FLUKE

Welcome to the world of the Micro-System Troubleshooter

and the repair of microprocessor-based equipment.

Our Micro-System Troubleshooter was very carefully researched, with you, our customers, before it was designed and built. We wanted to address and solve a difficult service problem — the repair of microprocessor-based equipment.

During this seminar you will become familiar with the Micro-System Troubleshooter and how to apply its operations to your specific troubleshooting problems. Through hands-on exercises you will learn effective Micro-System Troubleshooter techniques used to diagnose failures encountered in actual practice.

This seminar is another service we provide to enable you to use your Micro-System Troubleshooter as effectively as possible. We feel that continuing service to you is our obligation because we appreciate the opportunity of serving you as a customer and in making our products in your hands very useful to you. If, at any time, we may be of further service to you, please call us. Also, should we ever be of inadequate service to you, we would also appreciate a call from you so that we may correct any deficiencies. Lastly, we are privileged and pleased to have you as our customer.

John M. Fluke

John Fluke Mfg. Co., Inc. / PO Box C9090 / Everett WA 98206 / (206) 347 6100 / TWX 910 445 2943 / TLX 15 2662

Seminar Outline

I INTRODUCTION
- A. Introduce instructors
- B. Overview of Session

I. SELF TEST
- A. Power up
- B. POD

II. INPUTTING "KNOWN GOOD INFORMATION"
- A. Learn
- B. Manual map entry
- C. Viewing the memory map

IV. TAPE OPERATION
- A. Write
- B. Read

V. TESTING
- A. BUS Test
- B. ROM Test
- C. RAM Test
- D. I O Test
- E. AUTO Test
- F. RAM LONG Test
- G. Specifying Test Address

VI. MODE KEYS
- A. Without Errors
- B. With Errors
- C. RUN UUT

——— Lunch ———

VII. TROUBLESHOOTING FUNCTIONS
- A. Read
- B. Write
- C. Ramp
- D. Walk
- E. Toggle

VIII. PROBE OPERATION
- A. Logic Probe
- B. Pulser
- C. Signature Analysis
- D. Transition Counter

IX. REGISTERS
- A. Types
- B. Entering and Reading Data
- C. Counting with Registers (Incr & Decr)
- D. Shifting
- E. Binary operation

X. PROGRAMMING
- A. Creating
- B. Executing
- C. Editing
- D. Display
- E. Branching

XI. AUXILIARY INTERFACE
- A. Set Up Menu
- B. Select Switches
- C. AUX I/F Key

XII. THE PROBE AND THE PROGRAM
- A. Read Probe
- B. Identifying Device Addresses

Laboratory Workbook

The objective of this lab is to help familiarize the student with the different operating features of the Troubleshooter.

The following is a list of equipment that is necessary for the performance of this lab.

1 ea. 9010A Microsystem Troubleshooter
1 ea. 8080A POD
1 ea. NEC TK-80 with Schematic
2 ea. Microclip Leads
1 ea. Power supply cable/connector

Set-Up

1. Connect the 8080A POD to the Microsystem Troubleshooter using the connector on the under side of the mainframe.

2. Connect the power cord between the TK-80 (UUT) and the Troubleshooters RS-232 port.

Self-Test

The Troubleshooter does a self test on the mainframe when it is powered up. Turn on the Troubleshooter by pressing the GREEN power switch located on the front left edge of the mainframe. Note the Troubleshooter display for self test verification.

To test the POD insert the microprocessor plug into the microprocessor test socket on the POD and lock it in place. With power applied press BUS test on the mainframe and note the self test results on the Troubleshooter display. If BUS test fails ask for assistance from one of the instructors.

Now that you have tested the Troubleshooter system you are ready to use it with your UUT. Insert the POD plug into the UUT's microprocessor socket, insuring that the key on the plug is aligned with pin 1 of the μP socket.

With one exception, the Troubleshooter must know the location of RAM, ROM, and I/O before it can test the UUT. There are three ways of putting this information into the Troubleshooters memory. The input method will depend on whether or not the operator has the documentation of the UUT memory map.

LEARN

When the operator does not have a memory map of a UUT it is only necessary to LEARN the UUT with the following keystrokes.

KEY	DISPLAY
LEARN	LEARN @
ENTER	LEARN NOW 0000

Viewing the Memory Map

Now you can find out what the Troubleshooter has learned from the previous section.

1. To see where RAM is press the VIEW RAM key in the upper left corner of the keyboard.

What are the RAM locations for the UUT? _____
8C00 -8FFF

2. To see where ROM is, press the VIEW ROM key in the upper left corner of the keyboard.

What are the ROM locations for the UUT? _____
0000 - 7FFF 10 0C0 - 100FF

What is the Signature for this block of ROM? _____
F77C 083F (F77C)

3. To see where I/O is, press the VIEW I/O key in the upper left corner of the keyboard.

What are the I/O locations for the UUT? _____
NO

Learning a UUT in Sections

Sometimes a UUT's address space will have to be LEARNed in sections. One reason would be when two or more ROM chips are used in a microprocessor system, and their addresses are consecutive. This normally results in one block of ROM with one signature to be computed during the LEARN process. It is usually beneficial to develop signatures for each chip, to aid in faulty chip isolation.

For the benefit of this example assume that there are two 1K ROM chips on the UUT instead of one 2K chip. The previous learned information indicates that ROM is at address spaces 0 through 7FF. You will split this ROM section up into two equal spaces.

Before doing a sectionalized LEARN you will have to clear out the old ROM descriptor as follows:

KEY	DISPLAY
VIEW ROM	ROM @ 0000-07FF SIG F77C
CLEAR/NO	NO ROM INFO

You can do a LEARN on a specified section of the UUT address space as follows:

KEY	DISPLAY
LEARN	LEARN = _
0	LEARN @ 0_
ENTER	LEARN @ 0-_
3	LEARN @ 0-3_
F	LEARN @ 0-3F_
F	LEARN @ 0-3FF_
ENTER	LEARN @ 0-3FF NOW 0000

1. What is the new signature for the ROM at 0 to 3FF?

Manual Map Entry

If a memory map or documentation of a UUT is already known, it can be entered manually. In this next exercise you will be entering the second 1K block of ROM into the Troubleshooter using the manual entry method.

KEY	DISPLAY
VIEW ROM	ROM @ 0000-03FF SIG 0ED3
ENTER	ROM @ _
4	ROM @ 4_
0	ROM @ 40_
0	ROM @ 400_
ENTER	ROM @ 400-_
7	ROM @ 400-7_
F	ROM @ 400-7F_
F	ROM @ 400-7FF_
ENTER	ROM @ 400-7FF SIG _

Now the Troubleshooter is prompting you to enter a signature. If you knew the signature, you could manually enter it at this time. In this case you do not know the signature so you can have the Troubleshooter compute the signature.

KEY	DISPLAY
ENTER	ROM @ 400-7FF SIG WAIT

1. What is the new signature for the second half of the fictional ROM block? _____

Now view the ROM as you have stored it in the Troubleshooters memory.

KEY	DISPLAY
ROM VIEW	ROM @ 0000-03FF SIG 0ED3

Now notice that the MORE annunciator is flashing to let you know that there is more to the ROM descriptor stored in the Troubleshooters memory. To see what it is:

KEY	DISPLAY
MORE	ROM @ 0400-07FF SIG D23F

If you want to see what the first descriptor was just press:

KEY	DISPLAY
PRIOR	ROM @ 0000-03FF SIG 0ED3

RAM information can be entered into the Troubleshooters memory in the same way ROM information was. RAM, however; does not require a signature.

I/O information is also entered in the same way as ROM but instead of signature information, the Troubleshooter will ask for "BTS". BTS means Bits and in this case is used to specify which Data Bits, for this address, are read/writeable. This is usually referred to as a BIT-MASK. The "BTS" also indicates that the bit mask must be in hexadecimal form.

First you should practice converting some bit masks into hexadecimal form:

2. What hexadecimal number will represent a bit mask with bits 3 thru 6 read/writeable? _____

(Note: first construct a binary number that has bits 3 thru 6 with ones and then convert it to a Hexadecimal number.)

3. What hexadecimal number will represent a bit mask with bits 0 thru 7 read/writeable? _____

4. What hexadecimal number will represent a bit mask with bits 6 thru 13 read/writeable? _____

The 8080 POD has 8 data bits labeled 0 thru 7. For every bit of the bit mask that has a 1 in it, that bit will be designated read/writeable. This bit mask must be entered into the Troubleshooter in a hexadecimal form.

The UUT has a Programmable Interface for the Keyboard, display and cassette input/output circuitry. Just as the word programmable indicates the interface must be programmed for a particular configuration. This will be covered later in the course but for right now enter the proper I/O locations in the Troubleshooter as follows:

KEY	DISPLAY
I/O VIEW	NO IO INFO
ENTER	IO @ _
1	IO @ 1_
0	IO @ 10_
0	IO @ 100_
F	IO @ 100F_
8	IO @ 100F8_
ENTER	IO @ 100F8-_
1	IO @ 100F8-1_
0	IO @ 100F8-10_
0	IO @ 100F8-100_
F	IO @ 100F8-100F_
A	IO @ 100F8-100FA_
ENTER	IO @ 100F8-100FA BTS _

Now the Troubleshooter is prompting you for the bit mask. Remember, it must be in hexadecimal form. This particular space of I/O has all bits read/writeable.

5. What is the proper hexadecimal number that shows all bits read/writeable? _____

Now key in the proper bit mask found in question 5.

KEY	DISPLAY
ENTER	IO @ 100F8-100FA BTS (Ans Q.5)

To VIEW the I/O area that is mapped in the Troubleshooter memory you simply press I/O VIEW key.

Testing

Thus far, you have covered how information about a "known good" board is manually entered into the Troubleshooter or automatically "LEARNED" from the "known good" UUT. This next section will show how the Troubleshooter uses this information to test defective UUTs.

The easiest way to begin testing and troubleshooting UUTs, is to use the preprogrammed (canned) tests of the Troubleshooter. These "canned" tests are activated by the six buttons located in the lower left-hand corner of the front panel. These buttons will cause tests to be performed on the UUT which are "built in" to the Troubleshooter — no programming on the part of the user is required.

Bus Test

The BUS test button will cause the Troubleshooter to verify that the UUT microprocessor *bus* is functional. This test automatically checks that:

a) Data lines of the UUT bus are not being held high or held low (this might be caused by a short circuit which is accidently connecting a data line to ground or to a power supply, or it might be caused by an IC connected to the data bus which has failed),

b) No data line is shorted to any other data line,

c) Address lines of the UUT bus are not being held high or held low,

d) No address line is shorted to any other address line,

e) Other output control lines from the UUT microprocessor are not being held high or low.

Now perform the BUS test:

KEY	DISPLAY
BUS	BUS TEST WAIT
(after short delay)	BUS TEST OK

In this brief interval, all of the above-listed tests have been performed.

Actually, the BUS test can be performed on any UUT without first having "known good" information on the UUT in the Troubleshooter memory. It is good practice to activate the BUS test as the very *first* operation on any UUT. This will verify that the bus is operational. The Troubleshooter can only perform its function if the bus is up and operating.

