view ROM ENTER 0 ENTER 3FF ENTER ENTER. The 9010A displays WAIT while it computes a signature for the block.

Assume you know the signature for 400 to 7FF is D23F. Enter the descriptors and signature: press ENTER 400 ENTER 7FF ENTER D23F ENTER. The 9010A has now stored the address descriptors and signatures for all three blocks of ROM. To see them, press view ROM. What indication do you get that there is more than one set of ROM descriptors stored in the 9010A?25 Press the MORE key. What are the signatures for the 3 blocks of ROM?26

A final item to note about learning memory is that it is useful for quick ROM integrity check. To demonstrate this, connect a jumper between +5 Volts at the right-hand end of C27 (just above the TK-80A power plug) and U24 pin 11. This shorts address bit 10 to logic high. Now learn ROM anew by pressing LEARN 0 ENTER 7FF ENTER and waiting for the OK message. Now press view ROM. What can you deduce from the 9010A display?27 Remove the Jumper.

Since you will use the ROM descriptors further on for testing, it would be a good idea to put those for the actual ROM back into the 9010A. Since you

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

21. It tells that RAM exists from address 8C00 to 8FFF.
22. After going through the addresses, the 9010A displays LEARN @ 0-800 WAIT. After about 15 seconds it displays OK in place of WAIT.
23. The display shows that ROM exists from address 0 to 7FF, as we suspected. But it also shows a signature of F77C.
24. It was computing a signature of the entire contents of ROM. The signature is likely used for comparison when testing ROM.
25. The MORE light is flashing. Any time the 9010A has more than one line of message information to display it flashes the MORE light. Pressing the MORE key causes the next line to be displayed.
26. The signatures are: 0-7FF = F77C; 0-3FF = 0ED3; 400-7FF = D23F.
27. The display shows that only addresses 0-3FF exist and have a signature of D23F. This makes it obvious that the board contains a high-address line fault because the signature is correct for addresses 400-7FF and we know that the board actually contains addresses 0-7FF.

know the signature and the address range, you can enter them directly. Press view ROM. Notice that the faulty set of descriptors appears.

Press CLEAR. What comes up on the display and what does it mean?[28] Now press ENTER 0 ENTER 7FF ENTER F77C ENTER. The correct descriptors are now stored.

Learn I/O

The TK-80A does not contain the type of I/O registers which the 9010A learn algorithm can identify. The reason is this: in order to use the I/O registers, you must set them for input or output as explained in the block diagram description.

The 9010A has no way of predetermining this. It expects I/O registers to have one or more bits which it can write to and read from. Therefore, to get complete information on I/O, you may have to consult the schematic diagram, theory of operation, or IC data book.

The 8080 data book describes the 8255 PIA and how to set it up. It also describes how the 8228 bus system controller works and how to get the 8080 to address I/O space. Further, the 9010A looks at I/O as being in the range of addresses from 10000 to 100FF hex. From the descriptions we know the addresses for the PIA are 100F8-100FB.

You can use the same procedure for entering I/O descriptors as you used for memory descriptors. Press view I/O and note the display of NO I/O INFO. Now press ENTER 100FA ENTER ENTER FF ENTER. This will set up a bit mask of FF so that only the bits in register 100FA (PIA C register) which correspond to the 1-bits in hex value FF (1111 1111) will be tested by the I/O test. You can successfully run the I/O test on this address once you have set up the PIA.

At this point you have characterized the TK-80A board. You have determined that its busses are functional. And you have identified its available memory. Next, you will want to test it.

HOW TO USE THE BUILT-IN TESTS

Introduction

We have already discussed how to perform a BUS test. It is important to run BUS test before attempting any other operation on the UUT. The reason is that no other operation is reliable without first guaranteeing the integrity of the busses.

However, the 9010A does have a variety of other built-in tests for memory and I/O. The following exercises will demonstrate how to run them successfully.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

28. The 9010A displays NO ROM INFO because we just cleared the only ROM descriptors present. This means that the last LEARN cleared the descriptors previously entered.

RAM Tests

Notice the two RAM test keys on the left side of the keyboard. The RAM SHORT test simply verifies that each address in the range tested can write and read ones and zeros and that no other addresses or bits are affected. However, the RAM LONG test does complete address-decode-circuitry and pattern-sensitivity testing. It tests whether memory cells are affected by write-read activities to neighboring cells. And it runs considerably longer than RAM SHORT. In fact, a 64-kilobyte-by-8-bit memory would take several hours to test.

Because the TK-80A display circuit uses direct memory access to refresh itself, you can run RAM SHORT and RAM LONG on those addresses to get an idea of what the tests are like. You can SEE the results.

First, press RESET on the TK-80A to ensure the display is unblanked. Then press RAM-SHORT 8FF8 ENTER 8FFF. Watch the TK-80A display and press ENTER. Now watch the 9010A display and press RPEAT. What did the TK-80A and 9010A displays do?29

Jumper U34 pin 14 to pin 16 and pin 7 to pin 28. Press RPEAT. When the test fails, press LOOP. Alternately connect and disconnect the jumpers. How can you tell which bits are failing?30 Disconnect the jumpers.

Now press RAM-LONG 8FF8 ENTER 8FFF ENTER and watch the TK-80A display. Notice the difference in the display pattern and length of test. Press LOOP and watch the display. Although the test is interesting to watch, you could not detect a memory problem by looking at the display unless it were a solid failure somewhere in the eight display addresses. However, as we will see later, it can be used to test the display.

You do not have to enter address descriptors to run RAM tests. For example, press RAM-SHORT ENTER. Notice that WAIT is displayed while the test is in progress. What addresses did the 9010A test, and how did it know what they were?31

ROM Test

The 9010A allows you to enter addresses for ROM tests the same as for RAM tests. However, it tests ROM by computing a signature for the range of addresses specified, then comparing it to another signature.

For example, press test ROM 0 ENTER 3FF ENTER F77C ENTER; note the failure. Now press test ROM 0 ENTER 7FF ENTER ENTER. Why did it fail then pass?32

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

29. The TK-80A display flashed briefly, then ended up with some strange characters in it. The 9010A displayed RAM SHORT @ 8FF8-8FFF WAIT briefly, then replaced the WAIT with OK.
30. The display shows R/W ERR @ 8FF8 BTS {82,80,02}. This means bits 2 and 7 (the third and eighth bits) failed the read/write test. The one-bits in the 82 value tell which bit positions failed.
31. It tested addresses 8C00-8FFF. It knew to test them because they are the addresses the 9010A discovered during LEARN.
32. It used default signature F77C (addresses 0-7FF), but failed because it computed signature ED3 (addresses 0-3FF). It then passed because the correct block of addresses for the F77C signature is 0-7FF.

Recall from the discussion of registers that the B register contains the ROM signature. The signature is entered into the register at the time you press the last ENTER to run the test. Press REG B ENTER to verify that it contains F77C.

A benefit to having one signature for each ROM chip is that a failing ROM test will point to the bad ROM. This makes it unnecessary to have a signature stored for the entire set of ROMs. In fact, if you do, then the ROM test will take unnecessarily long to run. The reason is that the test runs for each set of descriptors present.

I/O Test

You can test I/O similar to memory. However, with the TK-80A it is not as straight forward. The reason is that you must set up the PIA first.

The setup requires you to write hex 92 to the setup address 100FB. To do this, press WRITE 100FB ENTER 92. Now watch the TK-80A and press ENTER. What happened to the TK-80A and why?33

Now run the I/O test. Remember that you will be testing all of the bits in the PIA C register (address 100FA) because that is what you entered for the I/O descriptors. Press test I/O ENTER LOOP. What is happening to the TK-80A display and why?34

Now let us examine the 9010A F register. Press REG F ENTER. Similarly examine the contents of registers A through E. Which ones were affected by the I/O test and why?35

To demonstrate the ease with which you can use the registers to hold default values, press test I/O REG F ENTER ENTER REG A ENTER. Notice that the test runs the same as when you let the 9010A use its internally stored default values. What happened when you pressed REG F?36

Auto Test

AUTO is a single-button substitute for BUS, ROM, and RAM SHORT tests. It simplifies quick-testing for high-volume integrity checks and for inexperienced operators. Any of the items which caused BUS, ROM, or RAM SHORT failures will also cause AUTO to fail.

To demonstrate this, connect the jumper on U38 between pins 8 (memory read enable) and 7 (ground). This disables a read operation by grounding the memory chip select circuit. Press AUTO. Note that the test executes

------------------ ANSWERS TO QUESTIONS IN TEXT ------------------

33. The TK-80A display went blank because sending the setup command reset the PIA registers. Bit 7 of the C register must be set to unblank the display.
34. The display is turned on and flickering slightly because the test sets bit 7 of the PIA C register most of the time.
35. The F register contains the I/O address tested, 100FA, because it was the last address accessed. The A register contains FF, the bit mask the 9010A used to identify which bits in the I/O register to test.
36. The display showed 100FA, the contents of the register, in place of the register name.

immediately and displays WAIT until a failure occurs. After each failure message, press CONT several times. What indication do you get that it is a decode problem?37 Remove the jumper. Press RPEAT to rerun the test and ensure it passes.

HOW TO USE BUILT-IN TROUBLESHOOTING FUNCTIONS

Introduction

You can use the troubleshooting keys, READ, WRITE, RAMP, WALK, TOGGL ADDR and TOGGL DATA to stimulate or monitor a particular address or bit on a microprocessor bus. All you need to do is press the key and enter an address, data, and/or bit pattern. The 9010A immediately performs the operation. As in previous examples, you may use the default values contained in 9010A registers where appropriate.

Read

The READ function allows you to examine the contents of a specific address. For example, press READ 0 ENTER. The 9010 displays 22, the value in address 0000. The OK only means that the 9010A has completed the read operation successfully; it does not verify that the address contents are correct. If you enter an address beyond the I/O limit of 100FF, the 9010A will display an error message.

You can also use the F register to automatically supply the address for a read. For example, press REG F 1 ENTER READ ENTER. Notice the display shows the contents of address 1 = F0. As well, you can use the arithmetic keys. For example, press READ INCR ENTER. Notice the display shows the contents of the address in register F incremented (address 2) = 8F. As an exercise, what do addresses 3DF, 7FF, 999, and 2 contain?38

Read Status

The 9010A can also read the status lines at the microprocessor socket. For example, press READ STS/CTL and note the 9010A display. What status line is held high?39

Press LOOP. Alternately press and release RESET on the TK-80A board. What indication do you get that you pressed reset?40

Jumper U38 pin 2 to pin 7 (ground). Note the POD TIMEOUT ERROR and flickering TK-80A display. From the schematic, note that you have forced the HOLD signal (normally not connected to anything) to the pod.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

37. Failing addresses are successive, showing all bits to be bad. It is likely that an area of memory which does not exist or no memory at all is being addressed.
38. 3DF = 9E; 7FF = 13; 999 = FF (non-existent address); 2 = 8F.
39. READY (refer to pod decal)
40. Bit 4 of the status word toggles.

The 9010A is set up to acknowledge the HOLD because it is a signal which the UUT often uses to stop the processor while it does such things as direct memory accesses (completely a designer prerogative). When HOLD is true, the POD stops communicating with the 9010A mainframe because it is being held. The mainframe waits a preset amount of time for communication. If none happens, it displays the timeout message and sends a reset to the pod to start it back up again.

Some UUTs use the READY and HOLD signals in unique ways so that they normally are in a state to stop the processor. For this reason the 9010A allows you to disable HOLD, and to change the amount of delay before a timeout is sensed.

For example, press SETUP. Press MORE until the 9010A displays SET-ENABLE HOLD? YES. Press NO to disable it. Now press READ STS/CTL LOOP. What is the difference in 9010A display from the last read status?[41]

Now return ENABLE HOLD to YES, and press MORE until setup message SET-TIMEOUT 200-CHANGE? appears. To change the time press ENTER 9999 ENTER. Press READ STS/CTL again. Notice the increased delay before the POD TIMEOUT MESSAGE occurs, and notice the TK-80A display flashing less often. Now change the timeout parameter back to 200 and remove the jumper.

Write

You can use the write function which allows you to send data to a specific address. For example, to write to the display, press WRITE 8FF8 ENTER FF ENTER. What did this do to the TK-80A display?[42]

For your reference, figure 3 shows the hexadecimal values required to turn on the individual segments of the display.

```
 ____01____
 |        |
 20       02
 |___40___|
 |        |
 10       04
 |___08___|  o 80 (decimal point)
```

Figure 3. Hex Codes for TK-80A Display Digit

To determine what hexadecimal number to write to the display address for a given combination of segments, just add together all the hexadecimal numbers, for lighting each segment that you want lit. Remember, you are adding base 16 numbers: 9 + 1 = A, not 10. As an exercise use the write function to display {1234. 5678} on the TK-80A. What values will you send what addresses?[43]

------------------ ANSWERS TO QUESTIONS IN TEXT ------------------

41. There is no more POD TIMEOUT, and status shows OK with bits 0 and 1 set. Bit 1 is the HOLD line.
42. It lit all the segments in the left-most digit.
43. Address 8FF8=06; 8FF9=5B; 8FFA=4F; 8FFB=E6; 8FFC=6D; 8FFD=7D; 8FFE=07; 8FFF=7F.

