MICRO-SYSTEM TROUBLSHOOTER
ADVANCED TRAINING

Outline

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

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

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

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

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

Micro-System Troubleshooter Training
Evaluation Form

Location:

Instructor:

Please indicate below the area which best describes your job function.

	R&D	Mfg. Test	Mfg. Repair	Depot Service	Field Service	Design	Calib.
Manager							
Engineer							
Sr. Technician							
Technician (0-4 years)							

Please circle the answer below which reflects your opinion of the training session.

Was the content or value to you?

Excellent			Good			Fair			Poor
10	9	8	7	6	5	4	3	2	1

Will the presented ideas be useful for your application?

Excellent			Good			Fair			Poor
10	9	8	7	6	5	4	3	2	1

How effective were the course materials?

Excellent			Good			Fair			Poor
10	9	8	7	6	5	4	3	2	1

What was the overall effectiveness of the instructor presentation?

Excellent			Good			Fair			Poor
10	9	8	7	6	5	4	3	2	1

Please circle the appropriate answer to the questions below:

Was the overall level:	Too advanced	About right	Too basic
Was the pace (tempo):	Too fast	About right	Too slow
Was time allowed for questions:	Too much	About right	Too little
Was the response to questions:	Too involved	About right	Inadequate

Comments:

ADVANCED MICRO-SYSTEM
TROUBLESHOOTER TRAINING

Laboratory Workbook

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

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

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

This binder and all material inside is now the property of the student. In addition, all programs developed in class will be recorded on a tape which is also yours to keep.

SECTION I - REVIEW BASIC OPERATIONS

Inputting "KNOWN GOOD" information

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

1. _____   3. _____

2. _____   4. _____(optional)

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

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

       ROM is MEMORY MAPPED from address 0 TO 7FF.

       I/O is I/O MAPPED from address F8 to FA
       with ONLY bits 3 & 5 read-writable.

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

Built-in Tests

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

   |___| Bus Test

   _____

   _____

   _____

   _____

   |___| RAM Short Test

   _____

   _____

   _____

   _____

   |___| RAM Long Test

   _____

   _____

   _____

   _____

   |___| ROM Test

   _____

   _____

   _____

   _____

   |___| I/O Test

   _____

   _____

   _____

   _____

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

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

   _____

   _____

Troubleshooting Functions

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

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

   _____

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

   _____

3. What troubleshooter functions are associated with STS/CTL ?

   _____

Probe Operation

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

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

	Probe Function	Valid Sync Modes		
		Free Run(F)	Sync'd (O-E)	No Affect
I.	_____	☐	☐	☐
II.	_____	☐	☐	☐
III.	_____	☐	☐	☐
IV.	_____	☐	☐	☐

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

   _____

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

   _____

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

Registers

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

1. How many registers are available for data storage in the Troubleshooter ? _____

2. How are the registers identified ? _____

3. What are DEDICATED registers and how are they identified ?

    _____

4. What are NON-DEDICATED registers and how are they identified ?

    _____

5. What are GLOBAL registers and how are they identified ?

    _____

6. What are LOCAL registers and how are they identified ?

    _____

7. Where was the above information found the quickest ?

    _____

Arithmetic Functions

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

1. What troubleshooter function(s) allows you to count within a program ? _____

2. The SHIFT LEFT and SHIFT RIGHT functions will cause the binary digits in a value to be shifted one place to the LEFT or RIGHT each time you press the appropriate key. If you took 8FFF and shifted it once to the RIGHT what would the result be in hex ?

    _____

3. If you took the answer to question 9 and shifted it LEFT one time what would the result be ? _____

4. What would the result be if you took 8FF8 and ANDed it with F00F ? _____

5. What would the result be if you took 8FF8 and ORed it with F00F ?

    _____

6. What is the COMPLIMENT of 8FF8 ? _____

Basic Programming

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

1. What are the proper key strokes for creating a program ?
    _____

2. What is the REAL limiting factor on program storage capability and what is its maximum value ? _____

3. Once inside a program, in the PROGRAMMING mode, what method(s) are available to move form one step to another ?
    _____