Now you will inject a bus error in order to see the diagnostic abilities of the bus test. Using one of the microhook jumper wires, connect one end of the jumper wire to the ground connection provided by the small loop of bare wire soldered into the UUT board near U36. Connect the other end of this jumper to U34 pin 15.

Now hit BUS test again.

1. What is the fault that is now reported on the display?

_____

_____

2. Which red LED annunciator is blinking? _____

_____

3. Is the user prompted with a question at the end of the displayed message? _____

_____

4. In addition to the diagnostic message (ending in a question) and the blinking red LED, did the Troubleshooter employ any other means for communicating that an error had occurred? (Hit BUS test again if you have difficulty.) _____

_____

Now remove the jumper wire and hit BUS test again to verify that you have cleared the fault. After a short delay, the display should respond with "BUS TEST OK".

ROM Test

The ROM test is used to verify proper data stored in the UUT ROM. The Troubleshooter accomplishes this by reading the data stored in one block of ROM and then computes a signature for the data read from this block. This signature is then compared with the signature "learned" or manually entered from a "known good" system. If the signatures differ, then the UUT contains ROM data which differs from the ROM data in the "known good" board.

The ROM test function automatically performs this procedure for each ROM descriptor contained in the Troubleshooters main memory.

To initiate the ROM test, press the ROM button in the group of buttons labelled "TESTS". You should get the following response:

KEY	DISPLAY
ROM	ROM TEST @ _
ENTER	ROM TEST WAIT
(after a short delay)	ROM TEST OK

The ROM test on the demo UUT takes about 11 seconds. During this period, the Troubleshooter checks each block of UUT ROM for its proper signature.

The ROM on the UUT is good, so this test should always pass. In order to inject a ROM fault, you will short a pin on the ROM IC to make it malfunction. Using one of the microhook jumper wires, connect one end of the jumper wire to the ground connection provided by the small loop of bare wire soldered into the UUT board near U36. Connect the other end to pin 17 of the ROM IC (U1) on the UUT (this is the upper left-hand corner pin on the IC which is in the upper left-hand corner of the board).

Now perform the ROM test again by pressing the ROM button in the group of buttons marked "TESTS". You should get the following:

KEY	DISPLAY
ROM	ROM TEST @ _
ENTER	ROM TEST WAIT
(after a short delay)	ROM ERR @ .....

As before when the Troubleshooter identified a UUT fault, the instrument generates an audible "beep" to alert the operator that a fault has been found. The displayed message also ends in a question (which allows the operator to loop on this failure if he so desires). Also, the "STOPPED" led is flashing to alert the operator that the ROM test has terminated prematurely. The test stopped and reported the *first* ROM error. There may be more ROM blocks yet to be tested. Other information contained in this diagnostic message is as follows:

1. The location (or address limits) of the failing block of ROM is given in the displayed message. For this failure, what are the address limits of the detected failure? ___

2. The "MORE" led is flashing. This indicates that the Troubleshooter has more information about this failure. To obtain this information, press the MORE button. What is the additional information about this fault? ___

This indicates precisely what the differences were between the signature calculated on the UUT and the "known good" signature.

3. Based on this second part of the diagnostic message, does the user still have an opportunity to loop on this failure? ___

4. Is there still more information about this fault, or do you have all the information available? ___

If you forgot what the address limits were for this particular failure, you could recall the first diagnostic message to the display by pressing PRIOR.

Now clear this fault by removing the jumper wire. Press ROM test and ENTER, to verify that the UUT ROM now tests "good".

RAM SHORT Test

The RAM SHORT test is a quick test to verify that the UUT RAM is good. The Troubleshooter performs this test by writing data into each RAM location, and then reading this data from each location and, finally, verifying that the data read was the same as the data written. This test is conducted by writing "ones" into all bit positions of each RAM location as well as by writing "zeros" into all bit positions of each RAM location. This verifies that all bits of each RAM address is "read-writeable."

In addition to these read-write tests, other tests are performed automatically on the UUT RAM when the user performs the RAM SHORT test. These additional tests verify that each RAM address is properly decoded. This type of testing is called address-decoding verification.

To initiate the RAM SHORT test, press the RAM SHORT button:

KEY	DISPLAY
RAM SHORT	RAM SHORT @ _
ENTER	RAM SHORT WAIT
(after a brief delay)	RAM SHORT OK

The RAM SHORT function performs all the above described testing in about 15 seconds.

To generate some RAM failures, you will gound a pin on one of the RAM ICs. Using one of the microhook jumper wires, connect one end of the jumper wire to the ground connection provided by the small loop of bare wire soldered into the UUT near U36. Connect the other hook to U17 pin 10 (this is the pin in the upper left-hand corner of U17.)

Now press RAM SHORT AGAIN:

KEY	DISPLAY
RAM SHORT	RAM SHORT @ _
ENTER	R/W ERR @ .....

The Troubleshooter is now informing the user that a RAM fault has been detected. Further, the message indicates that the fault is a Read/Write ERRor (R/W ERR). This means that, at one or more RAM locations, the Troubleshooter could not successfully perform the read/write test described above.

1. What is the address of the RAM location which failed the read/write test? ___

2. Is there any additional information about the failure contained in this message? ___

In diagnosing RAM failures, it is important to know if just one bit in a RAM location is failing, or more than one, or perhaps all bits are failing the read/write test. The Troubleshooter informs the user which bits in the RAM location are failing by indicating "BTS xx" in the displayed message. The hexadecimal value "xx" tells the user exactly which bits in the RAM location were *not* read/writeable.

In this particular case, to identify which bits are bad, you must convert the hexadecimal value "FF" to a binary number. This is easily accomplished by looking at the "F" key on the Troubleshooter and noticing that the binary equivalent (printed under the "F") is "1111". Hence, the binary value of the hex number "FF" is "11111111". This tells the user that all eight bits of the RAM location are bad. If only one bit had been bad, the binary value would have contained only one "1" and this would be positioned so as to indicate the bit position which had failed in the RAM location (e.g., "00001000").

Now disconnect the jumper wire and press RAM SHORT, then ENTER to verify that the RAM failure has now been cleared.

I/O Test

The I/O test verifies that the Input/Output registers in the UUT exhibit the read/write characteristics of those registers from the "known good" system. In a previous section of this lab you manually entered the address descriptors of I/O registers from known good information. You also entered which particular bits of these registers were read/writeable. For these particular bits, the I/O descriptor says these bits can be written to a

"one", and then read them as a "one", followed by writing them to a "zero", and successfully reading them as a "zero". The I/O test can be used to verify that these I/O register bits are still read/writeable on malfunctioning UUTs.

When the user activates the I/O TEST, the Troubleshooter simply reads and writes ("ones" and "zeros") to each I/O register and using "known good" information, automatically verifies that the proper bits in these registers are read/writeable.

On the UUT, I/O registers are built into a large IC called an 8255 Programmable Peripheral Interface. In order to make these I/O registers read/writeable, you must send the Interface chip a command. Once configured, you can test the registers using the I/O TEST function.

To send out this command, you will use a function on the Troubleshooter which have not yet been discussed. To properly configure the Interface you will WRITE to address 100FB using 80 as the data word.

KEY	DISPLAY
WRITE	WRITE @ _
1	WRITE @ 1_
0	WRITE @ 10_
0	WRITE @ 100_
F	WRITE @ 100F_
B	WRITE @ 100FB_
ENTER	WRITE @ 100FB = _
8	WRITE @ 100FB = 8_
0	WRITE @ 100FB = 80_
ENTER	WRITE @ 100FB = 80 OK

Now you can test the I/O registers on your UUT. Press the I/O button in the "TESTS" group and you should get the following response:

KEY	DISPLAY
I/O	I/O TEST @ _
ENTER	I/O TEST OK

The read/writeable bits in the three I/O registers are good.

To inject a fault into one of these I/O registers, you need only make one read/writeable bit malfunction. Since the keyboard *on the UUT* is wired up to these registers, you can generate such a fault by holding down certain multiple keys.

For example, on the UUT hold down the "4" key and the "C" key at the same time. Now — while you are holding these keys down — press I/O TEST and ENTER on the Troubleshooter.

The Troubleshooter should inform you immediately that one bit is not read/writeable in the UUT I/O registers.

1. The Troubleshooter informs you that there is a Read/Write error (R/W ERR) in one of the I/O registers. What is the address of the register which exhibits this error? _____

_____

2. The Troubleshooter diagnostic message is also telling you which bit failed the read/write test in this register. Specifically, what part of the message contains this information? _____

_____

3. By looking at the Troubleshooter display, can you tell at this time if this is the only I/O fault on the UUT? _

_____

Now try the I/O test again without holding down any keys on the UUT board. Press I/O TEST and ENTER to verify that the fault has been cleared.

Auto Test

Usually, when a repair technician begins troubleshooting a faulty UUT, he will have no idea where the fault lies. Hence, he will want to run all of the tests discussed so far. To make the operation of the Troubleshooter a little simpler for this application, an AUTO test button has been included which (when pressed) will cause the BUS test, ROM test, RAM SHORT, and I/O test to be run automatically in sequence.

To initiate this sequence of tests, simply press the AUTO button (no ENTER keystroke is required):

KEY	DISPLAY
AUTO	AUTO TEST WAIT
(after 26 seconds)	AUTO TEST OK

If there are any faults discovered in the BUS test, ROM test, RAM SHORT, or I/O test, these faults will be reported as described previously. The AUTO test stops and reports the *first* fault found.

RAM LONG Test

The RAM LONG test is a more extensive test of UUT RAM. Because of this, it is a longer test, taking about ten times as long to complete as the RAM SHORT test. Because of this, the repair technician will always use RAM SHORT first to get a quick verification of UUT RAM. Only if he suspects that there might be a RAM fault which "slipped by" the RAM SHORT test will he resort to the RAM LONG test.

The RAM LONG test is included on the Troubleshooter to test for a special type of RAM failure called a pattern-sensitive failure or RAM soft failures. this type of failure is peculiar to semiconductor RAM devices in use today. Although it is an infrequent type of error (it is usually the type of error that causes computers to "hiccup" intermittently. The RAM LONG test is specifically designed to catch these pattern-sensitive RAM failures.

The RAM LONG test is initiated like the other tests:

KEY	DISPLAY
RAM LONG	RAM LONG @ _
ENTER	RAM LONG WAIT
(after 3.2 minutes)	RAM LONG OK

The RAM LONG test will, like the RAM SHORT test, catch read/write faults, and address-decoding faults. In

addition, the RAM LONG test will catch pattern-sensitive RAM failures and report them with a message like:

"RAM PATT ERR @ 8C00-LOOP?"

This informs the user that a RAM pattern error has occurred and provides the first address wherein the error was detected. The Troubleshooter also provides additional information about the bits which were involved in the pattern error.

Use of Test Functions with Address Specification

Because the RAM LONG test takes so long, if the technician suspects a particular part of UUT RAM, he may want to test just this small portion of RAM rather than test the entire UUT RAM. The RAM LONG test will finish in a much shorter time if you test only a small portion of RAM. Hence you may want to "override" the address limits of the RAM descriptor stored in main memory as "known good" information and, instead, specify a smaller block of RAM to be tested by this test.