I/O Read/Write

Now that we have tried the read and write functions, let us use them to operate the PIA. Notice from the block and schematic diagrams that the PIA interfaces the keyboard, tape circuit, and even display control to the processor.

Let's set up the PIA to look at keyboard inputs. Press WRITE 100FB ENTER 92 ENTER. Now the PIA is set up to input from the Tape circuit, output to the keyboard columns, and input from the keyboard rows.

To read the keyboard, the keyboard column register in the PIA must be set to zero, which it was when you wrote 92 to 100FB. Note that the columns are schematic columns only. They are not laid out in the same columns on the board. However, you may want to unblank the display. To do so, press WRITE 100FA ENTER 80 ENTER. This sets the display unblank bit, and ensures the lower three bits are zero. Bit 0 enables keys 0-7, bit 1 enables 8-F, and bit 2 enables the rest.

Now press READ 100F8 ENTER LOOP. The 9010A is now continually reading the keyboard input register. Note from the schematic that the three columns of keys are in parallel. This means that three keys will produce the same number. Press the keys at will. What three keys produce the number BF?[44]

In practice, you would have to write a program which would sequentially enable the keyboard columns before reading the rows. The reason is that otherwise you would not know which column had a key pressed down.

The tape circuit is a bit more complicated than the keyboard circuit. But you can test it fairly easily. Press READ 100F9 ENTER LOOP. How does the display verify that part of the tape circuit is working properly?[45]

Now for the tape data in and out. Connect your jumper between the EAR and A/M jacks. Clip to the unsoldered ears which protrude from the under side of the board directly beneath the jacks. This connects the tape output circuit to the tape input circuit. The 9010A is still looping on the read at address 100F9. How can you tell from the display that the tape circuit is working properly?[46] Remove the jumper.

Write Control

You can also change the state of processor control lines. They are different for each pod. Those for the 8080 are shown on the pod decal. You can write to only three of them: INTE (interrupt enable to allow an external circuit to interrupt the processor); HLDA (hold acknowledge which

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

44. 6, E, and BRK.
45. Bit 7 is cycling on and off very fast. This is the zero-crossing bit which goes high (1) on every rising edge of the tape oscillator output, and low (0) on every falling edge. This verifies that the oscillator and path to the PIA are working.
46. Now not only is bit 7 toggling to show zero crossing, but also bit 1 is toggling. They seem to be synchronized. Since the tape input circuit sends a 0 to the PIA bit 1 when the input signal is high (and vice versa), this means the PIA is receiving 1's and 0's from the tape input circuit. Therefore, it is working.

the processor sets concurrent with setting the data and address bus to the high impedance state); and WAIT (to notify memory or another external device that the processor has received not READY).

To set the interrupt line, press WRITE STS/CTL 100 ENTER. The 100 is a binary value in which the 1 occupies the third bit position. From the pod decal you can see that the third bit is the INT line.

The write control value was just stored in register C. To verify this, press REG C ENTER. Why is the value not 100?[47]

The TK-80A is designed so that you can see the effect of writing to these control lines only when using the probe, which is discussed in a subsequent topic.

Ramp

You can use the RAMP function to write incrementing data to any valid address. The function starts at 0, then continually writes and increments the value by 1 until it reaches FF. You can demonstrate this function by writing a ramp to one of the TK-80 display addresses.

For example, press RAMP 8FF8 ENTER. Now press RPEAT several times while watching the display. Notice that the left-most digit runs rapidly through a pattern.

A function such as this can be useful for testing digital-to-analog converters or simply to exercise an address which failed a memory test.

Walk

The WALK function continually writes a data pattern to an address and shifts it to the right circularly until it has shifted eight times. As an exercise, walk a pattern through a display digit. Press WALK 8FFF ENTER A3 ENTER.

This function is performed so rapidly that you can hardly see the patterns as they are written to the display, even when in the LOOP mode.

Toggle Address

The TOGGL ADDR function toggles a specified address bit from one logic state to the other. For example, press TOGGL-ADR 8FFA ENTER 1 ENTER. The 1 is the hex bit number to toggle. Notice that this does not affect the TK-80A display. The reason is that the function performs two read operations: one at address 8FFA and one at 8FFA with the second bit toggled. What is the second address?[48]

This function is useful for troubleshooting a suspect decode circuit problem with the probe which we will discuss in a subsequent topic.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

47. Because 100 is binary and the register is displayed in hexadecimal. Hex 4 equals binary 100.
48. 8FF8

Toggle Data

The TGGL DATA function toggles a data bit from one logic state to the other much the same way that it was done for the TOGGL ADDR function. This is easy to see at a display address.

Assume we want to light the top segment of the first digit of the TK-80 display. Press TOGGLE DATA 8FF8 ENTER 1 ENTER 0 ENTER. This operation will write 1 to address 8FF8, then write the same value with bit 0 toggled to a 0.

Again this function goes fast to easily see in the display, so press LOOP. Now the 9010A is writing 01 and 00 to the display address. This is shown in the first digit of the TK-80 display. What would the data be to turn on only the decimal point of the display and what bit would you toggle to turn it off? Try this and press RPEAT a few times to see the effect.49

You can also toggle a specified control line by using the TOGGL DATA and STS/CTL keys. For this example you will want to toggle the control line labeled WAIT. Press TOGGL-DATA STS/CTL 0 ENTER 1 ENTER. This function can be demonstrated best with the PROBE, and will be in a subsequent topic.

HOW TO EMULATE THE UUT

The RUN UUT key causes the processor in the pod to run a program in UUT memory starting at a specified address, default 0. Since the pod clock is taken from the UUT, the program will run at normal UUT speed. This makes it fairly simple for you to compare UUT operation with a known-good processor (in the pod) to operation with the normal processor. During RUN UUT operations, the SETUP conditions for forcing lines, READY, HOLD, and INTERRUPT are ignored and the pod responds as a normal processor would.

The TK-80A operating system starts at address 0. This is handy because pressing TK-80A RESET also starts the processor at address 0. To demonstrate this feature, press RUN-UUT ENTER. The display tells you that you may need to press the reset key. This is unnecessary on the TK-80A. In RUN-UUT the TK-80A responds to keys you press.

For example, to read the contents of the display addresses, on the TK-80A board press MEM 8FF8 NEXT. The left four digits are the address; the right two digits are the contents. Now as you press NEXT and PREV, higher and lower address contents are displayed. Use those keys to scroll through display addresses 8FF8-8FFF. Why do the displayed contents for the fourth, seventh and eighth digits depend on which direction you scroll?50

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

49. Data = 80; Bit = 7
50. Because the operating system reads memory before displaying what is in it. Therefore, it displays what was last at that digit position before NEXT/PREV was pressed.

HOW TO USE THE PROBE

Introduction

Although we have learned a lot about the UUT and its condition without using the probe, the probe is a powerful troubleshooting tool. It is both simple and straightforward to use.

The probe is bi-directional. It can inject logic signals into circuits when the HIGH and/or LOW key is in the down position. It's red and green lights can display the logic activity in a circuit. And through the READ PROBE operation it can provide the 9010A with a count of falling-edge events, history of logic levels sensed, and signature at the point being probed.

You can synchronize the probe signature-taking and signal-injecting capability to the time when the processor (pod) is placing an address or data on the bus, or let it free-run at 1,000 Hz. These synchronization modes are known as Address Sync, Data Sync, and Free Run.

The idea of taking signatures and counting events is this: you stimulate a known-good board in a known way. Then take signatures/event counts where appropriate and store them on the 9010A tape. Then, when troubleshooting a faulty board, the 9010A can compare the actual signature/count with the expected signature/count to isolate the fault to a component.

The probe fuse socket is next to the probe jack. The fuse protects the mainframe circuitry from damage if the probe draws excessive current. The 9010A will display a message telling you whether it is blown.

Before using the probe, ensure the probe ground clip is connected to a TK-80A ground, such as the left end of C25 at the top of the board.

Read Probe and Register 0

The READ PROBE operation is a passive, data gathering operation, and does not by itself provide any stimulus to the logic node. The operator must cause the 9010A to perform a function (such as READ, WRITE or RAMP) which stimulates the node while the 9010A is accumulating data with the probe.

Register 0 stores the following response data from the probe: events count, the logic level history and the signature computed from successive logic states. See the topic on registers for the format register format. Note that no signatures are generated in FREE-RUN.

The Data in the Read Probe register is accumulated from one READ PROBE selection to the next READ PROBE selection. Whenever you perform a READ PROBE operation, the present register data is displayed and the register is cleared for the next measurement period. The 9010A automatically breaks down the contents of register 0 and displays it in the form

PROBE-LVL LXH COUNT ccc SIG ssss.

The LXH is the low, tristate, and high logic history at the probe tip since the last read probe operation. The ccc is the asynchronous count of events since the last read probe. The ssss is the signature computed from the logic activity since the last read probe.

To try out the read probe operation, probe U40 pin 6 and press READ-PROBE several times while watching the 9010A display. Notice how the system clock causes a change in the count. Now, verify that register 0 contains the data taken in the last read probe operation. What arithmetic operations would you do on register 0 so that only the signature was left in it (refer to Table 2)?51

Synchronization

As mentioned, there are three sync modes: address (A), data (D), and free run (F). The 9010A powers up in free run. To select a mode, press SYNC, then either A, D, or F. To see which mode is currently selected, press SYNC ENTER. Select free-run as follows: press SYNC F. The selected sync mode affects all of the probe operations as shown in Table 3.

Table 3. Relationship of Probe Uses to Sync Modes

PROBE USE	ADDRESS/DATA SYNC	FREE-RUN SYNC
Pulses (2 usec)	1 per A/D Valid	1000 Hz (Hi>4V, Lo<.2V)
Signature	Sensed @ A/D Valid	Not sensed
Events *	Always sensed	Always sensed
Logic History (see Table 4)	Sensed @ A/D Valid; Tristate not sensed	All states always sensed

* An event occurs only as the signal falls below the high threshold

In free run, the 9010A can issue 2 usec high and low pulses alternately, and at a 1 kHz rate. It also counts events and illuminates probe lights asynchronously, as logic activities occur. However, it does not take signatures because they would not be repeatable since they are not synchronized to anything.

With the 8080 pod, address and data sync are the same because they are concurrent. The effect of address and data sync is to freeze the action in the circuit: it allows you to see only the activity at the probe tip during the instant the pod places a valid address or data on the bus. During these sync modes, the 9010A counts events asynchronously, but it gathers signatures and logic histories, and illuminates probe lights only during address/data valid. In this way you have a stroboscopic window into the microsystem during read and write operations.

Also, consider that neither address nor data valid is a continuous event. There can be direct memory access or processor status activity on the data lines between address/data valid pulses. The 9010A ignores any such activity except for counting events, which is completely asynchronous.

From the foregoing discussion, how can we get "bad" signatures?52

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

51. Shift the register right 8 times to get rid of the count, then logically AND it with FFFF to get rid of the logic history.
52. We get bad signatures when we probe a point which has unsynchronized logic activity occurring during address/data valid. Also, if address/data valid does not occur between read probes, the signature will always be zero.

Probe Lights and Logic Level History

The probe body contains both a red and a green light which illuminate to notify you of activity at the point being probed. This is a simple yet powerful troubleshooting tool because it allows you to see free run or synchronized logic activity as it occurs. The condition of the lights reveal signal activity at the probe tip as shown in Table 4.

Table 4. Meaning of 9010A Probe Lights

PROBE LIGHTS	MEANING
Both off	Tristate (.8V to 2.4V, >100 ns), d/a sync only
Red	High (>2.4V, >75 ns)
Green	LOW (<.8V, >75 ns)
One steadily on	DC level
Both steadily on	Toggling between high and low (no tristate)
One flashing *	Toggling between respective level and tristate
Both flashing *	Toggling between high, low, and tristate

* Toggle rate is not related to flash rate

The probe detects high or low logic excursions in excess of 75 nanoseconds and in-between (tristate) excursions in excess of 100 ns. In free run the levels are detected when they occur. In data or address sync, however, only high and low levels are detected (not tristate), and then only during address or data valid. This allows you to use the probe as a pulse catcher.

To use the pulse catcher feature, press SYNC A to set up the sync mode, and READ-PROBE to clear the pulse catcher. Then probe a node and press READ ENTER. If the node were high or low during the read operation, the appropriate probe light will come on and stay on until the next read/write or read probe operation.

To use the probe in free run, press SYNC F, touch the probe to ground at U40 pin 8, and note the steady green: ground is logic low for more than 75 nanoseconds. Touch the probe to +5 Volts at U40 pin 16 and note the steady red. Now touch it to the TTL clock U40 pin 6 and note the steady red and green. This point is cycling high-to-low and is not entering the high-impedance state for more than 75 nanoseconds.