4. What Troubleshooter function(s) are used for UNCONDITIONAL JUMPING ? _____

5. What Troubleshooter function(s) are used for CONDITIONAL JUMPING ? _____

6. What Troubleshooter function can you use in a program to print messages on a printer ? _____

7. What Troubleshooter function can you use to write messages to the display ? _____

8. When entering a display message in the programming mode, what key(s) will delete only one character ? _____

Built-in Tests in a Program

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

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

Now execute program 1

SECTION II - UTILITY PROGRAMS

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

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

The Probe and the Program - B0163

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

1. What are the three basic steps when gathering probe information ?

       1.   _____

       2.   _____

       3.   _____

2. How is the probe data transferred into a program ?

_____

3. With the above information, make a program that:

    A. Gets the probe data.

    B. Seperates the three pieces of probe information into three seperate registers.

    C. And can be used as a subroutine whenever probe information is desired.

PROGRAM NOTES

4. To test the program developed in question 3 make another program
    that will:

    A. Stimulate the UUT with a RAMP function.

    B. Separate the probe information using the program in
        question 3.

    C. And display the three pieces of probe data.

PROGRAM NOTES

Identifying Address and Data lines

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

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

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

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

To help in developing the program answer the following questions:

1. What STIMULUS from the Troubleshooter should you use to identify the address and data lines ? _____

2. What piece of probe data should be used to identify the address and data line when using the stimulus from question 1 ?
   _____

3. Using the utility program for the probe, what register will contain the probe data for question 2 ? _____

4. What value will you expect to find in register ___ (Ans. Q3) when the probe is on an address or data line ? _____

Use the space provided to write your program.

PROGRAM NOTES

Mapping Devices on your UUT

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

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

        A.   Use the probe utility program.

        B.   Stimulate the UUT in such a way as to identify the conditions which cause the chip to be selected.

        C.   If found, display the condition.

        D.   If not found, display "CAN'T IDENTIFY".

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

1. What usually determines when a device is selected ?

   _____

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

   _____

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

   _____

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

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

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

Using the space below design the chip select program.

PROGRAM NOTES

Display and Print Messages

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

For example take the message:

PLACE PROBE ON PIN 5 OF U30

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

PROBE U30 PIN 5

Or simply

PROBE U30-5

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

For your next problem, design a program that will:

A. Use a single register to recieve all the message variables from the calling program.

B. Seperate each variable from the register.

C. Display the variables in their proper place in the message.

D. Returns to the calling program.

First answer the following:

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

   _____

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

   _____

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

   _____

PROGRAM NOTES

Probe Placement Detector

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

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

Your next problem is to design a program that:

        A.   When called will determine when the probe is placed on a
            node.

        B.   Will return to the calling program indicating that the
            probe is placed properly or is on an open circuit.

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

1.   What function of the probe will be used to detect that it is
      placed on a node ? _____

2.   Using the function choosen in question 1, what will indicate that
      the probe is sitting on a node ? _____

3.   How will the program detect an open node (high impedance)?
      _____

4.   What will be the SYNC mode for this program ? _____

5.   How will the program sense that the probe is firmly in place on a
      node ? _____

6.   How will this program indicate to the calling program that the
      probe is on an active node, or on an open circuit?
      _____

PROGRAM NOTES

PROGRAM NOTES

SECTION III - ACTUAL CIRCUIT APPLICATIONS

Working With I/O Devices

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

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

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

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

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

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

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

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

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

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

Example #1

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

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

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

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

1.   How many ports are available through the 8255 ? _____

2.   What are the different modes of operation for the 8255 ?

     _____

3.   How are the various modes set on the 8255 ?

     _____

4.   What does bit 7 in the control word do ? _____

     _____

5.   What function does pin 35 serve ? _____

6.   What lines are used to transfer data between the microprocessor
     and the PIA? _____

7.   Which pin(s) and what logic level does it take to select the PIA?

     _____

8.   Which pin(s) and what logic level(s) will select the port(s) of
     the 8255 ? _____

9.   Are there any other signal conditions that must be met in order
     to operate this device ? _____ If so, what are they ?

     _____

Step 2 - <u>Determine how the device is used for the UUT application.</u>

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

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

1.   What ports are used for the keyboard application ? _____

2.   Looking at the circuit design. what gives you the clue that a port is used as input? _____

3.   Which direction are these ports configured?

      _____

4.   What gives the best clue as to what mode is used on the 8255 ?

      _____

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

Step 3 - <u>Identify how the device is accessed by the microprocessor on the UUT.</u>

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

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

1.   What address must the microprocessor put on the address bus to select this chip ? _____

In addition to selecting the device itself the microprocessor must also select which of the ports it is communicating with (see Q.8 of step 1).

2.   What microprocessor line(s) are tied to the port select pin(s) ?

      _____

3.   What microprocessor signals are needed to satisfy the requirements of Step #1 question 9? _____

      _____

Step 4 - <u>Identify how to stimulate the device and how to evaluate the results.</u>

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

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

_____

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

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

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

_____

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

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

_____

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

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

With the PIA properly configured you will want to try and sense the keyboard back through the 8080 pod. To test out our approach to the problem, let's say that we want to test Key 0.