You can explicitly command the RAM LONG test on any portion of RAM as follows:

KEY	DISPLAY
RAM LONG	RAM LONG @ _
8	RAM LONG @ 8_
C	RAM LONG @ 8C_
0	RAM LONG @ 8C0_
0	RAM LONG @ 8C00_
ENTER	RAM LONG @ 8C00-_
8	RAM LONG @ 8C00-8_
C	RAM LONG @ 8C00-8C_
1	RAM LONG @ 8C00-8C1_
0	RAM LONG @ 8C00-8C10_
ENTER	RAM LONG @ 8C00-8C10 WAIT
(after 2 seconds)	RAM LONG @ 8C00-8C10 OK

This operation has caused the Troubleshooter to perform the RAM LONG test on only the first 17 RAM locations (addresses 8C00 to 8C10 inclusive). You have not altered the "learned" address limits of RAM; you have just called for a RAM LONG test to be performed over a special set of addresses which you explicitly state at the time that you command the test.

in a similar fashion, all of the test functions (except BUS and AUTO) can be commanded with explicit addresses specified at the time you command the test. If you do not wish to explicitly specify any addresses, press ENTER when the Troubleshooter prompts you for these addresses. For example, if you press RAM LONG and ENTER, the Troubleshooter will use the "known good" RAM address limits for this RAM LONG test.

Mode Keys

The effect that the STOP, CONTinue, RePEAT, and LOOP functions have on the Troubleshooter depends on the activity taking place at the time one of these keys is pressed. There are two general cases; one case where errors are involved, and the other case where errors are not involved.

Without Errors

The RePEAT key is pretty self explanatory, for it just repeats the last action performed by the Troubleshooter. e.g., you've just performed a RAM LONG TEST @ 8C00-8C10 in the previous section. To perform the same test without having to reenter all the information press:

KEY	DISPLAY
RPEAT	RAM LONG @ 8C00-8C10 WAIT

If you wanted the Troubleshooter to continually repeat the last operation over and over agin you should use the LOOP key.

KEY	DISPLAY
LOOP	RAM LONG @ 8C00-8C10 OK

Notice that the LOOPING annunciator, at the right end of the display, is flashing.

To regain control of the Troubleshooter simply press STOP and the LOOPING annunciator will go out and the STOPPED annunciator will light. The STOP key will always bring the Troubleshooter back to the immediate mode.

The CONTinue key has no meaningful effect on the Troubleshooter operation unless errors are involved or you wish to continue a stopped program.

With Errors

First you need to induce a failure into the UUT. Using the UUT drawing and a microclip jumper, short pins 15 and 17 together on U34. Now perform a BUS TEST.

KEY	DISPLAY
BUS TEST	BUS TEST WAIT
After a few seconds the display =	DATA BITS 0 AND 1 TIED-LOOP/

Notice that the STOPPED annunciator is flashing. This, of course, indicates that a malfunction has been detected and Troubleshooter operation has stopped somewhere in the BUS TEST cycle. The display is also asking, with LOOP?, if you wish to continually exercise the test operation on the detected malfunction. There are a number of ways to proceed from here. By pressing YES or LOOP you will cause the Troubleshooter to loop on the detected failure.

KEY	DISPLAY
YES	DATA BITS 0 AND 1 TIED

The LOOPING annunciator is now flashing. The Troubleshooter is now exercising only that portion of the BUS TEST that detects DATA BITS 0 and 1 TIED.

Now remove one end of the shorting jumper on the UUT and watch the display. Touch the loose end of the jumper back to its original pin and again watch the display.

1. What types of failures is the looping feature most useful for? _____

_____

After reconnecting the shorting jumper, press STOP.

2. What is the Troubleshooter display indicating? ___

Now press CONTinue.

3. What is the Troubleshooter display indicating? ___

Remove the microclip jumper from U34.

In the next steps of this example it is important to connect the jumper in the order shown (ground connection last). Now connect one end of the microclip jumper to pin 2 of U40. Next connect the other end to the wire loop that is next to the upper right hand corner of socket U36 (Ground). Now press:

KEY	DISPLAY
BUS TEST	

4. What is the displayed message? _____

Notice that the MORE annunciator is flashing.

MORE

5. What is the second part of the message? _____

The first message is telling you that one of the microprocessor FORCE lines is being held high. The second message is helping you identify which line is being held high. The second message should be:

DISPLAY

STS BITS 0001 0000-LOOP?

STS means status. This message is showing an eight bit Status word.

6. With the right most bit being bit 0, what bit is showing a 1? _____

7. The bit with the 1 in it is the line being held high. Using the STATUS LINES chart on the POD, what function is associated with this bit? _____

Now exercise this fault by pressing:

KEY	DISPLAY
YES (or LOOP)	ACTIVE FORCE LINE
Lift the ground end of the jumper.	NO FORCE LINE
Reground the Jumper.	ACTIVE FORCE LINE

With the jumper still connected, assume, for the purpose of this example, that you can not correct this ACTIVE FORCE LINE but you still want to continue troubleshooting the UUT. You have to ignore this problem to get the board operating again.

KEY	DISPLAY
SETUP	SET-TRAP BAD POWER SUPPLY? YES
Press MORE until the display =	SET-TRAP ACTIVE FORCE LINE? YES
NO	SET-TRAP ACTIVE FORCE LINE? NO
BUS TEST	BUS TEST WAIT

Now you can continue troubleshooting the UUT even though the reset line is being held high. Now return TRAP ACTIVE FORCE LINE to YES and remove the Microclip from ground first and then pin 2 of U40.

Now connect the microclip to pin 15 of U16 and pin 16 of U17 on the UUT. Then press:

KEY	DISPLAY
RAM SHORT	RAM SHORT @ __
ENTER	RAM SHORT WAIT
After a few seconds the display will =	RAM DCD ERR @ 8C00 BIT 9-LOOP?

What the Troubleshooter is trying to tell you is that bit 9 is not functioning properly.

8. What is the binary value of $8C00_{16}$? _____

9. Remembering that the LSB is bit 0, what would the Hexadecimal equivalent be to the binary number if you changed bit 9? _____

So, the message is trying to tell you that the Troubleshooter can't differentiate between address 8C00 and (Ans. to Q. 10).

Now LOOP on this failure and remove the jumper. What is the Troubleshooter display? _____

Reconnect the jumper to the same pins and perform a BUS TEST. The jumper shorts address line 8 and 9 together.

10. If the address lines are tied together why doesn't the BUS TEST reveal this problem like it did before? ____

You can also set the Troubleshooter up so that it will not report any errors and therefore becomes a go/no-go tester. e.g.

KEY	DISPLAY
SETUP	SET-TRAP ACTIVE FORCE LINES? YES
Press MORE until the display =	SET-EXERCISE ERRORS? YES
NO	SET-EXERCISE ERRORS? NO
MORE	SET-BEEP ON ERR TRANSITION? YES
NO	SET-BEEP ON ERR TRANSITION? NO

Now with the shorting jumper connected as in the previous step:

KEY	DISPLAY
ROM TEST	ROM TEST @ __
ENTER	ROM TEST WAIT

After awhile the Troubleshooter will display ROM TEST FAIL.

The error caused by the difference in signatures was not reported and the display was only appended with FAIL. If a device, like a printer, was connected to the Auxiliary Interface it would have printed all failures encountered during the TEST. This feature would be useful if you were doing a RAM LONG TEST overnight.

Run UUT

Remove the jumper from U16 and U17.

The RUN UUT function allows the operator to select a mode of operation that makes the Troubleshooter act as a microprocessor on the UUT so that normal UUT operation can be performed without removing the POD and replacing the microprocessor. e.g.

KEY	DISPLAY
RUN UUT	RUN UUT @ _

This message is now asking for the address of the first program instruction for the microprocessor to start at. You can either enter a starting address or use a stored default address. You will use the default address.

KEY	DISPLAY
ENTER	RUN UUT - MAY NEED RESET

Now the UUT will work normally. Hit reset and then type in some characters with the UUT keyboard and watch the UUT display.

Changing the default address for RUN UUT is done through the SET-UP function.

Troubleshooting Keys

The troubleshooting keys, READ, WRITE, RAMP, WALK, TOGGL ADDR and TOGGL DATA are used to stimulate or monitor a particular address or bit on a microprocessor bus.

The READ function allows the operator to examine the contents of a specific address. e.g.

KEY	DISPLAY
READ	READ @ _
0	READ @ 0_
ENTER	READ @ 0 = 22 OK

The 22 is a hexadecimal representation of the data contained in address space 0000. The OK only means that the Troubleshooter has completed the read operation successfully and does not verify that the address contents are correct.

Find the data contents of the following address locations.

1. 3DF = _____
2. 7FF = _____
3. 64 = _____
4. 2 = _____

The Troubleshooter can also read the status lines associated with the microprocessor. e.g.

KEY	DISPLAY
READ	READ @ _
STS/CTL	READ @ STS = 0000 00001 OK

5. What STATUS line is being held high? _____

To see how you can use the READ STATUS function in a real application, short pin 1 of U38 to the ground loop between U36 and U37 on the UUT. Next perform a BUS TEST.

6. What is the error message displayed by the Troubleshooter? _____

At this point the Troubleshooter has no control over the UUT. You have to either fix the problem or get the Troubleshooter to ignore the problem, provided the UUT can be run if you ignore the problem.

Now refer to the Technical Data Bulletin called "Avoiding Potential Problems When Getting Started" located in the Application notes section of your notebook. Locate the section that covers what to do when you have the error message that you recorded in question 6.

Now that you have cleared the problem you should identify what was causing it. Do a READ STATUS as you did before.

7. What status lines are active now? _____

Now ENABLE one of these lines and see if the operation changes.

8. What caused the problem detected in question 6? _

The WRITE function allows the operator to send data to a specific address or drive microprocessor control lines. The display of the UUT is located at memory address spaces 8FF8 thru 8FFF. 8FF8 is the first digit of the display and 8FFF is the last digit.

Now WRITE to the first digit in the UUT display as follows:

KEY	DISPLAY
WRITE	WRITE @ _
8	WRITE @ 8_
F	WRITE @ 8F_
F	WRITE @ 8FF_
8	WRITE @ 8FF8_
ENTER	WRITE @ 8FF8 = _

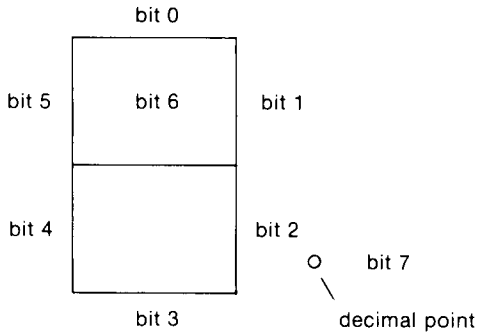
Now enter the value you want to send to 8FF8.