Press RESET on the TK-80A board. Probe address bus bit 2 at U26 pin 3. According to the probe lights what is the activity at the tip, what causes it, and how did you find out?53

Note this: the logic history is sensitive to the sync mode. Thus, in address or data sync, it is a history only of the logic levels which occurred during a read or write operation. Whereas, in free run, it is a history of all logic levels which occurred at the probe tip.
To demonstrate this, press SYNC F, probe U40 pin 8 and press READ-PROBE

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

53. The red and green lights are blinking. This means that the line is toggling between high, low, and tri-state. The 9010A is not doing any read/writes, but the display DMA circuit is. We can prove this by probing U26 pin 2 (toggling), and processor's address bus buffer U33 pin 2 (steady green).

twice. Do the same with the probe at U40 pin 16, and then with the probe touching nothing. Note how the display shows the level at those points. Now probe U26 pin 2 and press READ-PROBE twice. What level history does the display show?54

If your probe is not contacting a circuit, but a probe light is blinking, then the HIGH or LOW key is in the IN position. Press them until the probe lights are both off.

Select data sync by pressing SYNC D. Probe bit 2 of the data bus at U21 pin 3 and notice the probe lights are off. Press READ-PROBE twice and note from the count that there is logic activity at the node, but it isn't being detected by the probe lights because it occurs at a time other than data valid.

Now you can use the probe to find out whether TOGGL-ADDR actually toggles an address line. Probe address bit 2 at U26 pin 3. While watching the probe lights, press READ-PROBE TOGGL-ADDR 8FF8 ENTER 2 ENTER. What indication do you get that the bit toggled?55

In a similar fashion you can prove you are toggling a control line. For example, press TOGGL-DATA STS/CTL 0 ENTER 0 ENTER. Now hold the probe on pin 2 of U34. Press REPEAT and notice the probe lights change state.

Counting Events

The event count feature serves a singular purpose: it enables you to verify that a given number of falling-edge events occur between two successive read probe operations. By doing this, it allows you to test circuits which are asynchronous to the bus. And, it works the same in all sync modes.

Each time you press READ-PROBE, the 9010A clears register 0 and loads it with the new event count. The event counter is a 7-bit circular counter. Instead of counting 128, the count rolls over and starts back at zero. It does this continually, as long as events occur.

For example, if you want to verify that the on-board clock frequency is correct you can write a program which does two successive read probe operations and displays the count. The count will be the same (plus or minus a few counts) every time. This has to be done under program control. Although you can use this feature to verify a clock frequency is correct, you can also use it to count the precise number of pulses at a point.

For example, suppose you want to verify that the TK-80A RESET key works properly. Set the HIGH and LOW keys to the out position. Press SYNC F READ-PROBE and probe U30 pin 35, the RESET signal to the PIA. Now press TK-80A RESET six times, then press READ-PROBE again. What does the display show happened?56

One final item to note is that the count is incremented when the signal at

-------------------- ANSWERS TO QUESTIONS IN TEXT --------------------

54. Low and High.
55. The red light came on, then the green flashed on and off once, leaving the red on.
56. Six events occurred, using both high and low logic levels.

the probe tip falls below the high threshold of about 2.4 Volts. Therefore, if the signal is toggling between low and tristate (but never exceeding and falling below the 2.4-Volt threshold), the 9010A will not register any counts.

Gathering Signatures

A signature is an algorithmic compression (cyclic redundancy check) of a digital bit stream into a unique four digit hex number. A given sequence of logic states at the probe tip will render a signature unique to that sequence. The idea of signatures as well as event counting is to characterize a known-good board, then to compare the good values with actual ones from a board under test.

To take valid known-good signatures, you must sync on either address or data and repetitively exercise the portion of the circuit being tested with read/write-type operations. The idea is to do a read probe, exercise the circuit, do another read probe, then note the signature from either register 0 or the display. As with event counting, each read probe operation loads a signature and triggers the next.

It is important to note that only the data occurring during data/address valid are computed in the signature, even though data may also be on the node at other times.

As an exercise, probe data bus bit 2 at U21 pin 3 and press SYNC D READ-PROBE RAMP 8FFF ENTER READ-PROBE. This puts a known activity on the address and data busses. What signature did you get, can you expect to get the same one a second time, will the count be the same, and why?[57]

High and Low Stimulus: Signal Injection

As indicated by Table 3, you can use the HIGH and LOW buttons to inject 2-microsecond pulses into the circuit from the probe tip. In free run, the pulses have a 1-millisecond period (1kHz). In address or data sync, the probe issues a pulse in synchrony with each address/data valid. If you press the HIGH button IN, the probe will inject high pulses; if you push the LOW button IN, the probe will inject low pulses. If both are IN, it will inject high and low pulses alternately.

An important consideration is that the pulses will also be sensed if you perform a read probe operation. In that situation you should keep the HIGH and LOW keys in the out position so as not to generate any pulses.

As an example of how to use the signal injection capability, assume you think there is an open data line. So you decide to conduct a read at a non-existent address, then inject some pulses and see if they make it to the pod. Press SYNC D, then READ 9000 ENTER LOOP. Press the LOW key to the IN position.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

57. The signature is 96EC. It should be the same every time because identically the same activity occurs in sync with data valid. However, the count will be different each time because data is continuously being placed by the display DMA circuit.

FF is the value read FF when nothing is driving the bus because the lines are pulled up. To stimulate the bus we must drive one of the data lines low with the probe. What value will be read with the probe on data bit 4?58

Probe data bit 4 at U21 pin 16 and observe the display. It should agree with your answer. What probe light comes on and why?59

With the probe you can verify correct operation of the PIA (U30). Set up the PIA by pressing WRITE 100FB ENTER 92 ENTER, then WRITE 100FA ENTER 80 ENTER. This sets up the keyboard column select register for output and the row response register for input. Now we need only loop while reading address 100F8, the register which inputs from the keyboard.

Press READ 100FB ENTER LOOP. Now, probe the inputs to the register, U30 pins 4,3,2,1,40,39,38 and 37 which represent data bits 0 thru 7 respectively. What is the effect on the display and what causes it?60

HOW TO START PROGRAMMING

Introduction

The 9010A allows you to enter programs which are considerably more powerful than immediate mode operations. There is a variety of reasons for this: to combine a lot of small test steps into one large group so that testing personnel can conduct the tests faster; to prevent low-skilled operators from adversely affecting test integrity; to exercise circuits for taking a lot of signatures for guided fault isolation. Your biggest reasons would be to save time, money, and effort.

You will be delighted at the ease with which you can program the 9010A. Once you know how to use the 9010A in the immediate mode you have most of the knowledge necessary to develop a program.

A program is a sequence of 9010A keystrokes being performed automatically. Any built-in test or troubleshooting function may be included in programs, as well as the learn, READ PROBE, and arithmetic operations. In a program you cannot use SETUP, VIEW, CONT, HIGH, LOW, READ-TAPE, and WRITE-TAPE.

In addition, test sequencing keys help direct the flow of the programs and allow the construction of conditional and unconditional branches, step labeling and display of programmer-generated messages or prompts.

Program Versus Execute Mode

The 9010A has three major operating modes: Immediate, Program, and Execute.

------------------ ANSWERS TO QUESTIONS IN TEXT ------------------

58. EF
59. The light comes on because the probe is putting out a logic low during data valid, and is sensing it as such.
60. The value being read initially is FF, the result of all lines being pulled up. However, the FF changes each time a point is probed to an 8-bit pattern in which 7 bits are ones and one bit is a zero. This happens because the probe injects logic zeros into the respective bit of the PIA.

Immediate is the mode we have been using up to this topic. Operations are executed immediately when you enter them.

Program mode allows you to enter program steps, but does not execute them. In program mode, the PROGMING light flashes on the front panel. Execute mode executes the steps in a program and flashes the EXECUTING light.

To enter program mode press PROGM and enter a number from 0 to 99. To exit program mode and re-enter immediate mode press PROGM again. To enter execute mode, press EXEC and enter a program number. To re-enter immediate mode, either allow the program to terminate on its own, or execute a STOP step, or press STOP.

The 9010A memory can hold 10,192 bytes of program information contained in up to 100 programs. Program numbers do not have to be sequential.

Program Structure

The 9010A allows branching within a program, and conditional or unconditional jumps to labels. There may be up to 16 labels, 0-F, placed in the program in any order. The 9010A also allows branching to other programs. When a program is branched to, the 9010A returns control to the calling program when it is done, like a subroutine. You may nest subroutines up to 10 deep.

Once a program is created, you can change the program number only by opening a new program and keying it in entirely. For this reason, we recommend you plan your program needs carefully, considering the limit of 16 labels and inability to change program numbers.

An additional factor to consider is the concept of "spaghetti code." If you have a lot of programs on a tape, and they are assigned to program numbers in random fashion, then they may be hard to maintain because they appear to be convoluted. The same is true of the organization of steps within a program.

Therefore, in planning your system programs, it may be a good idea to draw up an organization chart as to what range of program numbers are assigned to what program activities. It is workable to employ a top-down, modular structuring to make it evident that program activity generally flows from the beginning to the end rather than is a spaghetti-like fashion.

For example, you may want to test the major circuits of the UUT, then use some guided fault isolation routines to troubleshoot those which fail. Your programs might be organized as shown in Table 5.

The main program is 0. The operator need only configure the test system, read the tape into memory, and press EXEC ENTER (program 0 assumed at power on) to run the main program. The main program prompts the operator to select a circuit to test, then executes that program as if it is a subroutine.

The main test programs (10, 20, 30, 40) are grouped with auxiliary programs (21 - 22) which are needed for modularity and to provide corresponding main programs with additional labels. Each main test calls a guided fault isolation routine (11, 12, 22, 32, 41) when requested by the operator after a failure occurs.

Table 5. Suggested Style of Program Structure

PROG	PURPOSE	COMMENT
0	Main program	Calls other major test programs.
1-5	Utility programs	Perform common actions needed by other programs, such as delays, etc.
6	Guided Fault Main Progm	Displays point to probe and checks for correct signature/count.
7-9	Circuit Exercise Progms	Exercise display, keyboard and tape circuits for fault isolation.
10	High-Level Tests	Test 1 - runs bus and memory tests.
11	Bus Fault Isolation	Helps you troubleshoot bus failures.
12	Memory Fault Isolation	Helps you troubleshoot memory failures.
20	Main Display Test	Test 2 - tests UUT display .
21	Display Test Subroutine	Computes actual UUT display codes.
22	Display Fault Isolation	Helps you troubleshoot display failures.
30	Main Keyboard Test	Test 3 - tests UUT keyboard.
31	Keyboard Test Subroutine	Computes which key was pushed down.
32	Keyboard Fault Isolation	Helps you troubleshoot keyboard circuit.
40	Tape Interface Test	Test 4 - tests tape interface circuits.
41	Tape Circuit Fault Isol.	Helps you troubleshoot tape circuit failures.
99	Tape Menu	Lists names of programs and uses.

Each fault isolation program sets up parameters in some registers to identify the point to be probed and expected results. It then calls the main fault isolation routine which breaks down the registers, prompts the operator, calls the appropriate circuit exercise routine, and compares expected with actual results. The circuit exercise programs perform read/-write-type actions to exercise the circuit being probed.

Each called (executed) program returns to the calling program when it is finished. Note that the programs which are called most frequently are assigned low program numbers to reduce the number of steps in a program.

Finally, it may be important to document the intended purpose and functioning of your programs. After the programs have been around a few years, your group may want to modify them after you have moved on. The documentation will assist them in doing that. Program 99 is an example of a way to document all your tapes at least enough to identify them if other information gets lost. Its only purpose is to list the names of the programs on the tape with a brief description.

Setup

Before executing a program you may need to change some of the 9010A SETUP conditions. For example, you may want to set TRAP ON ACTIVE FORCE LINE to NO because so the program won't stop when the operator pushes the UUT RESET key (as on the TK-80A). Or, you may want BUS TEST to test data lines at a display address so the operator can see the display digit toggling.

The important item here is that a program cannot change SETUP conditions. However, when you do a write tape operation, the SETUP conditions and address descriptors are stored on the tape along with the programs. This lets the operator load them with a read tape operation.

HOW TO WRITE A PROGRAM

Introduction

Let's create and delete a simple program. Press PROGM 50 ENTER. This opens program number 50 and puts the 9010A in the program mode. It also displays the message PROGRAM 50 CREATED. Now press BUS test. This creates a BUS TEST program step.

Press MORE to advance to the end of the program. Now press PRIOR to back up in the program. Press ~~MORE~~ PRIOR to display START OF PROGRAM 50. Now press PROGM to close the program. You have just created a one-step program which will conduct a bus test.

Now press EXEC 50 ENTER. It runs the bus test.