5. Looking at the schematic, what steps have to be taken before we can see the expected data back through the pod ? _____

_____

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

_____

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

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

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

_____

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

A.   Instruct the operator to reset the PIA and have the 9010A sense when it has been done.

B.   Configure the PIA for testing keys 0-7 on the keyboard.

C.   Instruct the operator which key to press.

D.   Test the keyboard to verify that the proper key was pressed. If correct, test the next key. If incorrect, report the error to the operator.

PROGRAM NOTES

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

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

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

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

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

1. Before you can probe the port, however, what logic level should be selected on the probe to stimulate port A? _____

2. Is it necessary to synchronize the probe to provide the proper stimulus? _____

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

PA0 _____

PA1 _____

PA2 _____

PA3 _____

PA4 _____

PA5 _____

PA6 _____

PA7 _____

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

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

MAKE SURE PULSER IS OFF

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

	INPUT		OUTPUT
D_0	96EC	PC0	_____
D_1	_____	PC1	_____
D_2	_____	PC2	_____
D_3	_____	PC3	_____
D_4	_____	PC4	_____
D_5	_____	PC5	_____
D_6	_____	PC6	_____
D_7	_____	PC7	_____

This same technique can be use for all output ports.

4. If you use the same stimulus should the signatures be different for each output port ? (For example is PA0 different than PB0 or PC0) _____

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

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

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

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

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

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

6. What is the significant difference between Mode 0 and Mode 1
    operation ? _____

7. What control word will make port A input and port B output in
    Mode 1 ? _____

8. Can port C be used while ports A and B are in mode 1 ? _____
    If so how ? _____

9. What are the pin numbers and labels for the handshake lines for
    both the input and output ports ? _____

10. How can you use the Troubleshooter to pass data in and out of
    these ports when handshaking is necessary ?
    _____

Using the clothes pin adapter, test the 8255 in mode 1 by configuring
the device as port A input and port B output. Write data to port B,
pass the data to port A by using the answer from question 12 then read
port A.

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

_____

_____

_____

_____

_____

_____

_____

_____

Example #2

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

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

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

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

         D   _____

         CK _____

         PRE _____

         CLR _____

2.   Are any of these inputs set by the design of the UUT? _____ If so, which lines? _____

3.   Is it possible to change the configuration of U52B through program control? _____

Step 2 - <u>Determine how the device is used for the UUT application.</u>

1.   What is the application of U52B?

      _____

      _____

Step 3 - <u>Identify how the device is accessed by the microprocessor on the UUT.</u>

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

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

      _____

2.   Determine the configuration that U30 must be in to test U52B.

      _____

3.   What is the control word, in hex, that will configure U30 properly ? _____

Step 4 - <u>Identify how to stimulate the device and how to evaluate the results.</u>

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

2. What 9010A stimulus could be used to satisfy the clock input?

    _____

3. Once the stimulus has been determined for the clock, what format should that statement take to leave the clock LOW when the stimulus is complete? _____

4. What piece of probe data could be used to determine if U52B is operating properly? _____

5. What is the complete 9010A command that will configure U30 for this test ? _____

6. What addresse(s) will be needed to test U52B? _____

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

        A. Configure U30 properly.

        B. Have the operator place the probe on the Q output using the following utility programs:

            1. display
            2. probe placement detector

        C. Have the program stimulate U52B so that a predetermined count will be collected by the probe.

        D. If the count is incorrect, have the 9010A display the results.

        E. If the count is correct, then display OK.

PROGRAM NOTES

PROGRAM NOTES

USING RUN UUT

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

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

```
8C00    21 40 8C    LXI H,8C40H     8C1B    77          MOV M,A
8C03    36 50       MVI M,50        8C1C    01 FF 0F    LXI B,0FFFH
8C05    31 00 8D    LXI SP,8D00H    8C1F    0B          DCX B
8C08    3E 01       MVI A,01        8C20    79          MOV A,C
8C0A    21 F8 8F    LXI H,8FF8H     8C21    B0          ORA B
8C0D    77          MOV M,A         8C22    C2 1F 8C    JNZ 8C1FH
8C0E    2C          INCR L          8C25    F1          POP PSW
8C0F    C2 0D 8C    JNZ 8C0DH       8C26    07          RLC
8C12    F5          PUSH PSW        8C27    C3 0A 8C    JMP 8C0A
8C13    21 40 8C    LXI H,8C40H     8C2A    3E 01       MVI A,01H
8C16    7E          MOV A,M         8C2C    D3 FA       OUT FAH
8C17    3D          DCR A           8C2E    76          HLT
8C18    CA 2A 8C    JZ 8C2AH
```

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

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

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

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

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

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

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

KEYPAD