KEY	DISPLAY
7	WRITE @ 8FF8 = 7
ENTER	WRITE @ 8FF8 = 7 OK

8. What is the first digit of the UUT display? _____

To get the UUT to display this character you had to send it a hexadecimal number that would light the proper segments of the display.

The diagram below identifies which bit of the data word will turn on a specific segment of the display.



To determine what hexadecimal number to write, to the display for a given combination of segments, just set the bit, for each segment you want lit, to 1 and convert the resulting data word to hexadecimal.

9. What Hexadecimal number will represent a 3 in the UUT display? _____

10. What Hexadecimal number will represent a 9 in the UUT display? _____

Not only can the Troubleshooter write to any address space but it can also write to some of the control lines.

11. Using the chart on the POD, which control lines are USER WRITEABLE? _____

Remember, the control lines are displayed in binary.

12. What would the BINARY word be to write to the control lines with only HLDA set high? _____

Now using the WRITE and STS/CTL keys set the HLDA line high.

KEY	DISPLAY
WRITE	WRITE @ _
STS/CTL	WRITE @ CTL = _
(Ans. Q 12)	WRITE @ CTL = (Ans. Q 12) OK

The RAMP function will write data, to an address, that will be an increasing number starting at zero and progressing in one bit steps until all data lines are ones. This function can be demonstrated by writing a ramp to one of the UUT displays.

The first digit of the UUT display is 8FF8.

KEY	DISPLAY
RAMP	RAMP @ _
8FF8	RAMP @ 8FF8_
ENTER	RAMP @ 8FF8 WAIT

This show doesn't last very long so press REPEAT and watch the first digit of the UUT display.

The WALK function causes the Troubleshooter, through a series of write operations, to WALK a bit pattern at an operator specified address. This is accomplished by writing the original data specified and then rotate the data one bit, and write the new data to the same address. This process continues until the data is rotated around completely. The following example illustrates the eight write operations performed by the WALK function on an eight line data bus. The data specification is entered into the Troubleshooter is Hexadecimal (01), but to make the process easier to see, it is shown here in binary (00000001):

00000001
10000000
01000000
00100000
00010000
00001000
00000100
00000010

Now perform a WALK at one of the display addresses.

KEY	DISPLAY
WALK	WALK @ _
8FF8	WALK @ 8FF8_
ENTER	WALK @ 8FF8 = _
01	WALK @ 8FF8 = 01_
ENTER	WALK @ 8FF8 = 01 WAIT

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.

The TGGL DATA function will cause the Troubleshooter to toggle an operator specified data bit from one logic state to the other. Now perform a TOGGL DATA at one of the UUT display addresses.

KEY	DISPLAY
TOGGL DATA	DTOG @ _

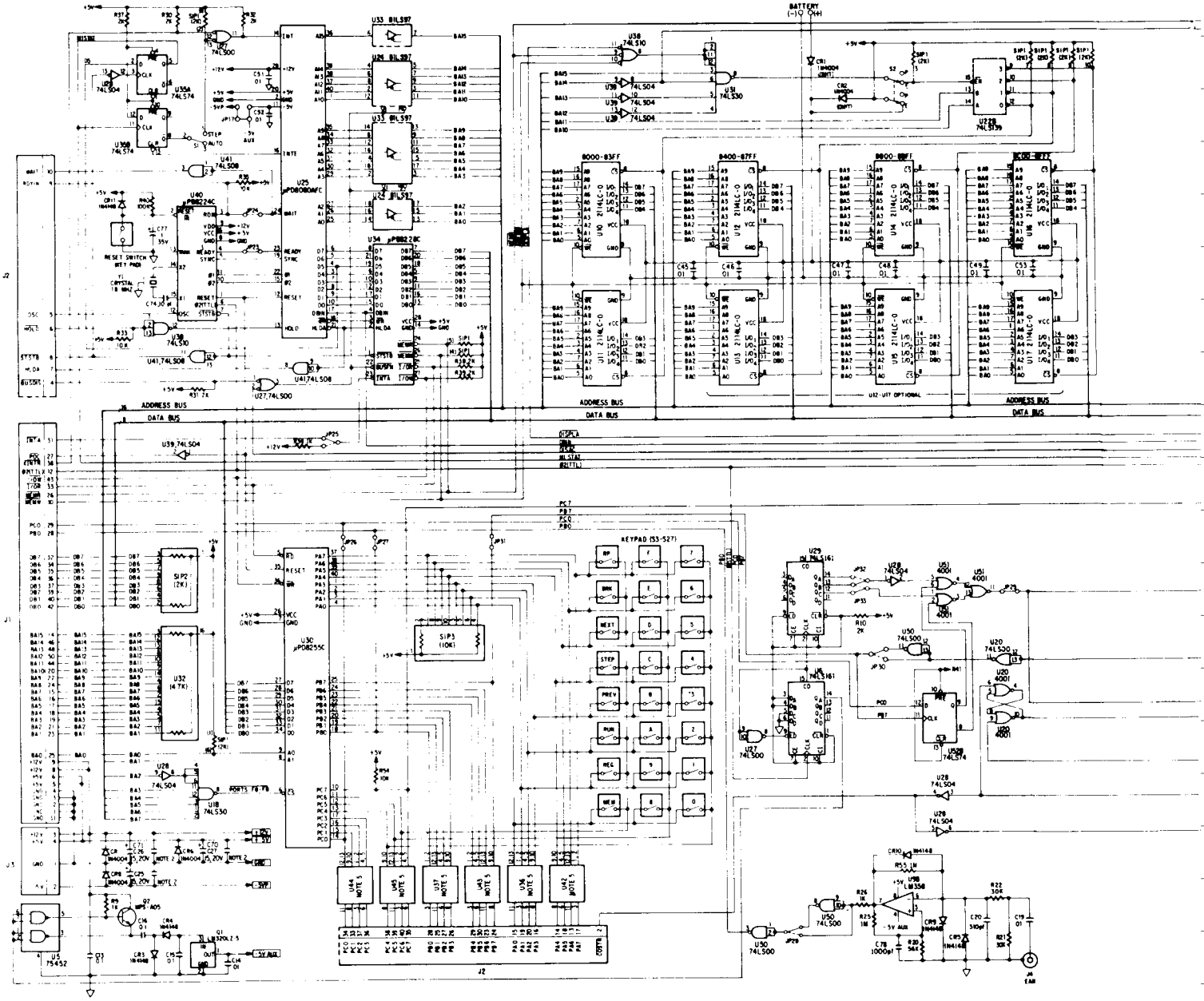
Now enter the address at which you want to do a data toggle.

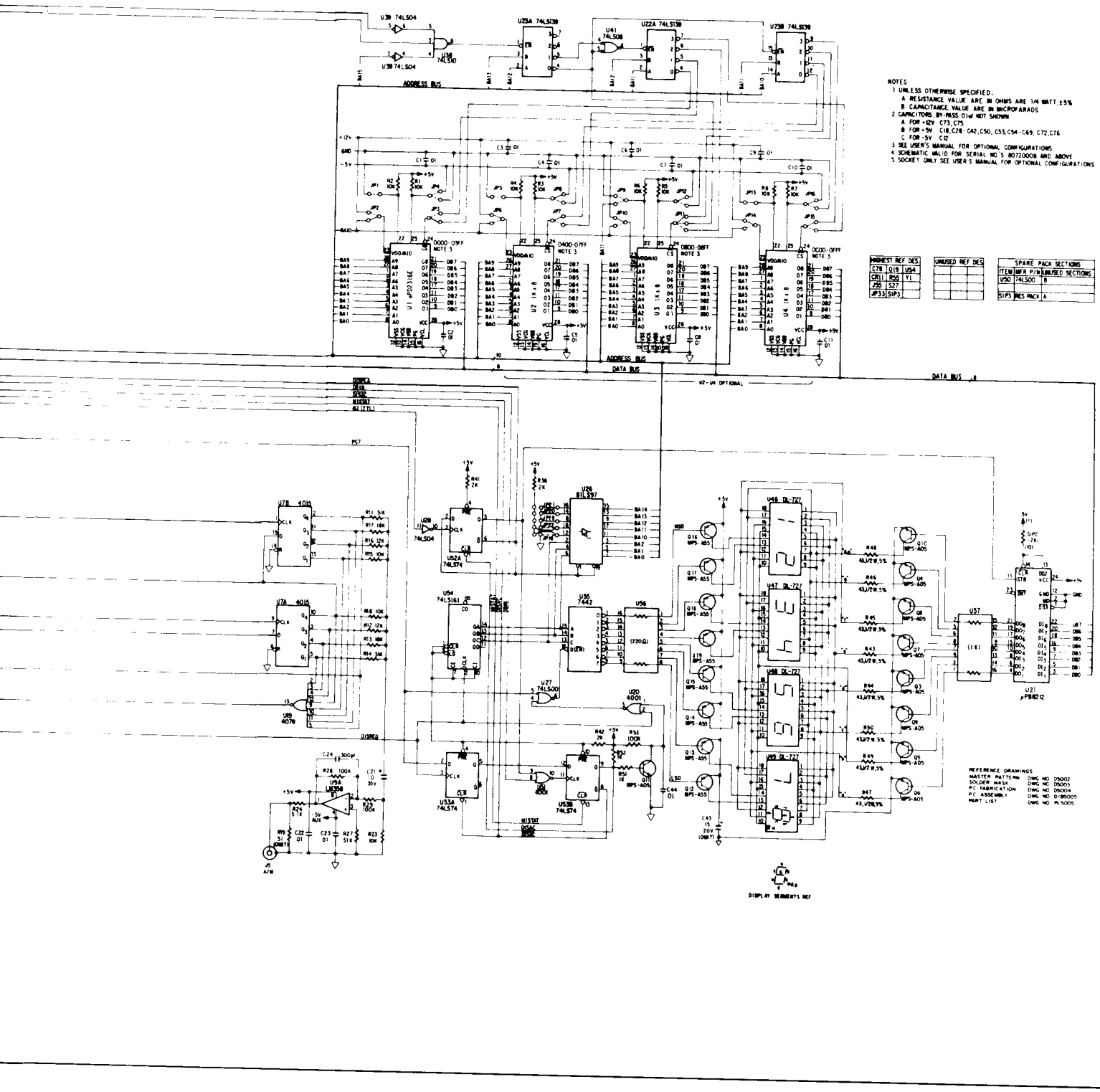
KEY	DISPLAY
8FF8	DTOG @ 8FF8_
ENTER	DTOG @ 8FF8 = _