It is easy to delete a program. To demonstrate this, re-open program 50 by pressing PROGM ENTER. Notice that the 9010A assumes you wish to re-open the last program opened. Now press CLEAR and notice that the 9010A asks whether you are sure you want to delete it. This avoids inadvertent destruction of your work. Press YES. Notice how the 9010A also displays how many bytes remain in program memory.

There is an important thing to note about the use of registers in a program. Registers 8-F are global: when a program executes another, those registers remain intact. However, registers 0-7 are local: when a program executes another, they are automatically set to zero. When you execute a program from immediate mode, all registers remain intact. We will be using registers both locally and globally in writing programs.

Enter a Working Program

The program in Table 6 runs a bus test and writes a one (binary) to each segment bit of each of the eight display addresses. Its purpose is to test the display. Enter the program by pressing the as shown in the left column.

Edit the Program

The 9010A contains an invisible step pointer which causes a step to be displayed. You can change the pointer and thus display different steps in the program. You have already learned how to do this with the MORE and PRIOR keys. Notice how this moves the pointer forward and backward through the program.

You can also move the pointer by labels. Press keys 0-F one at a time. Which did not cause the 9010A to beep at you and why?61

It is easy to edit a program step. If you made a mistake on the entry or

Table 6. Sample Display Test Program

KEYSTROKES	EXPLANATION
BUS-TEST	Run bus test
WRITE 100FB ENTER 92 ENTER	Set up the TK-80A PIA
WRITE DECR ENTER 80 ENTER	Unblank the display
REG F 8FF8 ENTER	Set address register to address of digit 8
REG 8 8 ENTER	Initialize digit counter for eight digits
LABEL 1	Entry for digit loop
REG E 100 ENTER	-Setup data register for segment code
LABEL 2	-Entry for segment loop
SHIFT-RIGHT E	--Shift the segment code to the right
WRITE ENTER ENTER	--Send segment code to display address
IF REG E > 0 GOTO 2	--Loop until all segments are done
INCR F	-Increment display address
DECR 8	-Decrement digit counter
IF REG 8 > 0 GOTO 1	Loop until last digit is tested

want to delete a step, simply press CLEAR until the 9010A displays STEP DELETED. Try this on the BUS test step. To insert a step, simply display the step preceding the one you want to insert, then make the key entries for that step. Try this by re-entering the BUS test step.

Execute the Program

Press PROGM to close the program and display the number of bytes remaining. Press PROGM =. Notice how the 9010A displays the fact that its memory contains only program 50.

Now run the program by pressing EXEC 50 ENTER. What did the program do?[62] Why does the program use registers F and E?[63]

To rerun the test, press EXEC ENTER. Note that the 9010A assumes you want to re-execute the last program you executed. Run the program again by pressing RPEAT. Repetitively rerun the program by pressing LOOP. How can you tell that the program is executing and looping?[64] Press STOP. Note that both the STOPPED and EXECUTING lights are flashing. Now press CONT and notice that the program continues looping.

The purpose of the preceding exercise is to demonstrate that the 9010A treats a program like it does any built-in test or function. It handles errors in a similar way, too. A program step can halt the program to display an error message just as it would if the error occurred in

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

61. 0, 1, 2, E, and F. 0 moves the pointer to label 0 or the beginning of the program. 1, 2, and E move it to those labels. F moves it to label F or the end of the program. The others only beep because their labels do not exist.
62. It ran the bus test then lit each segment in turn of each digit in turn for all eight display digits from left to right. Then it stopped. The EXECUTE light flashed while running.
63. Since they are the default address and data registers, using them saves keystroke in the WRITE step because the F and E registers are assumed.
64. The EXECUTING light is flashing and occasionally the LOOPING light flashes. Also, the display ramping runs continually.

executing mode. Some errors permanently halt a program. An example is a step which attempts to execute a non-existent program.

To demonstrate error handling, short pins 15 and 17 on U34. Then execute the program by pressing EXEC 50 ENTER. When the error occurs LOOP on it and then remove the jumper. What error did you get?65 Now press CONT. Notice that the looping stops and the remainder of the program executes from the point at which the error occurred.

Enter a Display Message

You have probably noticed that the 9010A display gives you no indication of what is being tested. Let's assume you want to keep the operator from getting bored while the 9010A is testing the UUT. To do this, you need to display a message.

In order to display a message, you will need to press the DISPL key. When you do, the 9010A instantly redefines most of the keyboard so that when you push a key, it puts a single character on the display. The characters are printed near the lower right corner of each key, except for the hex keypad which is used as is. If you press the wrong key, press PRIOR to delete unwanted characters. To enter the display message, press ENTER.

Some of the characters sent to the display cause special activities to happen. With them you can display a register's contents in decimal or hexadecimal, let the operator enter a value into a register, toggle the enabling of a keyboard interrupt to the program, or beep. For the most part, those same characters are used for special purposes in sending data to an external device via the RS-232 port (AUX-I/F). Table 7 shows the special characters and their uses.

Table 7. Special Characters for the Display and Aux I/F

CHAR	WHAT IT DOES TO THE DISPLAY	WHAT IT DOES TO AUX I/F
bell *	Beep	Send CTRL G (hex 7)
$x	Dsp hex contents of reg x	Send hex contents of reg x
@x	Dsp decimal contents of reg x	Send decimal contents of reg x
/x	Wait for next keyboard entry; put hex entry in reg x	Wait for next byte from i/f; put hex i/f byte in reg x
\x	Wait for next keyboard entry; put decimal entry in reg x	Wait for next byte from i/f; put decimal i/f byte in reg x
?x	Wait for YES/No key input; put 1 (YES) or 0(NO) in reg x	Not used
%x	Toggle kbd interrupt enable; put code for next key pressed in reg x if interrupt enabled	Send low byte of reg x
+	If first character, add rest of line to current display.	If last character, do not send newline terminator-see setup

* Bell is the ROM key.

For a display example, open program 50 and enter the steps from Table 8

------------------ ANSWERS TO QUESTIONS IN TEXT ------------------

65. DATA BITS 0 AND 1 TIED.

immediately after the SHR REGE step. The characters "sp" in the display step mean space (STS/CTL key)). What will these steps do, why is the second step needed, and what is the purpose of the + symbol?66

Table 8. Display Message Keystrokes

KEYSTROKES	COMMENT
DISPL TESTINGspDIGITsp@8spSEGMENTsp$E	Displays a message
DISPL +b	Note: b=bell (view ROM key)

How to Send Data to the Auxiliary Interface

You can send messages to an external device by pressing AUX-I/F rather than DISPL, then entering the message. The 9010A sends the newline terminator at the end of each AUX-I/F operation unless you specify otherwise.

You can change newline using SETUP. Press SETUP, then press MORE until you see the NEWLINE display of eight hex characters. The left two represent the number of 2.4 millisecond periods to delay between lines. The right six are the codes for three ASCII characters to be sent at the end of each line.

Press PRIOR twice. The STALL and UNSTALL characters are sent to hold-off and restart transmission of characters. The 9010A responds to these but never has to send them because it is faster than the highest data communication rate.

If you press MORE three times you will see the LINESIZE message. This is used only when you send programs or dump memory to an external device. It limits the maximum length of a line. You may set it to any value between 9 and 256. It does not affect data sent during program execution.

The 9010A treats certain characters in an AUX-I/F step in a special way, similar to a DISPLAY step. Table 7 lists them. We will not do an AUX-I/F exercise because the the 9010A is not connected to an RS-232C device.

Create and Execute A Subroutine

You may not be satisfied with the program we have created because it tests the segments too fast. It would be nice to slow it down. Then you would have a test your operator could really use. So, let's add another program to create a delay, and execute it as a subroutine.

Reopen program 50. Insert the steps in Table 9 after WRITE @ REGF = REGE. Close and execute the program. The display shows what happens when you try to execute a non-existent program.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

66. The first step displays TESTING DIGIT x SEGMENT y, where x is the decimal contents of register 8, and y is the hexadecimal contents of register E. The second step adds a beep to the display. It is needed because the display is too full to enter it with the first step. The + symbol keeps the second display step from deleting what the first step displayed. Together, the steps will beep and display the digit and segment number each time they are sent to the UUT.

Table 9. Keystrokes to Call a Subroutine

KEYSTROKES	COMMENT
REG 9 10 ENTER	Sets up register 9 with a delay count
EXEC 51 ENTER	Executes program 51 as a subroutine

Now create the subroutine which will act as a delay. We will make it a separate program rather than to integrate it into program 50. The reason is to allow other programs to use it for needed delays. Open program 51 and insert the steps in Table 10. Why do we use register 9 for the delay counter?67 Close the program and execute program 50.

Table 10. Keystrokes to Enter the Delay Subroutine

KEYSTROKES	COMMENT
LABEL 1	Delay loop entry
IF REG 9 = 0 GOTO 2	Loop until delay count equals 0
DECR 9	Decrement the delay counter
LABEL 2	End loop entry

Request Entries from the Operator

So far, our program tests the display all right, but it doesn't allow the operator any prerogative over how the test is to be conducted. Most operators, once they get familiar with how to conduct a test, like to shortcut procedures which are necessary only for less experienced operators. Let's give our program these capabilities:

1. The test name is displayed.
2. The operator can elect to run RAM test or not.
3. The operator can enter the delay parameter.
4. The operator can terminate the test at any time by pressing a key.

For all of these changes to the program, open the program, advance to the indicated step in parentheses, and delete the old steps and/or insert the new ones given in Table 11.

The 9010A has assigned a code to each key except HIGH and LOW. When you press one of the keys, its key code is automatically entered into the B register, provided the keyboard interrupt is enabled. All of the key codes are less than 40.

In the Table 11, the items marked with the asterisk enable the operator to terminate the test by pressing any key. First we set the B register to 40 and enable the keyboard interrupt with the display %B step. Therefore, pressing a key changes the contents of the B register so that the IF 40 > REGB step will later cause a branch to label F and end the program.

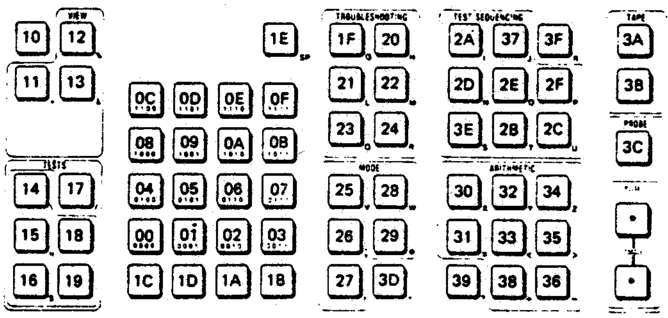
Figure 4 shows the codes for the keys (Note - STOP ends the program).

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

67. Because it is a global register. It has to be global in order to put an initial value in it in program 50, then retain the value upon executing program 51.

Table 11. Keystrokes to Request Operator Entry

KEYSTROKES	COMMENT
(Start of program 50)	<Display this step>
DISPL DISPLAYspTESTsp-spRAMspTOOb?2 <ENTER>	Beep and ask for entry
DISPL ENTERspDELAYspCOUNTsp/3 <ENTER>	Ask for delay count
REG B 40 ENTER	* Set B reg to 40
DISPL +%B <ENTER>	* Enable key interrupt for reg B
(Bus test)	<Display this step>
IF REG 2 = 0 GOTO 0	Skip next step if no entered
RAM-LONG 8FF8 ENTER 8FFF ENTER	Test display RAM
LABEL 0	Entry if RAM test bypassed
(Label 2)	<Display this step>
IF 40 > REG B GOTO F	* End if any key is pressed
(Reg9 = 10)	<Display and CLEAR this step>
REG 9 = REG 3	Set delay count = delay entered
(End of program 50)	<Display this step>
LABEL F	End-of-program entry
DISPL ENDspOFspDISPLAYspTESTb <ENTER>	Display the end message and beep



00-ZERO	08-EIGHT	10-LEARN	18-RAMST	20-WRITE	28-STOP	30-AND	38-REG
01-ONE	09-NINE	11-VRAM	19-IOTST	21-RAMP	29-R UUT	31-OR	39-RDPROBE
02-TWO	0A-A	12-VIO	1A-PRIOR	22-WALK	2A-PROGM	32-SHLFT	3A-RTAPE
03-THREE	0B-B	13-VROM	1B-MORE	23-TADDR	2B-LABEL	33-SHRT	3B-WTAPE
04-FOUR	0C-C	14-AUTOT	1C-ENTER	24-TDATA	2C-GOTO	34-INCR	3C-SYNC
05-FIVE	0D-D	15-BUST	1D-CLEAR	25-CONT	2D-IF	35-DECR	3D-SETUP
06-SIX	0E-E	16-ROMT	1E-S/CTL	26-RPEAT	2E- >	36-COMPL	3E-DISPL
07-SEVEN	0F-F	17-RAMLT	1F-READ	27-LOOP	2F- =	37-EXEC	3F-AUXIF

Figure 4. Asynchronous Keyboard Interrupt Keycodes

Table 12 gives the steps for a short program to display the code for any valid key you push. Enter the program and execute it to see how it works.

Program a Read Probe Operation

Now you should be ready to write a troubleshooting program. Assume you are concerned about the asynchronous circuits on the UUT. If you were working on the TK-80A board, you might want to see whether the clock, tape, and display circuits are working properly during a troubleshooting operation.

What you might do is write a program which tells the operator where to place the probe, then compares the actual to the expected signature/count. However, we do not need such detail to demonstrate how effectively the 9010A can guide the operator. Therfore, our program will be rather simple.

Table 12. Program to Read Keycodes

DISPLAYED STEP	COMMENT
DPY-PRESS A KEYb	Initial message with beep
LABEL 1	Program loop entry
REGB = 40	Initialize interrupt register
DPY-+%B	Enable keyboard interrupt
LABEL 2	Wait loop entry
IF REGB = 40 GOTO 2	Wait for key to be pressed
DPY-KEYCODE = $B	Display the code
GOTO 1	Loop

Because the 9010A continually communicates with and monitors the status of the pod, it seems to run programs slowly compared to a microcomputer. However, its internal timing is regular enough to use a program to check the frequency of the UUT clock with event counts. To demonstrate this, open, key in, and close program 53 using the keystrokes in Table 13. Why give the operator a choice of sync?68

Table 13. Read Probe Program

KEYCODES	COMMENT
SYNC D	Initialize sync on data
DISPL SYNC ON FREE-RUN?1 <ENTER>	Ask operator for choice of sync
IF REG 1 = 0 GOTO 1	Bypass next step if data/address sync
SYNC F	Sync on free-run
LABEL 1	Read probe loop entry
READ-PROBE	Clear reg 0 and initiate read probe
READ-PROBE	Take good reading
IF REG 1 = 1 GOTO 2	Bypass bus test if free run
BUS-TEST	Exercise data and address lines
READ-PROBE	Read probe for bus test
LABEL 2	Entry for bypassing bus test
REG A = REG 0 AND 7F ENTER	Mask out all but the count
REG 0 = SHIFT-RIGHT(8 times) AND FFFF ENTER	Mask out all but the sig
DISPL SIGsp=sp$0;spCOUNTsp=sp@A <ENTER>	Display the sig and count
GOTO 1	Loop

Execute the program, select Free-run, and probe the following points and note the results: U40 pins 6 and 7 (system clock); U7 pin 9 (tape circuit clock); U26 pins 1, 2, 4, and 6 (display DMA address generator). Why is

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

68. Because if only free-run is selected, the signature will always be zero. If only data is selected, the signature will not be stable.

this a useful way to check those asynchronous circuits?69

Let's prove your answer. Look at the tape circuit on the schematic. Jumper U27 pins 7 and 8 together. This grounds the gate which loads counter U6, thus simulating a fault.

Now probe U7 pin 9 again and notice the different count (0). This proves there is a fault somewhere in the clock or tape circuit.

Now trace backward in the circuit through U51, U28, U29, U6 and U27 until you have found the fault with your program and the probe. How might you tell from the probe alone where this particular fault is?70

Why does this program use three read-probe operations?71

Remove the jumper and stop the program by pressing STOP. Using SETUP, set the BUS TEST address to 8FFC. Execute the program again and select NO free-run sync. Referring to the schematic, probe some address and data lines on both sides of the processor's buffers and notice the 9010A display. The bus test runs identically every time through, but the counts displayed are quite unstable. However, the signatures are stable. Why is this?72

To prove that you can find a fault in circuits synchronous to the bus, jumper U38 pins 7 and 8 together. This grounds the control line to the address decode circuitry (top of the schematic). Notice how it lights up the entire TK-80A display.

Seeing this fault, you might want to know why spurious data is being displayed. Perhaps it is coming from addresses which aren't supposed to be selected during the display DMA cycle.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

69. Because even though the count cannot be directly converted into frequency, it is stable within one to three counts. If there were an error in the circuit, it would be obvious because the count would be off more than that amount. Further, tracing backwards from the first incorrect count could easily isolate a failure in the asynchronous circuit to one component.
70. The probe lights indicate that the nodes are all stuck low or high, and that the outputs are correct for the inputs...until you get back to the culprit, U27. It shows a good clock entering U6 pin 2 because both lights are on. But it shows that U27 pin 8 (the output) is the same level as pin 9 (the input). This is incorrect for a NAND gate, so U27 is bad.
71. The program does two read probes in a row without sending out any addresses or data. The purpose is to minimize the 9010A delay between them so the period will be repeatable. This will give a relatively stable count for free-run sync. The third read probe is done only for data sync operations (the first read probe is wasted for these). The purpose is to execute a bus test before doing the last read probe. The bus test generates address valid and thus opens the signature window for valid signatures.
72. The 8080 puts data on the data lines during non-read/write times to notify any interested external devices of its status. Also, the display DMA circuit puts addresses on the address lines and data on the data lines ONLY during 8080 non-read/write times. Thus there are many events occurring which are not in sync with address/data valid from the processor. This affects the count because ALL events are counted. However, it does not affect the signature because the signature window is open only during read/write times (address/data valid).

So you decide to run a set of built-in tests to see what's going on. Press BUS TEST. It passes. Assuming RAM information from a previous learn is still intact (8C00-8FFF), press RAM-SHORT ENTER. It fails, and when you press CONT several times, there is no change. In fact, you set SETUP's EXERCISE ERRORS to NO and run RAM SHORT again. The 9010A beeps continuously to say that every address is failing to read/write all bits. You set EXERCISE ERRORS back to YES and run ROM TEST. It passes!

All bits and all addresses, RAM only. That means no read/write activity for RAM. So you decide to probe around in the chip select (address decode) circuit. Execute program 53, select NO free-run sync. Probe chip select signals at pin 8 of the RAM chips U16 and U17.

You get no count, which means no DMA accesses. You get a signature. What causes that?[73] Probe backwards in the chip select circuit until you locate the fault. Now remove the jumper and probe through the circuit again to see how it looks when it works properly. While the fault was in, why did the TK-80A display light up?[74]

HOW TO SAVE AND RETRIEVE PROGRAMS

There are two ways you can save programs. The first is on the 9010A cassette tape. The second is on another 9010A or a computer connected to the RS-232 interface.

Tape

To save your programs, setup information, and address descriptors, insert a blank cassette (not the demo tape!) and press WRITE-TAPE YES. The 9010A will dump its entire read/write memory to the tape. To load the same information from tape into memory, press READ-TAPE YES. Notice how the 9010A reminds you to be sure you are not wiping out your programs.

RS-232

There are seven ways you can transfer data via the auxiliary interface in immediate mode. Each requires that you press the AUX-I/F key first. They are given in Table 14.

The 9010A Operator and Programmer manuals describe RS-232 transfers in detail, as well as the formats. Because we provide no RS-232 compatible device in the evaluation kit, it is not practical to demonstrate these features in this manual. Contact your Fluke representative if you have related questions.

HOW TO USE THE 9010A DEMONSTRATION TAPE

Introduction

This topic describes the 9010A demonstration tape which is part of the kit. We have included this because it gives an example of an acceptable approach

Table 14. AUX I/F Operations

KEY	OPERATION
=	Sends the program numbers and sizes in ASCII
PROGM	Sends listing of all programs' contents in ASCII
dd ENTER	Sends listing of program dd (decimal) in ASCII
LEARN	Sends address space descriptors in ASCII
SETUP	Sends setup information in ASCII
WRITE	Sends entire memory contents in ASCII hexadecimal format
READ	Receives programs and info in ASCII hexadecimal format

to testing and troubleshooting a micro-system. We wrote it for the TK-80A to be a demonstration tape, not a comprehensive test of the board. So, even though it does not test the entire board, it does offer a good approach to structuring programs, and grouping them together.

Appendix A lists the setup information, address descriptors, and programs. They appear in a format similar to what you will get if you perform AUX-I/F PROGRAM.

To load the tape into memory, slide the cassette into the drive, shut the door, and press READ-TAPE YES. If the READ TAPE FAIL display appears, perform the whole operation anew. READ TAPE OK will appear when the operation is successful.

Program 0 calls other programs as if they are subroutines. You may also execute those same programs from immediate mode. This double feature enables you to modularize programs, thus allowing them to be written, executed, and updated easily.

You may not want to continue running a test once started. To illustrate the asynchronous keyboard interrupt feature of the 9010A, the tests are constructed to allow you to exit back to the main program (0) if you press the CLEAR key.

Run the Display Test

The display test tests the display by illuminating segments in a pattern. First it turns all the segments on, then off, then lights one segment at a time for a digit, then displays the numbers 0-9, and leaves the digit blank. You must watch the display to ensure it works properly. If the test fails, you can isolate the fault with another test.

Press EXECUTE 0 ENTER. You will receive your first prompt, asking whether you wish to run the display test. Press YES. During the display test, the 9010A is deciding what to write to that portion of memory and actually doing so.

------------------- ANSWERS TO QUESTIONS IN TEXT -------------------

73. It is the signature for +5 Volts. To prove it, probe +5 Volts at pin 18 of the RAM chips. You get a signature for +5 Volts because the signal is high during every signature window.
74. Because all the data lines are pulled high when no memory accesses are being made.

This is an easy circuit for the 9010A to test. It can put predetermined codes in those addresses, then wait for the DMA circuit to read and display them. It demonstrates the simplicity and ease of the absolute control the 9010A has over the UUT memory and its relative control over any circuit which performs DMA operations with it.

Press STOP and CONT several times and verify that the TK-80A is illuminating the correct segment. While the test is running, connect U30 pin 10 to pin 7 (ground), the unblank line from the PIA. This will cause the display to go blank.

Terminate the display test before it reaches the end by pressing CLEAR. The 9010A will display END OF DISPLAY TEST and then DID DISPLAY TEST PASS? Press NO in answer to this question.

The 9010A will display WANT TO TROUBLESHOOT DISPLAY? Press YES in answer to this question.

Perform Guided Fault Isolation on the Display Circuit

This program instructs you where to place the probe, exercises the display circuit, and does a read probe. It then compares the logic history, signature, or event count against prestored limits and notifies you of a failure. You can choose to loop on any failure. If you do not loop on a failure, the 9010 will tell you which component (chip) to replace. If the program does not find a bad component, it assumes the trouble is in the display driver transistors or the display modules themselves and prompts you to repair that circuit.

The program tests the display circuit which we know to be asynchronous, and thus a good test of the ability of the 9010A to ferret out a failing component. In doing so, it tests for correct logic history, signatures, and event counts. The program first checks the inputs to the circuit. These are asynchronous clock and counter outputs (asynchronous to the bus). Since we cannot get repeatable signatures on these signals, we merely look for logic activity and know from this that the related gates are passing signals. Then, the program looks for signatures at the places where they can be found. Finally, it checks for correct event counts at the display segment decoder.

Place the probe tip exactly where the display tells you to, referring to the TK-80A component locator as necessary. Ensure you are on the correct pin and IC. Ensure the point of the probe is firmly touching the pin.

The test will check for the correct logic history at the point being probed. Then, it will compare actual to expected results and display them both as follows:

    U28-11 LEVEL HL= HL, GOOD

Continue probing the circuit as instructed. Note that the test checks for correct logic history, event counts, and signatures at the various test points. When you probe U53 pin 2 and press CONT, the 9010 will display:

    U53-2 SIG FCE6= 0, BAD;LOOP?

This display means: "The signature at U53 pin 2 should be FCE6, but is actually zero; do you want to loop on the failure?".

Press YES to loop. You would typically look in the probed area for evidence of the cause. Disconnect then reconnect the jumper to simulate an intermittent fault and note the results. Press CONT to stop the looping (CLEAR you will abort the test; MORE will proceed to the next test point). The test will display the failure and ask if you wish to loop.

Press NO. The display will instruct you to replace the suspected bad chip U30. Remove the ground from U30 pin 10. The simplicity of this type of troubleshooting makes it easy for unskilled operators to find faults using the 9010A.

The display has also asked whether you wish to continue. Since you are going to replace the bad chip, press NO. The guided fault isolation test will abort. The display will ask whether you wish to test the Keyboard circuit. Press YES.

Run the Keyboard Test

The 9010A will display WAIT. At this point it is loading display codes into the TK-80A memory. Since the memory was tested prior to running the program, it makes sense to use the UUT memory to assist in testing other circuits. After about 10 seconds, the 9010A and TK-80A displays will instruct you to push a key.

With the exception of RESET, press and hold several of the TK-80A keys, one at a time. Notice that both displays show which key you are pressing, but only while it is held down. The 9010A continuously reads the PIA to determine which row of keys is held down, then isolates the particular key in the row.

Press and hold RESET for a few seconds then release it. This will cause the TK-80A to send the RESET signal to the pod. Recall that RESET is one of the forcing lines which was disabled in SETUP information read from the tape. The 9010A has the capability of trapping on lines such as these or not. They would reset the 8080 processor on the board if it were plugged in, and make it difficult to troubleshoot the RESET circuitry.