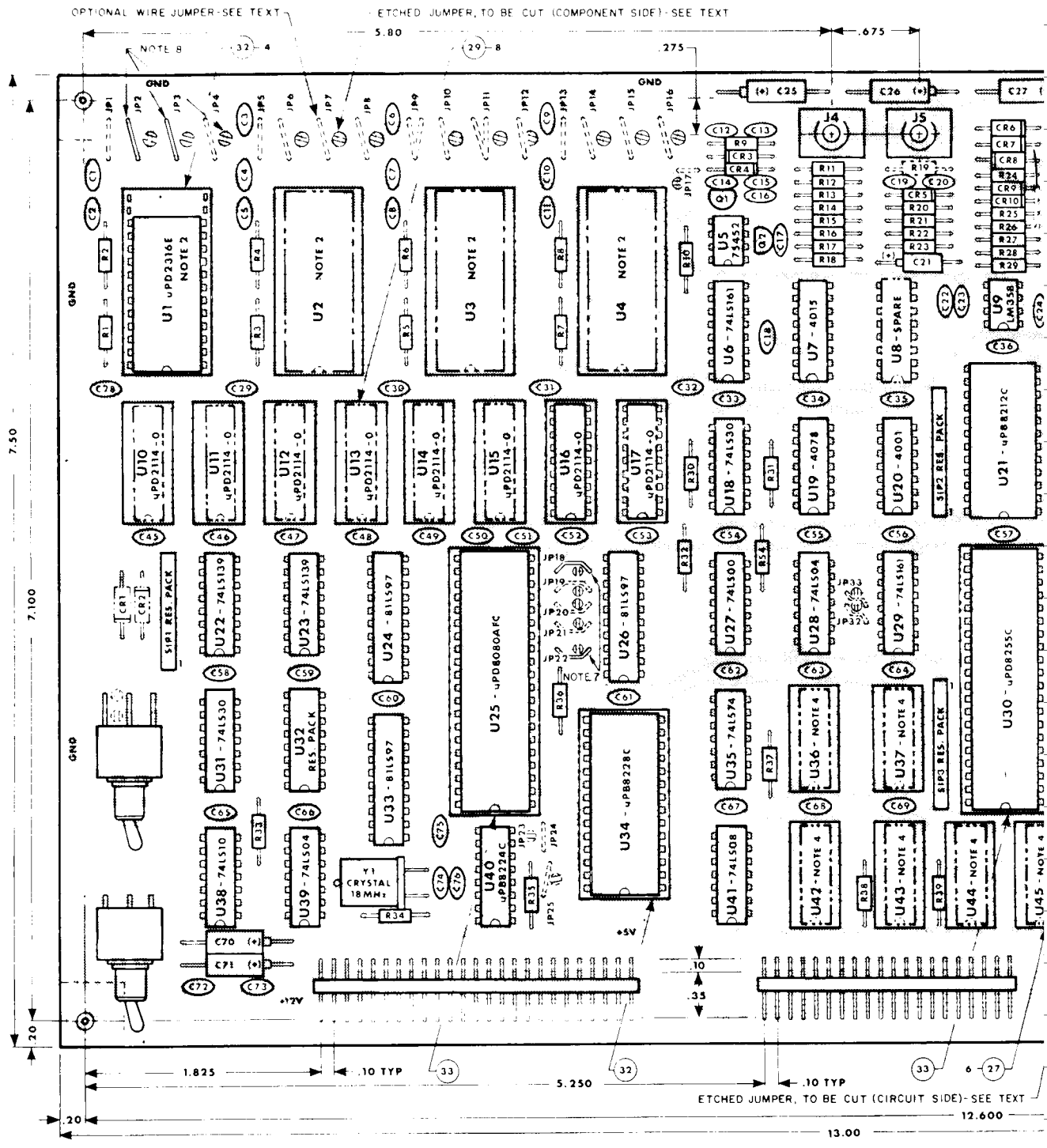
Now it is prompting you for the data to write at this address before it is toggled. To light the top segment of the first digit of the UUT display press:

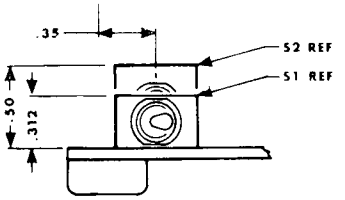
KEY	DISPLAY
1	DTOG @ 8FF8 = 1_
ENTER	DTOG @ 8FF8 = 1 BIT _

Next you have to enter which of the eight data bits you want to toggle. To toggle the same top segment, toggle bit 0.




NOTES
1 UNLESS OTHERWISE SPECIFIED:
A RESISTANCE VALUE ARE IN OHMS ARE 1/4 WATT, ±5%
B CAPACITANCE VALUE ARE IN MICROFARADS
2 CAPACITORS BY-PASS 01uf NOT SHOWN
A FOR +12V C73, C75
B FOR +5V C18, C28 - C42, C50, C53, C54 - C69, C72, C76
C FOR -5V C12
3 SEE USER'S MANUAL FOR OPTIONAL CONFIGURATIONS
4 SCHEMATIC VALID FOR SERIAL NO'S 80720008 AND ABOVE
5 SOCKET ONLY SEE USER'S MANUAL FOR OPTIONAL CONFIGURATIONS
HIGHEST REF DES
C78 Q19 U54
CR11 R66 Y1
J55 S27
JP33 SIP3
UNUSED REF DES
SPARE PACK SECTIONS
ITEM MFR P/N UNUSED SECTIONS
U50 74LS00 B
SIP3 RES PACK A
REFERENCE DRAWINGS
MASTER PATTERN DWG NO D50002
SOLDER MASK DWG NO D50003
P.C. FABRICATION DWG NO D50004
P.C. ASSEMBLY DWG NO D1850005
PART LIST DWG NO PL5005
ADDRESS BUS
DATA BUS
U39 74LS04
U38 74LS04
U23A 74LS138
U41 74LS08
U22A 74LS138
U23B 74LS138
U1 μPD2316E
U2 1K x 8
U3 1K x 8
U4 1K x 8
U5 1K x 8
U26 81LS97
U27 74LS00
U28 74LS04
U52A 74LS74
U53A 74LS74
U53B 74LS74
U54 74LS161
U55 7442
U56 (220Ω)
U57 (1K)
U20 4001
U21 μPB8212
U7A 4015
U7B 4015
U9 4078
U46 DL-727
U47 DL-727
U48 DL-727
U49 DL-727
U50A LM358
Q1 MPS-A55
Q2 MPS-A55
Q3 MPS-A05
Q4 MPS-A05
Q5 MPS-A05
Q6 MPS-A05
Q7 MPS-A05
Q8 MPS-A05
Q9 MPS-A05
Q10 MPS-A05
Q11 MPS-A05
Q12 MPS-A55
Q13 MPS-A55
Q14 MPS-A55
Q15 MPS-A55
Q16 MPS-A55
Q17 MPS-A55
Q18 MPS-A55
Q19 MPS-A55
DISPLAY SEGMENTS REF



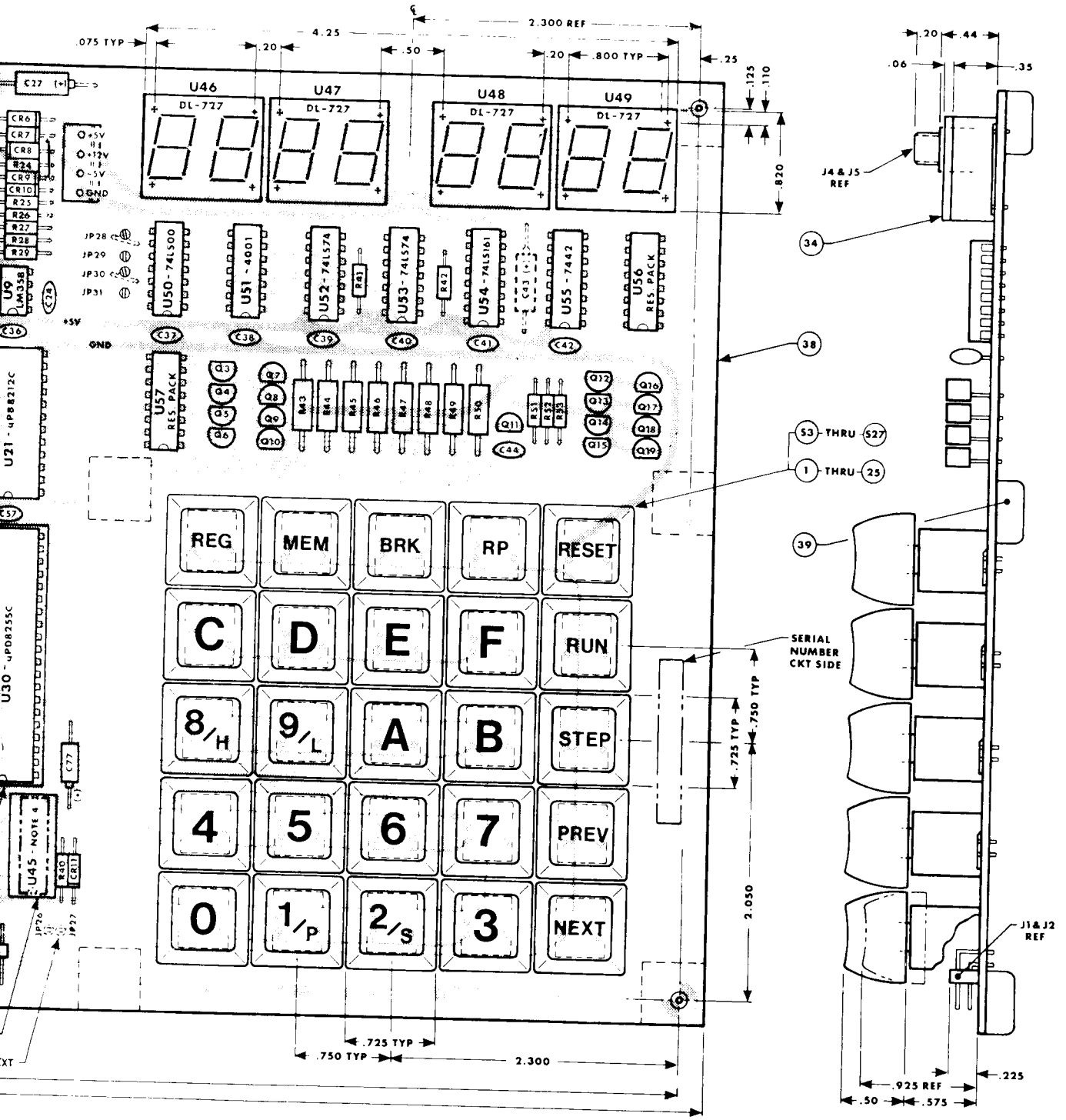


NOTES

1. DO NOT INSTALL R19 AND C43.
2. ASSEMBLE ITEM 32, SOCKET, IN LOCATIONS U1 THRU U4. INSTALL U1, uPD2316E IN ITS SOCKET AND OMIT OPTIONAL uPD458'S. U2, U3 & U4.
3. ASSEMBLE ITEM 29, SOCKET, IN LOCATIONS U10 THRU U17. INSTALL U10 & U11, uPD2114-45'S, IN THEIR SOCKETS AND OMIT THE OPTIONAL uPD2114-45'S, U12 THRU U17.
4. ASSEMBLE ITEM 27 SOCKET, IN LOCATIONS U36, U37 & U42 THRU U45, BUT DO NOT INSTALL THESE OPTIONAL DEVICES - SEE TEXT.
5. U8 IS A SPARE PACK LOCATION.
6. REMOVE PROTECTIVE PLASTIC FROM U46-U49 AFTER SOLDERING AND BEFORE OVEN DRYING.