DEMO TAPE CONTENTS

Introduction

This section describes the contents of the demo tape. Appendix A lists the contents as they would be printed on a device connected to the RS232C port if you were to press AUX-I/F SETUP, AUX-I/F LEARN, and AUX-I/F PROGM.

Setup

The setup information loaded from the tape is different from the power-on default setup information. TRAP ON ACTIVE FORCE LINE is set to NO so the 9010A won't respond to RESET and interrupt the test when you press the TK-80A RESET key.

Also, TRAP ACTIVE INTERRUPT is set to NO so that the S-A toggle switch on the TK-80A board will not abort BUS TEST by generating an interrupt in in the S position. The BUS TEST data address is set to 8FFF so that you can see the right-most TK-80A digit change when you run BUS TEST.

Address Descriptors

The address descriptors loaded from the tape are not the power-on values of zero. They are the actual descriptors derived from a LEARN operation on the TK-80A board. Except for the I/O area, they are the real addresses present on the board. The descriptors are:

ROM = 0000-07FF
RAM = 8C00-8FFF
I/O = NO INFORMATION

The actual I/O addresses on the board are 100F8 - 100FB. The section on the TK-80A board describes the use of these addresses.

Programs

A 9010A tape may contain up to 10,192 bytes of program data, approximately one byte per keystroke. The demo tape uses about 4600 bytes. The tape contains the programs listed in Table 15.

Table 15. Programs on the Demo Tape

PROG	NAME/PURPOSE
00	Main program
03	Wait for asynchronous keyboard interrupt
04	Enable asynchronous keyboard interrupt
05	Output eight bytes to TK-80A memory at address F+1
06	Delay loop n
07	Delay loop 4n
08	Delay loop 8n
09	Delay loop 16n
11	Output four bytes to TK-80A display memory
20	TK-80A display circuit guided fault isolation
21	GFI test's routine: display prompts and report errors
22	Routine for user to derive known-good signatures/counts
23	Routine to select sync mode and accept reading
24	Routine to ensure probe is attached or removed
25	Routine to display actual/expected logic history
26	Routine to display expected signature, count, or history
27	Routine to exercise circuit under test
30	TK-80A keyboard test
31	Keyboard test's routine: display which key was pressed
32	TK-80A display test
33	Display test's routine: display message & write address

You may run any of the programs by pressing EXEC, the program number, then ENTER. However, it is pointless to run some of the programs by themselves as they are intended to be executed as subroutines by the more high-level programs.

Main Program, Program 0

Program 0 displays the tape version and asks whether you wish to run a particular program. If you press the YES key, it runs the program. If you press NO, it asks for the next. It will run programs 32, 20, and 30: the display, fault isolation, and keyboard tests, respectively.

It runs the display fault isolation test only if you indicate that the display test failed. At the end of the last test it displays END OF TEST and stops.

Asynchronous Keyboard Interrupt (Programs 3 and 4)

Program 4 enables the Asynchronous keyboard interrupt to place the value of the 9010A key you press into the B register after setting it to 40. Since the maximum key value is 3F, the register will contain 40 only until the next key is pressed or the interrupt disabled by another EXECUTE PROGRAM 4 operation, whichever occurs first.

Program 3 executes program 4 to enable the interrupt, then waits until the B register value changes (a key is hit) before returning to its calling program (the program which executed it).

Programs 3 and 4 are the programs which enable the asynchronous keyboard interrupts for the demo. They are called (executed) extensively by the other programs.

Output Bytes to TK-80A Memory (Programs 5 and 11)

Program 5 sends eight bytes to eight consecutive TK-80A memory addresses. It starts by assuming that register F (the address register) is one less than the starting address, and that registers 8 and 9 contain four bytes each in the following order:

Reg 8 = bytes 1,2,3,4;   Reg 9 = bytes 5,6,7,8

It then breaks down the contents of registers 8 and 9, and sends out bytes 1-8 one byte at a time to the default address (register F), incrementing register F just prior to sending each byte. It is up to the calling program to setup registers 8, 9, and F prior to calling program 5.

Program 11 causes the coded contents of registers 8 and 9 to be displayed on the TK-80A seven-segment display. The calling program must have previously set up registers 8 and 9 with TK-80A display codes. Program 11 sets up register F to 8FF7, one less than the lowest display address, then calls program 5. The TK-80A uses Direct Memory Access (DMA) to read the contents of RAM addresses 8FF8-8FFF between normal 8080 read/write memory accesses. It displays the coded values from those addresses on its 7-segment display. The left-most digit corresponds to address 8FF8.

Delays (Programs 6 - 9)

Program 6 is a loop which delays the calling 9010A program before continuing, usually to allow a dynamic display to remain long enough for you to read it. Programs 7-9 increase the delay to 4x, 8x, and 16x, respectively: 7 calls 6, 8 calls 7, and 9 calls 8. Other demo programs use the delays extensively.

TK-80A Display Test (Programs 32 and 33)

Program 32 exercises the TK-80 display and automatically returns to the main program when done, or when you press CLEAR. The test starts with the right-most display position and tests each position right-to-left. First it displays each segment in turn for the least- to the most-significant bit. Then it displays the digits 0-9 and leaves that character blank. It ends the test by displaying END on the TK-80 Display. It calls program 33 to display which digit is currently being tested.

Guided Fault Isolation Test (Programs 20-23)

Program 20 guides you through troubleshooting part of the TK-80A display control circuit. It checks the outermost inputs to the circuit first, then checks internal signals.

Program 20 sets up the point to probe and the expected logic history, count, or signature in registers 8 and 9, then executes program 21 to take the signature. Program 21 breaks down registers 8 and 9 to display information to you as the test is running. Registers 8 and 9 must have the format given in Table 16.

Table 16. Register 8 and 9 Format for the GFI Program

```
Register 8, bytes 4,3,2,1:

Byte 4,3 = Expected signature, or
     4   = minimum expected count and 3 = maximum expected count, or
     4   = don't care, 3 = OLXH, expected logic history
     2   = Chip (U) number to probe
     1   = Pin number to probe

Register 9, bytes 4,3,2,1:

Byte 4   = Third-most-likely U number to replace if a fault exists.
     3   = Second-most-likely U number to replace
     2   = First-most-likely U number to replace
     1   = Selected mode; bits = xxxc xfhe, where f selects the sync
           mode, and h and e select the meaning of bytes 4 and 3 in
           register 8

           c = 1 to have operator press CONT when probe is in place
           f = 1 for free run; 0 for address sync
           h = 1 to probe for logic history
           e = 1 to probe for event counts
           h and e = 0 to probe for signature
```

If you press CLEAR at any time other than when the STOP light is on, program 21 returns to program 20 which will end the test. If you press MORE, it returns to program 21 which will then set up registers 8 and 9 for the next point to probe.

Program 21 instructs you to probe a point in the TK-80A display circuit, and calls program 26 to display the expected value. It then calls program 24 to ensure you put the probe in place. If you press CONT during the probe message (program 24), the test will assume no response at the probed point and return to program 21 to display the BAD message. Otherwise, program 24 performs free run read probes until it gets a history other than X (tristate) for 10 operations, at which time it returns to program 21.

At this point, program 21 calls program 23 which performs a READ PROBE, exercises the circuit being tested if the parameters in registers 8 and 9 enabled it to, performs another READ PROBE, and returns to program 21.

Next, program 21 compares the signature and count with those expected. Based on the result, the program and operator interact as follows:

GOOD HISTORY/SIGNATURE/COUNT: Displays GOOD message then returns to program 20 to get next point to probe.

BAD HISTORY/SIGNATURE/COUNT: Displays the values expected and received, and asks you if you want it to loop on the failure.

NO LOOP: If you press NO, it displays a message telling you to replace the suspected bad component and asks if you wish to continue.

CONTINUE: If you then press YES, it returns to program 20 to probe the next test point.

NO CONTINUE: If you press NO, program 21 returns to program 20 which terminates the fault isolation routine .

LOOP: If you press YES to loop, the program loops on the current signature/event test, displaying the test results and loop message until you press CONT, MORE, or CLEAR.

CONTINUE: If the test passes, the program displays the GOOD message and returns to program 20 to get the next point to probe. If it fails, it repeats the BAD;LOOP message.

MORE: The test progresses to the next point to probe.

CLEAR: If you press CLEAR during the loop, the fault isolation test terminates.

Program 22 acts the way program 21 does in that it performs the identical testing operations except it does so under operator control. It allows you to press 0 for taking signatures, 1 or 5 for logic history (level), and 2 or 6 for event counts. 0, 1, and 2 perform write operations in address sync mode, while 5 and 6 perform no write operations and use free run sync. To change modes, simply press the appropriate key.

Program 22 executes program 23 which exercises the keyboard circuitry the same as for program 21. However, it does not look for errors. It only displays the signature, count, or logic history received. Thus, program 22 is useful for deriving signatures from a known-good TK-80A board for comparison with a faulty board. In fact, we used program 22 to determine what values to load into registers 8 and 9 for program 20.

TK-80A Keyboard Test (Programs 30 and 31)

Program 30 is an example of how to use the UUT memory, once it is known to be good, to assist in running a test program from the 9010A. When the program first starts, it displays a wait message. During the wait time it is downloading TK-80A display messages into the TK-80A memory, four bytes per message, and one message for each TK-80A key.

Later, each time you press a key, program 30 creates an index for that particular key. Using that index, it reads 4 bytes from TK-80A memory, then writes those bytes to addresses 8FFC-8FFF, the display addresses for the right 4 digits. In this way, the 9010 lets the TK-80A display which key you pressed.

Program 30 enables the key matrix via the 8255 chip, then monitors continuously for a depressed key. When you press a TK-80A key, the 9010 and the TK-80A both display which key you pressed as long as you hold it down. If something is not working in the circuit, then you will be able to tell it with this test because either the display will be blanked or the key number will not be displayed.

Program 30 executes program 31 in order to display which key was pressed.

The test is terminated and control passed back to the main monitor when you press the the 9010A CLEAR key or the TK-80A RESET key. The 9010A uses the READ STATUS function to check the ability of the 8224 chip to generate a RESET signal to the 8080 processor. To keep it from stopping the test with an error message, the demo tape contains the NO TRAP ON ACTIVE FORCING LINE setup information.

HOW TO GET STARTED WITH YOUR OWN UUT

Introduction

By now you are probably wondering whether the 9010A will really work on your UUT. It will if the microprocessor in your UUT is the same as the one in the pod included with the evaluation kit. If your UUT uses a different processor, your Fluke representative may be able to obtain the proper pod.

In this section we describe how to approach the configuring of the test system, what to do if things don't seem to go well, and some tips on getting the 9010A in use to reduce your test and troubleshooting times.

Essentially, we will follow steps similar to those for the TK-80A.

System Configuration

Before connecting the 9010A to your UUT, make the UUT accessible. This may mean putting one or more boards on extender cards or cables. Most important, observe the following safety precaution:

WARNING!!!

DISCONNECT ALL PERIPHERAL DEVICES WHICH OPERATE OR CONTROL SWITCHING, MACHINERY OR POWER SOURCES BEFORE CONNECTING THE 9010A TO THE UUT. THE REASON IS THAT THE 9010A COULD EXERCISE OR ENERGIZE THOSE DEVICES AND THEREBY DAMAGE EQUIPMENT OR INJURE SOMEONE.

Also before plugging the pod into the UUT, run a bus test on the pod with the flat cable plugged into the pod socket. Once that is done (OK message on the 9010A), plug the pod flat cable into the UUT's microprocessor socket so that the diagonal corner of the plug is aligned with pin 1 of the socket. Observe the following safety precaution:

CAUTION!

PLUG THE POD INTO THE 9010A ONLY WHEN 9010A POWER IS OFF AND INTO YOUR UUT ONLY WHEN 9010A POWER IS ON AND UUT POWER IS OFF. FAILURE TO DO THIS MAY DAMAGE THE POD OR 9010A.

What to do about System Errors

At almost any time you can get a system error message on the 9010A display. The system errors signify high-level, catastrophic malfunctioning of the interconnections, 9010A setup, UUT, or test equipment, in that order of likelihood. The following paragraphs discuss their causes (and implicit cures).

Active Forcing Lines (RESET, not-READY, HOLD, INTERRUPT)

These are lines which normally force the microprocessor into a specific state. Although their name and number depend on which pod you have, typically they are as given above. You can use the SETUP conditions to disable these lines from forcing the processor. Whether it is necessary is totally dependent on UUT design. It is important to note that you cannot disable them in RUN UUT mode because the processor in the pod relies on them for proper operation.

A good general approach to finding whether any are holding up your testing operation is to run BUS TEST. If you get a forcing line message, disable it in SETUP or provide the signals needed. do this with other lines as needed until the test passes. For example, the UUT may have a circuit which forces RESET or removes READY if the processor does not address it periodically. Or, removing the main board from the system to test it may force the HOLD line. In these cases you might be able to disable READY or FORCING LINES in setup, or ground the HOLD line.

A forcing line could also represent a fault in the UUT. Therefore, don't just disable them indiscriminately.

Pod Timeout

Pod timeout means the pod did not communicate to the 9010A within the SETUP time limit. Reasons for this are: the HOLD signal (or one of the other forcing lines) is keeping the processor in the pod quiet; the UUT clock is require you to send them setup commands before they are testable.

Other Testing

The philosophy of testing is to start with the most critical items. Then test the circuits nearest the processor and work outward from there. This guarantees that the most deadly problems will be found first.

The first items to test if you have doubts are power and clock. The probe lights tell whether power is available, while the read probe routine will verify the accuracy of the clock.

Run the other built-in tests, expecially AUTO (BUS, ROM, RAM SHORT, and I/O). These will thoroughly test any straight forward memory or I/O. To test PIAs and bank-switched memory, you will need to alternately send them some setup commands and test the setup portion.

Next, test the other bus-oriented circuits, such as directly driven I/O devices. Examples are keyboards, displays and digital-to analog-converters.

After the bus-oriented circuits, test the asynchronous circuits. These include DMA and other circuits which are not synchronized to processor bus activities. Start with the easy ones and work toward the hard ones. Use functional testing where possible. Otherwise, use an event count routine.

Troubleshooting

If any of the above tests fail, then troubleshoot the failing circuit. To do this, get familiar with how the probe works. And prepare some guided fault isolation programs like the one we used on the TK-80A.

Often you can find the cause of a problem by following a schematic through the circuit with the probe and looking at the probe for indication of a stuck node.

In the old days technicians were forced to swap known-good components for suspect ones. Often they would trace the problem to an area with 5 or 10 components, then replace them all. Now you can do away with that form of troubleshooting. You can use a guided fault isolation program immediately to find the failing component. Further, you can write the program so that an unskilled operator can do the probing just by following displayed instructions.

One technique you may need to use is that of jumpering signals into open connectors. The reason would be that you do not have the circuit available which would normally be connected there. Sometimes, you can use the probe to inject signals into open edge connectors, etc., but it is not the ideal way to troubleshoot. Try wherever you can to duplicate an operational system.

HOW TO BE SURE THE 9010A IS RIGHT FOR YOU

Time Is Money, As Always

Usually, by the time you have purchased a system which tests microprocessor products, you have $15,000 (US) or more invested. At that, it is usually much more cumbersome and less effective to use than the 9010A. For this reason, return on investment is not an issue.

Where the 9010A Troubleshooter pays and pays big is in TIME.

For example, one of our customer's manufacturing groups was scrapping processor boards for months because they were cheaper to throw away than to troubleshoot and repair. Average repair time was 4 hours. The cost to them was either the price of the board ($75), or the repair time ($120) plus parts ($5).

Our design group brought in a 9010A prototype in an effort to help. In 4 days they had diagnosed and repaired the faults in all but two of the 37 scrap boards. Those two had internally damaged signal lines. At 45 minutes average repair time, they saved over $50 per board, $1850 for the batch. And, we thought we'd never get our prototype 9010A back from the manufacturing group.

It Can't Do Everything

The 9010A is not much better-suited to some tasks than for others. For example, it can effectively test or troubleshoot only digital boards which are controlled by a microprocessor.

Also, as one might expect, it is incapable of making analog measurements.

But Maybe It's Right For You

However, you may find the 9010A without an equal in troubleshooting microprocessor-based systems. Having worked through this manual, you should have a good idea whether the 9010A can fit your application. If it can, and if you save anywhere near the amount of time we think you'll save, then maybe the 9010A is for you.

Add to that the fast response from our world-wide service team, Fluke's enduring reputation for product excellence, and our committment to continuing support. When you put the 9010A Micro-System Troubleshooter to work in your manufacturing and service organizations, you will have the competetive edge you've been wanting.

If you still have questions about how to match the 9010A to your needs, don't hesitate to ask your Fluke representative.

APPENDIX A - 9010A DEMONSTRATION TAPE LISTING

This appendix lists the contents of the 9010A demonstration tape as it would be printed via the AUX-IF =, SETUP, LEARN, and PROGM operations. The program steps in the listings are the same as those displayed when you are editing a program except that labels are identified in the left margin for reading convenience. We printed them in 40-character lines (we changed LINESIZE in SETUP) so we could get two columns on a page.

AUX-I/F =, SETUP, and LEARN

PROG	SIZE
00	280
03	18
04	12
05	120
06	16
07	14
08	8
09	8
11	15
20	1300
21	452
22	234
23	204
24	246
25	74
26	136
27	58
30	631
31	253
32	484
33	37

AUX-I/F PROGM

```
SETUP INFORMATION

POD - 8080
ENABLE READY-YES
ENABLE HOLD-YES

TRAP BAD POWER SUPPLY-YES
TRAP ILLEGAL ADDRESS-YES
TRAP ACTIVE INTERRUPT-NO
TRAP ACTIVE FORCE LINE-NO
TRAP CONTROL ERROR-YES
TRAP ADDRESS ERROR-YES
TRAP DATA ERROR-YES

EXERCISE ERRORS-YES
BEEP ON ERR TRANSITION-YES

BUS TEST @ 8FFF
RUN UUT @ 0000
TIMEOUT 200
STALL 13
UNSTALL 11
NEWLINE 00000D0A
LINESIZE 79

ADDRESS SPACE INFORMATION

RAM @ 8C00-8FFF

ROM @ 0000-07FF SIG F77C
```

```
PROGRAM 0   280 BYTES

    DPY-9010A DEMO, VERSION
    DPY-+30-JUN-81
    SYNC ADDRESS
    REG1 = 7
0:  LABEL 0
    DPY-+#
    EXECUTE PROGRAM 6
    DEC REG1
    IF REG1 > 0 GOTO 0
    EXECUTE PROGRAM 9
1:  LABEL 1
    DPY-WANT TO TEST TK-80 DISPLAY
    DPY-+#?1
    IF REG1 = 0 GOTO 4
    EXECUTE PROGRAM 32
2:  LABEL 2
    DPY-DID DISPLAY TEST PASS#?1
    IF REG1 = 1 GOTO 4
3:  LABEL 3
    DPY-TROUBLESHOOT TK-80 DISPLAY
    DPY-+#?1
    IF REG1 = 0 GOTO 4
    EXECUTE PROGRAM 20
4:  LABEL 4
    DPY-WANT TO TEST TK-80
    DPY-+ KEYBOARD#?1
    IF REG1 = 0 GOTO 5
    EXECUTE PROGRAM 30
5:  LABEL 5
    DPY-END OF TK-80 TEST#

PROGRAM 3   18 BYTES

    EXECUTE PROGRAM 4
1:  LABEL 1
    IF REGB = 40 GOTO 1

PROGRAM 4   12 BYTES

    REGB = 40
    DPY-+%B

PROGRAM 5   120 BYTES

    REG0 = REG8
    REG4 = 1
1:  LABEL 1
    REG1 = REG0 SHR SHR SHR SHR SHR SHR SHR SHR
    REG2 = REG1 SHR SHR SHR SHR SHR SHR SHR SHR
    REG3 = REG2 SHR SHR SHR SHR SHR SHR SHR SHR
    WRITE @ REGF INC = REG3 AND FF
    WRITE @ REGF INC = REG2 AND FF
    WRITE @ REGF INC = REG1 AND FF
    WRITE @ REGF INC = REG0 AND FF
    IF REG4 = 2 GOTO 2
    INC REG4
    REG0 = REG9
    GOTO 1
2:  LABEL 2

PROGRAM 6   16 BYTES

0:  LABEL 0
    INC REG1
    IF 5 > REG1 GOTO 0

PROGRAM 7   14 BYTES

    EXECUTE PROGRAM 6
    EXECUTE PROGRAM 6
    EXECUTE PROGRAM 6
    EXECUTE PROGRAM 6

PROGRAM 8   8 BYTES

    EXECUTE PROGRAM 7
    EXECUTE PROGRAM 7

PROGRAM 9   8 BYTES

    EXECUTE PROGRAM 8
    EXECUTE PROGRAM 8

PROGRAM 11   15 BYTES

    REGF = 8FF7
    EXECUTE PROGRAM 5
    EXECUTE PROGRAM 9

PROGRAM 20   1300 BYTES

    DPY-DISPLAY FAULT ISOLATION#
    EXECUTE PROGRAM 9
    WRITE @ 100FB = 92
    WRITE @ REGF DEC = 80
0:  LABEL 0
    DPY-TEST TIMING AND INPUTS
    EXECUTE PROGRAM 9
    REG8 = 5002811
    REG9 = 4005
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = FE733427
    REG9 = 3400
    EXECUTE PROGRAM 21
    IF REGB = 1D GOTO F
    REG8 = 10103426
    REG9 = 3402
    EXECUTE PROGRAM 21
    IF REGB = 1D GOTO F
    REG8 = 5002810
    REG9 = 402805
    EXECUTE PROGRAM 21
    IF REGB = 1D GOTO F
    REG8 = 5002803
    REG9 = 51071905
    EXECUTE PROGRAM 21
    IF REGB = 1D GOTO F
    REG8 = REG8 INC INC
```

```
    REG9 = 2805
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
1:  LABEL 1
    DPY-TEST INTERNAL CONTROLS
    EXECUTE PROGRAM 9
    REG8 = FCE65302
    REG9 = 3000
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5005303
    REG9 = 2805
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5005108
    REG9 = 3405
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    REG9 = 53515205
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5005311
    REG9 = 53525105
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    REG9 = 51525305
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 13135313
    REG9 = 41283502
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5005308
    REG9 = 51525305
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    REG9 = 51525305
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
2:  LABEL 2
    REG8 = 7BF2001
    REG9 = 5300
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5002704
    REG9 = 2005
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
3:  LABEL 3
    DPY-TEST ADDRS COUNT / CONTROL
    EXECUTE PROGRAM 9
    REG8 = 5002601
    REG9 = 51535205
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    REG9 = 5405
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 1062603
    REG9 = 2606
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
4:  LABEL 4
    DPY-TEST DIGIT DECODER
    EXECUTE PROGRAM 9
    REG8 = 5005512
    REG9 = 2705
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 5005501
    REG9 = 5505
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    INC REG8
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
5:  LABEL 5
    DPY-TEST SEGMENT DRIVE BUFFER
    EXECUTE PROGRAM 9
    REG8 = AA55AA55
    REG9 = REG8
    EXECUTE PROGRAM 11
    REG8 = 2082104
    REG9 = 2106
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 2082110
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 2082115
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = REG8 INC INC
```

```
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
    REG8 = 2082121
    EXECUTE PROGRAM 21
    IF REG8 = 1D GOTO F
 6: LABEL 6
    DPY-#REPLACE DISP DIGIT/SEGMENT
    DPY-+ CKT
    STOP
 F: LABEL F
    DPY-END FAULT ISOLATION#
    EXECUTE PROGRAM 8
```

PROGRAM 21   452 BYTES

```
    REG3 = REG8 AND FF
    REGA = REG8 SHR SHR SHR SHR SHR SHR SHR SHR
    REG2 = REGA AND FF
    REG4 = REGA SHR SHR SHR SHR SHR SHR SHR SHR
    REG5 = 0
    EXECUTE PROGRAM 4
    DPY-#PROBE U$2 PIN $3
    REGA = 0
    EXECUTE PROGRAM 24
    IF REG8 = 40 GOTO 0
    IF 40 > REG8 GOTO F
    REG8 = 40
    GOTO 2
 0: LABEL 0
    REGD = REG4
    DPY-U$2-$3
    EXECUTE PROGRAM 26
    REG0 = REGC
    REG1 = REGD
    EXECUTE PROGRAM 23
    IF 40 > REG8 GOTO C
 5: LABEL 5
    IF REG9 AND 2 > 0 GOTO 1
    IF REGA = REG0 GOTO B
    GOTO 2
 1: LABEL 1
    IF REGA > REG1 GOTO 2
    IF REGA >= REG0 GOTO B
 2: LABEL 2
    IF 40 > REG8 GOTO C
    DPY-+BAD; LOOP
    IF REG5 > 0 GOTO 0
    DPY-+#?5
    IF REG5 > 0 GOTO 0
    REG6 = REG9
    DPY-REPLACE BAD
 3: LABEL 3
    REG6 = REG6 SHR SHR SHR SHR SHR SHR SHR SHR
    IF REG6 = 0 GOTO 4
    REG7 = REG6 AND FF
    DPY-+ U$7
    GOTO 3
 4: LABEL 4
    DPY-+#: CONT?6
    IF REG6 = 1 GOTO C
    IF 40 > REG8 GOTO D
    DPY-+%B
    GOTO D
 B: LABEL B
    DPY-+GOOD
    IF 40 > REG8 GOTO C
    IF REG5 > 0 GOTO 0
 C: LABEL C
    EXECUTE PROGRAM 9
    IF REG8 = 40 GOTO E
    IF REG8 > 25 GOTO F
    IF 25 > REG8 GOTO F
    REG8 = 40
    IF REG5 = 0 GOTO F
    REG5 = 0
    DPY-+%B
    GOTO 5
 D: LABEL D
    REG8 = 1D
    GOTO F
 E: LABEL E
    DPY-+%B
 F: LABEL F
    REGA = 1
    EXECUTE PROGRAM 24
```