NOTES

7. CUT JP18 & JP22 ETCHES AND ADD JUMPERS.
8. CUT JP2, JP3 & JP4 ETCHES AND ADD JUMPE JP3 AS SHOWN.



PERS AS SHOWN.
UMPERS JP2 &

REFERENCE DRAWINGS

SCHEMATIC	DWG NO	R/B5001
MASTER PATTERN	DWG NO	D5002
SOLDER MASK	DWG NO	D5003
P.C. FABRICATION	DWG NO	D5004
PARTS LIST	DWG NO	PL5005

KEY	DISPLAY
ENTER	DTOG @ 8FF8 = 1 BIT 0 OK

Again this function goes pretty fast to be seen in the display, so press LOOP. Now the Troubleshooter is writing 01 and 00 to the display address. This is shown in the first digit of the UUT display.

1. What would the value of the data be to turn on only the decimal point of the display? _____

_____

2. What bit would have to be toggled to turn it off?

Enter the values of the two last questions in TOGGLE DATA function and verify that the decimal flashes when you LOOP on the TOGGL DATA function.

TOGGL ADDR will cause the Troubleshooter to toggle a specified address bit from one logic state to the other, much the same way as in the TOGGLE DATA function.

KEY	DISPLAY
TOGGL ADDR	ATOG @ _

Now you enter the address.

KEY	DISPLAY
8FF8	ATOG @ 8FF8_
ENTER	ATOG @ 8FF8 BIT _

Now you enter which address bit you wish to toggle.

KEY	DISPLAY
2	ATOG @ 8FF8 BIT 2 _
ENTER	ATOG @ 8FF8 BIT 2 OK

The Troubleshooter now reads the data at address 8FF8. Then it toggles address bit 2, creating a new address.

3. If $8FF8_{16} = 1000\ 1111\ 1111\ 1000_2$ and bit 2 is toggled, what would the new address be in hexadecimal? _____

_____

The Troubleshooter will then read the data contained at this second address. This operation is hard to show with the UUT alone.

The Troubleshooter 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.

4. What bit of the control word is used for the WAIT line? _____

_____

(See the chart on the POD)

Now enter the information in this manner:

KEY	DISPLAY
TOGGL DATA	DTOG @ _
STS/CTL	DTOG @ CTL = _
0	DTOG @ CTL = 0 _
ENTER	DTOG @ CTL = 0 BIT _
(Ans. Q 12)	DTOG = CTL = 0 BIT (Ans.Q12)_
ENTER	DTOG @ CTL = 0 BIT (Ans.Q12) OK

This function can't be demonstrated without the use of the PROBE.

Probe Operation

Using the PROBE with the Troubleshooter is simple and straight forward. The PROBE will be most useful for the circuitry that is off the bus, like checking I/O's and associated circuits.

Connect the probe to the jack located under the keyboard of the Troubleshooter Note that right next to the jack is a fuse socket. This is to protect the probe ground lead from damage should it be accidentally connected to a voltage source.

One mode of operation for the PROBE is to use it as a logic probe. In this next exercise you will use the PROBE to detect various logic levels at different nodes on the UUT. The following table shows what the different combinations of the indicator lights mean.

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

1. Place the PROBE on pin 7 of U34

What do the indicator lights tell you about this point on the UUT? _____

_____

Now remove the PROBE from U34.

One of the most unique features of the Troubleshooter is the ability to SYNC the probe to a specific time period of the Troubleshooter's operation. For the 8080A POD these periods are when the address or data is valid at the UUT. Simply put, this means that the Probe is only enabled when the address or data information is applied to its specific bus. The probe also has a FREE-RUN mode where the probe is enabled at approximately a 1 kHz rate.

To see what sync mode the Probe is presently in, proceed as follows:

KEY	DISPLAY
SYNC	SYNC MODE (0-F) _
ENTER	SYNC MODE (0-F) FREE-RUN

The 0-F indicates the allowable values that can be used with the sync mode. For the 8080 POD only A for address, D for DATA, and F for FREE RUN are valid sync modes for the PROBE. To change the SYNC mode to DATA press:

KEY	DISPLAY
SYNC	SYNC MODE (0-F) _
D	SYNC MODE (0-F) DATA OK

Now that the probe is synchronized to the DATA cycle, check the activity at pin 7 of U34.

The PROBE now shows no activity at the Node. Once the probe is synchronized to DATA you have to make the Troubleshooter perform a function before it will enable the probe to take a reading.

With your PROBE still on pin 7 of U34 Press:

KEY	DISPLAY
AUTO	AUTO TEST WAIT

This little show lasts about 30 seconds.

One example of using the logic PROBE is to identify which IC is used for a specific address. In this next example you will identify the socket(s) assigned to address 8800. (Remember the UUT uses two 4 Bit chips to handle 8 bit words.)

What you will do is perform a READ @ 8800 and using the Probe identify which socket(s) has the proper chip select signal. First you need to identify which pins on an IC is the chip select pin. One clue that is given is that 8800 is a RAM addres. The next step is to identify which pin on a 2114 is chip select (chip enable).

1. Refering to the reference section of the notebook which pin on a 2114 is chip select? _____

_____

2. What is the proper level on the chip select pin to select a 2114? _____

_____

Now that you have the proper information you need to set up a looping read operation and probe the proper pin of sockets U10 thru U17.

KEY	DISPLAY
READ	READ @ _
8800	READ @ 8800 _
ENTER	READ @ 8800 = FF OK
LOOP	READ @ 8800 FF OK

Without using the schematic to identify the proper RAM sockets for the address space specified use the assembly drawing and the probe the chip select pin of U10 thru U17. Find the sockets that have the proper level on the chip select pin.

1. Which sockets are being enabled? _____

_____

Another way of utilizing the PROBE is to use it as a pulser. You can use the pulser to stimulate a node to an operator specified state. HIGH or LOW and TOGGLING between HIGH and LOW.

WARNING: Always ground the ground lead before using the PROBE as a PULSER otherwise you may damage the Troubleshooter circuitry.

In this next example you will use the PROBE to stimulate a DATA Bus line while the Troubleshooter is performing a READ operation at some address location. First insure that your PROBE is synchronized to DATA as follows:

KEY	DISPLAY
SYNC	SYNC MODE (0-F) _
D	SYNC MODE (0-F) DATA OK

Now put the Troubleshooter into a LOOPING READ operation as follows:

KEY	DISPLAY
READ	READ @ _
FFFF	READ @ FFFF_
ENTER	READ @ FFFF = FF OK
LOOP	READ @ FFFF = FF OK (LOOPING)

Set the PROBE so that it is pulsing LOW by pressing the PULSE LOW key in the lower right hand corner of the keyboard.

In this next step you will place your PROBE on a DATA BUS line and watch the Troubleshooter display for a change in the DATA information.

Place your PROBE on pin 5 of U34.

1. What does the DATA read now at address FFFF?

_____

Place your PROBE on pin 15 of U34.

2. What does the DATA read now? _____

_____

The ability to take SIGNATURES through the PROBE is another feature of the Troubleshooter. The Troubleshooter provides the proper stimulus while the PROBE takes a SIGNATURE at a specified node of the UUT. It's a very simple matter of comparing signatures to one developed on a "known good" UUT to help isolate a failure.

In this next example we will use the Troubleshooter to perform a RAMP at FFFF and then use the PROBE to develop a signature on a couple of the DATA Bus lines.

READ PROBE key is used for this operation and is located on the bottom row of keys fourth key from the right.

First you will have to clear out the old information in the READ PROBE register. This is accomplished by simply pressing READ PROBE. This operation causes the old information to be displayed and clears the register to all zeros.

Place the PROBE on pin 15 of U34 and press:

KEY	DISPLAY
RAMP	RAMP @ _
FFFF	RAMP @ FFFF_
ENTER	RAMP @ FFFF WAIT
After the display =	RAMP @ FFFF OK
READ PROBE	

3. What is the SIGNATURE for a ramp on this data line (DBO)? _____

_____

Now place the PROBE at pin 17 of U34 and press:

KEY	DISPLAY
RAMP	RAMP @ _
ENTER	RAMP @ FFFF WAIT
After the display @	RAMP @ FFF OK
READ PROBE	

4. What is the SIGNATURE for a ramp on this data line (BD1)? _____

_____

TYPE OF REGISTER	REGISTER	FUNCTION
DEDICATED	A	Stores the last bit mask specified by the programmer. Also, if the UUT I/O address descriptors are invoked as default values when the I/O Test is performed, the bit mask specified by the last I/O address descriptor is stored in Register A.
	B	Stores the last ROM signature specified by the programmer. Also, if the UUT ROM address descriptors are invoked as default values when the ROM Test is performed, the ROM signature specified by the last ROM address descriptor is stored in Register B.
	C	Stores the last status/control information specified by the programmer for the Write Control or Toggle Data Control functions, or generated by the 9010A during performance of the Read Status function.
	D	Stores the last bit number (in the range 0-31) specified by the programmer for the Toggle Address, Toggle Data, or Toggle Data Control functions.
	E	Stores the last data specified by the programmer or generated by the 9010A during operation.
	F	Stores the last address specified by the programmer or generated by the 9010A during an operation involving the interface pod.
	0	Stores data accumulated during the Read Probe operation.
NON-DEDICATED	1-9	Non-dedicated registers for sole use by programmer for storage and manipulation of data as specified by the programmer.

NOTES:

1. *Dedicated Registers A through F and 0 are also available to the programmer for storage and manipulation of data.*
2. *Register 0 through 7 are local registers whose values are local to the currently executing program.*
3. *Registers 8 through F are global registers and unaffected by passing between called and calling programs.*

Programming

Programming the Troubleshooter is very easy. Once you know how to use the Troubleshooter in the immediate mode you have most of the knowledge necessary to program.

A program is nothing more than a sequence of Troubleshooter key strokes being performed automatically. Any of the built-in tests or troubleshooting functions may be included in programs, as well as the learn, READ PROBE, and arithmetic operations. In addition, test sequencing keys are available to 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.

There are three operating modes for the Troubleshooter:

IMMEDIATE MODE — actions are performed as soon as they are selected. This is the mode you have been using up to now.

PROGRAMMING MODE — actions are NOT performed but stored as program steps as they are selected.

EXECUTING MODE — actions are performed as specified by the steps in A program that is being executed.

Creating a program is a very simple matter of pressing:

KEY	DISPLAY
PROGM	PROGRAM _

At this point you need to specify a program number and can use from 0 to 99.

KEY	DISPLAY
1	PROGRAM 1_
ENTER	PROGRAM 1 CREATED

Notice that the PROGMING annunciator is flashing, indicating that the Troubleshooter is in the programming mode. Now let's say we want our program to perform a BUS TEST and then an I/O TEST.