PROGRAM 22   234 BYTES

```
 0: LABEL 0
    WRITE @ 100FB = 92
    WRITE @ REGF DEC = 80
    REGD = 0
    DPY-SIG=0; LEVEL=1,5; COUNT=2,6
    DPY-+?
    EXECUTE PROGRAM 3
 1: LABEL 1
    REG9 = REGB
    IF REG9 > 6 GOTO 0
    IF REG9 = 3 GOTO 0
    IF REG9 = 4 GOTO 0
    EXECUTE PROGRAM 4
    IF REG9 AND 4 = 0 GOTO 2
    REG1 = REG9
    REG8 = AA55AA55
    REG9 = REG8
    EXECUTE PROGRAM 11
    REG9 = REG1
 2: LABEL 2
    REG8 = 0
    REGC = 0
    REGD = 0
    DPY-$9; SYNC
    IF REG9 AND 4 = 0 GOTO 3
    DPY-+ FREE RUN..
    GOTO 4
 3: LABEL 3
    DPY-+ ADDRESS..
 4: LABEL 4
    EXECUTE PROGRAM 26
    IF 40 > REG8 GOTO 1
    EXECUTE PROGRAM 23
    EXECUTE PROGRAM 8
    GOTO 2
```

PROGRAM 23   204 BYTES

```
    SYNC FREE-RUN
    IF REG9 AND 4 > 0 GOTO 1
    SYNC ADDRESS
 1: LABEL 1
    IF REG9 AND 3 > 0 GOTO 2
    REG1 = REGD
    READ PROBE
    EXECUTE PROGRAM 27
    READ PROBE
    REGA = REG0 SHR SHR SHR SHR SHR SHR SHR SHR
```

```
   AND FFFF
      DPY-+$A
      REGD = REG1
      GOTO E
 2:   LABEL 2
      IF REG9 AND 1 > 0 GOTO 4
      IF REG9 AND 2 = 0 GOTO F
      READ PROBE
      READ PROBE
      IF REG9 AND 4 > 0 GOTO 3
      EXECUTE PROGRAM 27
      READ PROBE
 3:   LABEL 3
      REGA = REG0 AND 7F
      DPY-+$AH
      GOTO E
 4:   LABEL 4
      IF REG9 AND 4 = 0 GOTO 5
      READ PROBE
      EXECUTE PROGRAM 6
      READ PROBE
      GOTO 6
 5:   LABEL 5
      READ PROBE
      EXECUTE PROGRAM 27
      READ PROBE
 6:   LABEL 6
      REGA = REG0 AND 7000000
      REGC = REGA
      EXECUTE PROGRAM 25
 E:   LABEL E
      DPY-+,
 F:   LABEL F
```

PROGRAM 24   246 BYTES

```
      SYNC FREE-RUN
      IF REGA > 0 GOTO 0
      IF REG9 AND 10 > 0 GOTO 4
      IF REG9 AND 1 = 0 GOTO 0
      IF REG8 AND 7000000 = 2000000 GOTO 4
 0:   LABEL 0
      REG1 = 10
 1:   LABEL 1
      READ PROBE
      IF REGA = 0 GOTO 2
      IF REG0 AND 5000000 = 0 GOTO 3
      DPY-REMOVE PROBE
      GOTO 0
 2:   LABEL 2
      IF 40 > REGB GOTO D
      IF REG0 AND 5000000 = 0 GOTO 0
 3:   LABEL 3
      DEC REG1
      IF 40 > REGB GOTO F
      IF REG1 > 0 GOTO 1
      GOTO F
 4:   LABEL 4
      DPY-+, CONTINUE
 5:   LABEL 5
      IF REGB = 40 GOTO 5
      IF REGB = 25 GOTO 6
      GOTO F
 6:   LABEL 6
      EXECUTE PROGRAM 4
      GOTO F
 D:   LABEL D
      IF 25 > REGB GOTO F
      IF REGB > 25 GOTO F
      EXECUTE PROGRAM 4
      REGB = 41
      DPY-+
 F:   LABEL F
```

PROGRAM 25   74 BYTES

```
      IF REGC AND 1000000 = 0 GOTO 1
      DPY-+H
 1:   LABEL 1
      IF REGC AND 2000000 = 0 GOTO 2
      DPY-+X
 2:   LABEL 2
      IF REGC AND 4000000 = 0 GOTO 3
      DPY-+L
 3:   LABEL 3
```

PROGRAM 26   136 BYTES

```
      IF REG9 AND 3 = 0 GOTO 1
      IF REG9 AND 1 = 1 GOTO 2
      IF REG9 AND 2 = 2 GOTO 3
      GOTO 4
 1:   LABEL 1
      REGC = REGD
      DPY-+SIG $C
      GOTO 4
 2:   LABEL 2
      REGC = REGB AND 7000000
      DPY-+LEVEL
      EXECUTE PROGRAM 25
      GOTO 4
 3:   LABEL 3
      REGC = REGD SHR SHR SHR SHR SHR SHR SHR SHR
   AND 7F
      REGD = REGD AND 7F
      DPY-+CNT $C-$D
 4:   LABEL 4
      DPY-+=
```

PROGRAM 27   58 BYTES

```
      WRITE @ 100FB = 92
      DTOG @ REGF DEC = 0 BIT 7
      REGF = 8FF7
 1:   LABEL 1
      WRITE @ REGF INC = AA
      WRITE @ REGF = 55
      IF 8FFF > REGF GOTO 1
```

PROGRAM 30   631 BYTES

```
      DPY-TK-80 KEYBOARD TEST#
      DPY-+...WAIT
      EXECUTE PROGRAM 4
      REGF = 88FF
      REG8 = 3F
      REG9 = 6
      EXECUTE PROGRAM 5
      REG8 = 5B
      REG9 = 4F
      EXECUTE PROGRAM 5
      REG8 = 66
      REG9 = 6D
      EXECUTE PROGRAM 5
      REG8 = 7D
      REG9 = 7
```

```
    EXECUTE PROGRAM 5
    REG8 = 7F
    REG9 = 6F
    EXECUTE PROGRAM 5
    REG8 = 77
    REG9 = 7C
    EXECUTE PROGRAM 5
    REG8 = 39
    REG9 = 5E
    EXECUTE PROGRAM 5
    REG8 = 79
    REG9 = 71
    EXECUTE PROGRAM 5
    REG8 = 377937
    REG9 = 5078 6F
    EXECUTE PROGRAM 5
    REG8 = 501C54
    REG9 = 73507B
    EXECUTE PROGRAM 5
    REG8 = 6D787B73
    REG9 = 547B
    EXECUTE PROGRAM 5
    REG8 = 7C50
    REG9 = 5073
    EXECUTE PROGRAM 5
    REG8 = 733E6D76
    REG9 = 3F3779
    EXECUTE PROGRAM 11
    REG2 = 100FA
    REG3 = 100F8
    WRITE @ 100FB = 92
    DPY-+#
1:  LABEL 1
    READ @ STS
    IF REGC AND 10 = 10 GOTO E
    IF 40 > REGB GOTO F
    REG1 = EF
2:  LABEL 2
    WRITE @ REG2 = REG1 AND FF
    READ @ REG3
    IF FF > REGE GOTO 3
    IF REG1 = 3BC GOTO 7
    REG1 = REG1 SHL
    GOTO 2
3:  LABEL 3
    REGA = 0
    REGE = REGE CPL AND FF
4:  LABEL 4
    SHR REGE
    IF REGE = 0 GOTO 5
    INC REGA
    GOTO 4
5:  LABEL 5
    IF REG1 = EF GOTO 6
    SHR REG1
    REGA = REGA INC INC INC INC INC INC INC INC
    GOTO 5
6:  LABEL 6
    REG8 = 0
    EXECUTE PROGRAM 31
    IF 40 > REGB GOTO F
    GOTO 1
7:  LABEL 7
    DPY-PUSH A TK-80 KEY; RESET=
    DPY-+EXIT
    WRITE @ 8FFC = 0
    WRITE @ REGF INC = 3F
    WRITE @ REGF INC = 37
    WRITE @ REGF INC = 79
    WRITE @ 100FB = 92
    GOTO 1
E:  LABEL E
    READ @ STS
    DPY-KEY "RESET"
    REG9 = 5078 6D
    EXECUTE PROGRAM 11
    IF 40 > REGB GOTO F
    IF REGC AND 10 = 10 GOTO E
    EXECUTE PROGRAM 4
F:  LABEL F
    DPY-END OF TK-80 KEYBOARD TEST
    REG8 = 0
    REG9 = 79545E
    EXECUTE PROGRAM 11

PROGRAM 31    253 BYTES

    DPY-TK-80 KEY PRESSED =
    IF REGA = 10 GOTO 0
    IF REGA = 11 GOTO 1
    IF REGA = 12 GOTO 2
    IF REGA = 13 GOTO 3
    IF REGA = 14 GOTO 4
    IF REGA = 15 GOTO 5
    IF REGA = 16 GOTO 6
    IF REGA = 17 GOTO 7
    DPY-+$A
    GOTO 8
0:  LABEL 0
    DPY-+MEM
    GOTO 8
1:  LABEL 1
    DPY-+REG
    GOTO 8
2:  LABEL 2
    DPY-+RUN
    GOTO 8
3:  LABEL 3
    DPY-+PREV
    GOTO 8
4:  LABEL 4
    DPY-+STEP
    GOTO 8
5:  LABEL 5
    DPY-+NEXT
    GOTO 8
6:  LABEL 6
    DPY-+BRK
    GOTO 8
7:  LABEL 7
    DPY-+RP
8:  LABEL 8
    REG1 = REGA SHL SHL OR 8C00
    REG2 = 8FFC
9:  LABEL 9
    READ @ REG1
    WRITE @ REG2 = REGE
    INC REG1
    INC REG2
    IF 9000 > REG2 GOTO 9

PROGRAM 32    484 BYTES
    DPY-#TK-80 DISPLAY TEST
    EXECUTE PROGRAM 4
    EXECUTE PROGRAM 8
    WRITE @ 100FB = 92
```

```
    WRITE @ REGF DEC = 80
    IF 40 > REGB GOTO F
    DPY-LIGHT ALL DISPLAY SEGMENTS
    REG8 = FFFFFFFF
    REG9 = FFFFFFFF
    EXECUTE PROGRAM 11
    IF 40 > REGB GOTO F
    DPY-CLEAR TK-80 DISPLAY
    REG8 = 0
    REG9 = 0
    EXECUTE PROGRAM 11
    IF 40 > REGB GOTO F
    DPY-START WITH RIGHT DIGIT
    EXECUTE PROGRAM 7
    REGF = 8FFF
    REGA = 1
1:  LABEL 1
    REGE = 1
2:  LABEL 2
    DPY-TESTING DIGIT $A SEGMENT $E
    WRITE @ REGF = REGE
    EXECUTE PROGRAM 6
    IF 40 > REGB GOTO F
    SHL REGE
    IF 100 > REGE GOTO 2
    REGC = 0
    REGE = 3F
    EXECUTE PROGRAM 33
    REGE = 6
    EXECUTE PROGRAM 33
    REGE = 5B
    EXECUTE PROGRAM 33
    IF 40 > REGB GOTO F
    REGE = 4F
    EXECUTE PROGRAM 33
    REGE = 66
    EXECUTE PROGRAM 33
    REGE = 6D
    EXECUTE PROGRAM 33
    IF 40 > REGB GOTO F
    REGE = 7C
    EXECUTE PROGRAM 33
    REGE = 7
    EXECUTE PROGRAM 33
    REGE = 7F
    EXECUTE PROGRAM 33
    REGE = 67
    EXECUTE PROGRAM 33
    DPY-TESTING DIGIT $A = BLANK
    WRITE @ REGF = 0
    EXECUTE PROGRAM 6
    IF 40 > REGB GOTO F
    INC REGA
    DEC REGF
    IF 9 > REGA GOTO 1
    IF 40 > REGB GOTO F
    EXECUTE PROGRAM 4
F:  LABEL F
    DPY-END OF DISPLAY TEST
    REG8 = 0
    REG9 = 79545E
    EXECUTE PROGRAM 11

PROGRAM 33    37 BYTES
    DPY-TESTING DIGIT $A = $C
    WRITE @ REGF = REGE
    INC REGC
    EXECUTE PROGRAM 6
```

APPENDIX B - TK-80A SINGLE BOARD COMPUTER DRAWINGS

NOTE: Drawings are on next page

Figure 5. TK-80A Schematic Diagram

Figure 6. TK-80A Component Locator