KEY	DISPLAY
BUS TEST	BUS TEST
I/O TEST	IO TEST @ _
ENTER	IO TEST

Now that we have entered all the program steps necessary for our program you simply close the program with:

KEY	DISPLAY
PROGM	PROG 1 CLOSED-10187 BYTES LEFT

The message is telling you that the program is closed and that you have 10187_{10} bytes left in memory still unused. Also note that the PROGMING annunciator has stopped flashing. To execute the program:

KEY	DISPLAY
EXEC	EXECUTE PROGRAM _
1	EXECUTE PROGRAM 1 _
ENTER	EXECUTE PROGRAM 1

The EXECUTING annunciator will flash as long as the program is running and will extinguish when the program stops. If there had been any errors they would have been reported in much the same way as in the immediate mode. e.g.

Short pins 15 and 17 on U34. Then execute program 1. When the error occurs LOOP on it and then remove the jumper. Now when you press CONT the looping will stop and the remainder of the program will be executed from the point at which the error occurred. Now remove the jumper.

Now we want to change the program so that we do a ROM test in place of the I/O test and tell us when it is done.

First you have to get back into the programming mode.

KEY	DISPLAY
PROGM	PROGRAM _
1	PROGRAM 1 _
ENTER	PROGRAM 1 OPENED - 5 BYTES

This message is telling you that you have reopened program 1 and that it presently occupies 5 bytes in memory. First we want to delete the I/O test.

KEY	DISPLAY
MORE	BUS TEST
MORE	IO TEST
CLEAR/NO	STEP DELETED

Now the I/O test has been deleted from the program. Now add ROM test.

KEY	DISPLAY
ROM TEST	ROM TEST @ _
ENTER	ROM TEST

Now to add a message to tell the operator that the program is done key in the following:

KEY	DISPLAY
DISPL	DPY _

Now the keyboard becomes an ASCII keyboard for entering display characters. Most keys have an ASCII character to its lower right corner, some exceptions are keys 0 thru 9 and A thru F.

We want the display to read DONE and then beep.

KEY	DISPLAY
D	DPY-D _
>	DPY-DO _
IF	DPY-DON _
E	DPY-DONE _
ROM VIEW	DPY-DONE (BELL) _
ENTER	DPY-DONE (BELL)
PROGM	PROG 1 CLOSED-10180 BYTES LEFT

Now execute program 1.

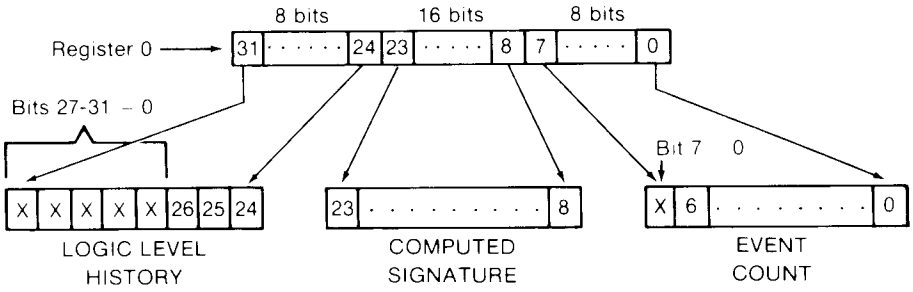
For this next program we want the Troubleshooter to write all 0's to the UUT display digits. The space below is provided to draw a flow chart and program steps for this program. You will work this problem out in class.

The Probe and The Program

Up to this point you have learned the various features of the Troubleshooter each separately. It is now time to put these features together so you can use the Troubleshooter to its fullest capability. The next part of this lab will take you through the steps of using the probe as part of a Troubleshooter program.

You may remember from a previous class discussion that in order to get useful information with the probe you had to do a READ PROBE and then run stimulus by the probe followed by another READ PROBE. Each READ PROBE causes the probe data to be loaded into the display. If you also refer back to the list of data registers in the Troubleshooter you will find that register 0 is a dedicated register for "Storing data accumulated during the READ PROBE operation." It is through register 0 and only register 0 that probe data can be collected by a program. When a READ PROBE is performed data from the probe is transfered into register 0.

The probe response data is stored in bits 0-31 of Register 0 as follows:



Bit 24 = 1 if logic high detected
Bit 25 = 1 if logic tristate (invalid) detected
Bit 26 = 1 if logic low detected

NOTES:

1. Accumulated probe data is placed in Register 0 when the Read Probe step is executed.
2. Signatures may range from 0000 to FFFF.
3. Event Counts may range from 0 to 127. When a count of 127 is reached, the counter begins counting again at 0.

This chart shows how the probe information is stored in register 0.

Put your probe on pin 13 of U34 then press the following keys:

KEY	DISPLAY
SYNC	SYNC MODE (0-F)
D	SYNC MODE (0-F) DATA OK
READ PROBE	PROB-LVL xyz COUNT ccc SIG ssss
RAMP	RAMP @ __
FFFF	RAMP @ FFFF __
ENTER	RAMP @ FFFF WAIT

After the WAIT changes to OK

KEY	DISPLAY
READ PROBE	PROBE-LVL L H COUNT xxx SIG 96EC

Now make a note of the three quantities given in the display, and read out register 0.

KEY	DISPLAY
REG	REG __
0	REG0 = __
ENTER	REG0 = xxxxxxx

Note that the register displays its value in hexadecimal form. The count that you recorded from the previous step was in decimal form. The signature, however, is hexadecimal in both cases. The level history is just a three digit binary word of all three states each digit representing one of the three levels.

To be of any significant value to a program, the desired information will have to be stripped from register 0. To perform this stripping of register 0 you will have to use the arithmetic functions of the Troubleshooter that was dicussed previously.

1. What would have to be done, to transfer only the COUNT value to register 8 without changing register 0?

_____

2. What would have to be done to leave only the SIGNATURE portion of register 0 in register 9, without changing register 0? _____

_____

3. How would you leave only the LEVEL history from register 0 in register A, without changing register 0? __

_____

Now design a program that will perform a READ PROBE and then break the probe data down into its three parts leaving the COUNT in register 8, the SIGNATURE in register 9, and the LEVEL history in register A. Once made this program can be used as a subroutine that will strip out all probe data for you. The choice of registers is due to the fact that data in registers 0-7 is lost when returning to a program from a subroutine.

Now test the stripping routine with another program that simply does a read probe and generates a stimulus that the probe can sense and then display it.

This program is very useful when probe data is used in a program.

Intel® 2114A 1024x4 Bit Static Ram

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Max. Access Time (ns)	100	120	150	200	200	250
Max. Current (mA)	40	40	40	40	70	70

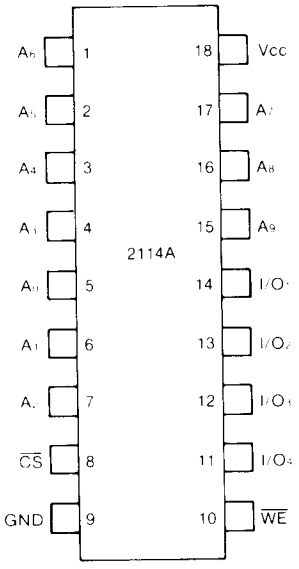
- HMOS Technology
- Low Power, High Speed
- Identical Cycle and Access Times
- Single +5V Supply ±10%
- High Density 18 Pin Package
- Completely Static Memory - No Clock or Timing Strobe Required
- Directly TTL Compatible: All Inputs and Outputs
- Common Data Input and Output Using Three-State Outputs
- 2114 Upgrade

The Intel® 2114A is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using HMOS, a high performance MOS technology. It uses fully DC stable (static) circuitry throughout. In both the array and the decoding, therefore requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

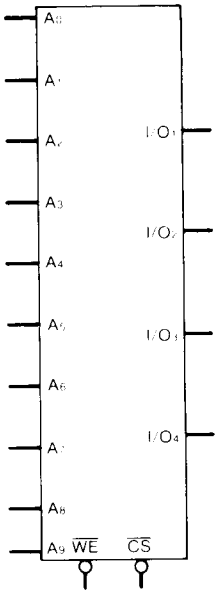
The 2114A is designed for memory applications where the high performance and high reliability of HMOS, low cost, large bit storage, and simple interfacing are important design objectives. The 2114A is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: Inputs, outputs, and a single +5V supply. A separate Chip Select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are or-tied.

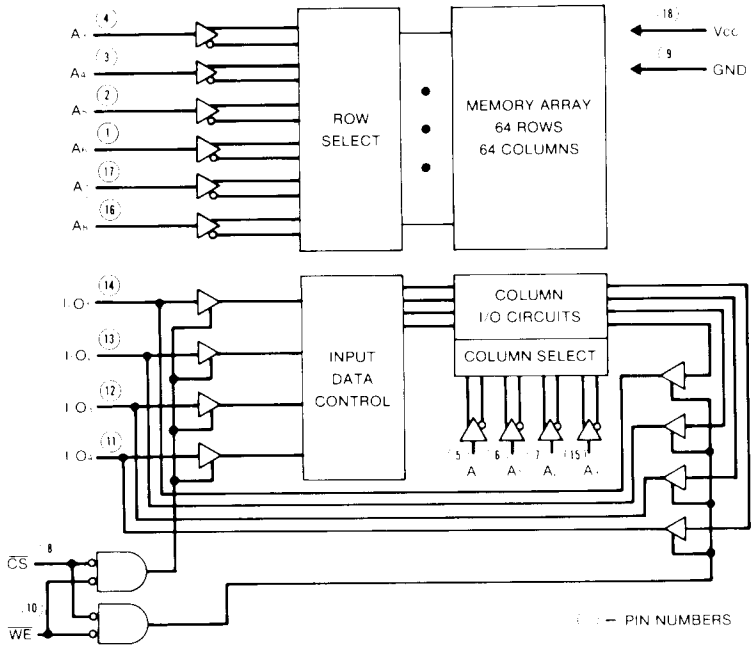
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0-A_9	ADDRESS INPUTS	VCC POWER (+5V)
WE	WRITE ENABLE	GND GROUND
CS	CHIP SELECT	
I/O_1-I/O_4	DATA INPUT/OUTPUT	

A.C. CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 10\%$, unless otherwise noted.

READ CYCLE [1]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{RC}	Read Cycle Time	100		120		150		200		250		ns
t_A	Access Time		100		120		150		200		250	ns
t_{CO}	Chip Selection to Output Valid		70		70		70		70		85	ns
t_{CX}	Chip Selection to Output Active	10		10		10		10		10		ns
t_{OTD}	Output 3-state from Deselection		30		35		40		50		60	ns
t_{OHA}	Output Hold from Address Change	15		15		15		15		15		ns

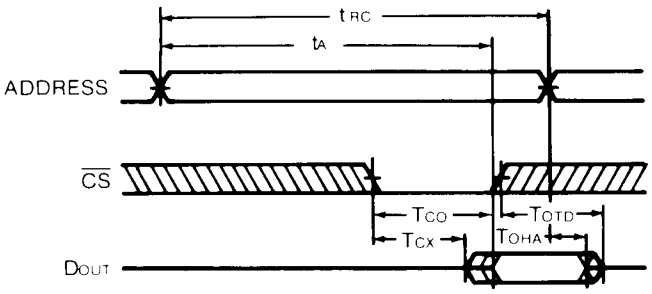
WRITE CYCLE [2]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{WC}	Write Cycle Time	100		120		150		200		250		ns
t_W	Write Time	75		75		90		120		135		ns
t_{WR}	Write Release Time	0		0		0		0		0		ns
t_{OTW}	Output 3-state from Write		30		35		40		50		60	ns
t_{DW}	Data to Write Time Overlap	70		70		90		120		135		ns
t_{DH}	Data Hold from Write Time	0		0		0		0		0		ns

NOTES:

1. A Read occurs during the overlap of a low $\overline{CS}$ and a high $\overline{WE}$.
2. A Write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$. t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to the earlier of $\overline{CS}$ or $\overline{WE}$ going high.

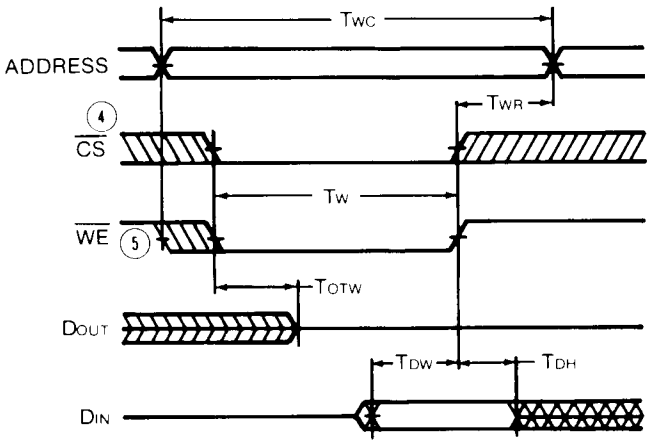
WAVEFORMS
READ CYCLE (3)



NOTES:

3. $\overline{WE}$ is high for a Read Cycle.
4. If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transition, the output buffers remain in a high impedance state.
5. $\overline{WE}$ must be high during all address transitions.

WRITE CYCLE



Intel® 2716 16K (2K x 8) Erasable Prom

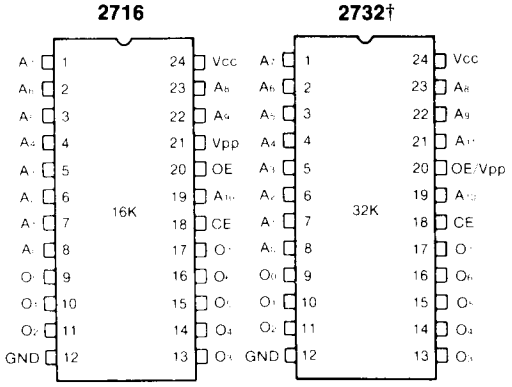
- Fast Access Time
  - 350 ns Max. 2716-1
  - 390 ns Max. 2716-2
  - 450 ns Max. 2716
  - 490 ns Max. 2716-5
  - 650 ns Max. 2716-6
- Single +5V Power Supply
- Low Power Dissipation
  - 525 mW Max. Active Power
  - 132 mW Max. Standby Power
- Pin Compatible to Intel® 2732 EPROM
- Simple Programming Requirements
  - Single Location Programming
  - Programs with One 50 ms Pulse
- Inputs and Outputs TTL Compatible during Read and Program
- Completely Static

The Intel® 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features last single address location programming. It makes designing with EPROMs faster, easier and more economical.

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with the newer high performance +5V microprocessors such as Intel's 8085 and 8086. A selected 2716-5 and 2716-6 is available for slower speed applications. The 2716 is also the first EPROM with a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 has the simplest and fastest method yet devised for programming EPROMs — single pulse TTL level programming. No need for high voltage pulsing because all programming controls are handled by TTL signals. Program any location at any time — either individually, sequentially or at random, with the 2716s single address location programming. Total programming time for all 16,384 bits is only 100 seconds.

PIN CONFIGURATION



†Refer to 2732 data sheet for specifications

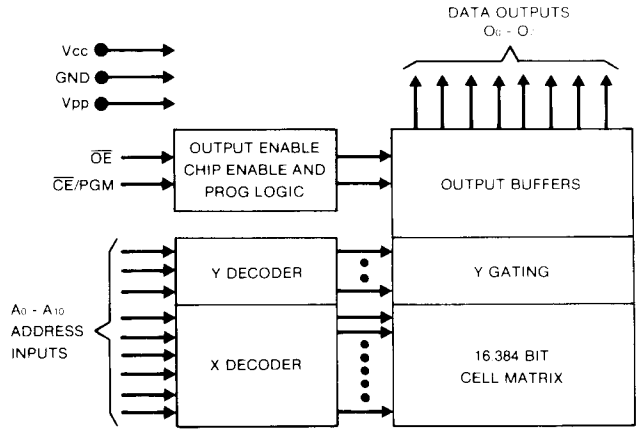
Mode Selection

Mode \ Pins	CE/PGM (18)	OE (20)	VPP (21)	VCC (24)	Outputs (9-11, 13-17)
Read	VIL	VIL	+5	+5	DOUT
Standby	VIH	Don't Care	+5	+5	High Z
Program	Pulsed VIL to VIH	VIH	+25	+5	DIN
Program Verify	VIL	VIL	+25	+5	DOUT
Program Inhibit	VIL	VIH	+25	+5	High Z

PIN NAMES

A0-A10	ADDRESSES
$\overline{CE}$/PGM	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
O8 - O2	OUTPUTS

BLOCK DIAGRAM



Erasure Characteristics

The erasure characteristics of the 2716 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (A). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000A range. Data show that constant exposure to room level fluorescent lighting could erse the typical 2716 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions to extended periods of time, opaque labels are available from Intel which should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog PROM/ROM Programming Instruction Section) for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (A). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

Device Operation

The five modes of operation of the 2716 are listed in Table 1. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other two modes.

Table 1. Mode Selection

Mode \ Pins	CE/PGM (18)	OE (20)	V_{PP} (21)	V_{CC} (24)	Outputs (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

Read Mode

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs 120 ns (t_{OE}) after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

Standby Mode

The 2716 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output or Tieing

Because 2716s are usually used in larger memory arrays, Intel has provided a 2 line control function that accomodates this use of multiple memory connections. The two line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 18) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

Programming

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{OE}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active high, TTL program pulse is applied to the $\overline{CE}$/PGM input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the $\overline{CE}$-PGM input.

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the $\overline{CE}$/PGM input programs the paralleled 2716s.

Program Inhibit

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for $\overline{CE}$/PGM, all like inputs (including $\overline{OE}$) of the parallel 2716s may be common. A TTL level program pulse applied to a 2716s $\overline{CE}$/PGM input with V_{PP} at 25V will program that 2716. A low level $\overline{CE}$/PGM input inhibits the other 2716 from being programmed.

Program Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

A.C. Characteristics

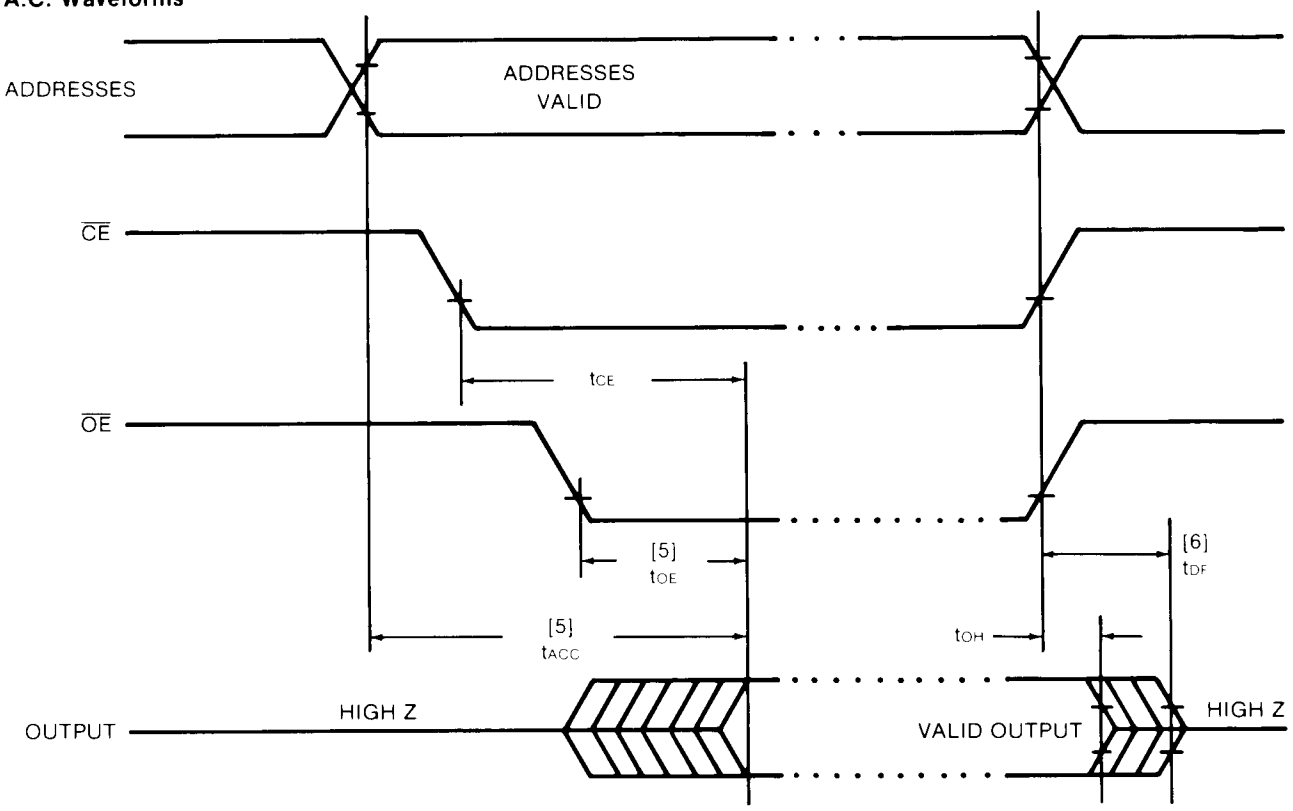
Symbol	Parameter	Limits (ns)										Test Conditions
		2716		2716-1		2716-2		2716-5		2716-6		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Output Delay		450		350		390		450		450	CE = OE = V_{IL}
t_{CE}	CE to Output Delay		450		350		390		490		650	OE = V_{IL}
t_{OE}	Output Enable to Output Delay		120		120		120		160		200	CE = V_{IL}
t_{DF}	Output Enable High to Output Float	0	100	0	100	0	100	0	100	0	100	CE = V_{IL}
t_{OH}	Output Hold from Addresses, CE or OE Whichever Occurred First	0		0		0		0		0		CE - OE = V_{IL}

Capacitance [4] T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

A.C. Test Conditions:
Output Load: 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times: ≤20 ns
Input Pulse Levels: 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs 1V and 2V
Outputs 0.8V and 2V

A.C. Waveforms [1]



NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for T_A = 25°C and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested.
5. $\overline{OE}$ may be delayed up to t_{ACC} - T_{OE} after the falling edge of $\overline{CE}$ without impact on t_{ACC}.
6. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$, whichever occurs first.