FLUKE ®

CUSTOMER SERVICE

9000A-8085 INTERFACE POD TEST FIXTURE

PREPARED BY ED FERGUSON

CUSTOMER SERVICE ENGINEERING

JOHN FLUKE MFG. CO., INC.

INTRODUCTION

The Fluke Customer Service 9000 series test fixtures will verify proper operation of 9000 interface pods. Accompanying test software will exercise the pod and identify faulty functions and lines. A separate test fixture and program is required for each pod type. Each test fixture consists of test points for all UUT cable lines, a ROM to execute a 'RUN UUT' program, and a divider circuit to simulate power supply faults. Once the software has identified a faulty line, a technican familiar with the pod theory may use the 9010A's troubleshooting functions to locate the cause.

The test program utilizes the 9010A and probe to verify proper activity at all test test points in both a NORMAL and 'RUN UUT' mode. One hand operation is allowed with software that senses when the probe is in place, stimulates the test point, takes a reading, and compares the result with the expected result. Input lines are stimulated by jumpering a test point high or low. The software will optionally loop on a failure to allow probing back thru the pod circuitry. A complete pod test takes under ten minutes to complete.

OPERATION

Plug the test fixture into the pod self test socket and the UUT cable into the fixture socket. Load the 8085 pod tape and execute program 0. A menu will appear allowing selection of either the 'NORMAL' or the 'RUN UUT' tests. Follow the displayed test instructions to probe or jumper the fixture test points. A pass is indicated with a single beep and brief display message such as :

```
TP 17 LOGIC LVL HXL = HXL PASS
```

A failure is indicated with three beeps and a display message such as :

```
TP 17 LOGIC LVL HXL= H FAIL LOOP?
```

The operator may loop on the failure by pressing YES or LOOP. When looping on a failure a beep will indicate a pass condition, allowing intermittents to be traced without watching the 9010 display. Press CONT to exit the loop and continue to the next test. In addition to faults detected by the test program, the 9010A will interrupt and report any time that it's software detects a failure. Note however that the test program has disabled certain UUT system errors with the set up commands. Refer to the program listings for set up information.

NOTE

A 'POD TIMEOUT-ATTEMPTING RESET' error message indicates an inoperative pod and will not allow the program to run. Refer to section 5 of the pod manual to troubleshoot an inoperative pod.

NORMAL TEST

The 'NORMAL' test is divided into 13 sub tests. Upon selection of this test, the starting sub test number (1-13) must be entered. This allows branching to a specific routine during troubleshooting. The tests are sequenced to find major faults early. If the condition of the pod is unknown begin with sub test 1; the remaining tests will automatically follow in sequence.

SUB TEST 1 - POWER SUPPLY CHECK
The probe is used to check the presence of the +5 volt supply.

SUB TEST 2 - CLOCK CHECK
The probe is used to verify the crystal input lines are toggling.

SUB TEST 3 - STATUS CHECK
All status lines are probed for proper inactive levels.

SUB TEST 4 - READ STATUS TEST
The status lines are read by the pod for proper inactive levels. Each status line is then jumpered to the active state and read by the pod.

SUB TEST 5 - POWER SUPPLY STATUS TEST
Power supply status is read by the pod and checked for a no-fault condition. Divider switch S1 is then pressed and status is checked for a fault condition.

SUB TEST 6 - CONTROL CHECK
Each contol line is read by the probe for proper levels.

SUB TEST 7 - WRITE CONTROL TEST
User writable control lines are toggled in sequence and verified with the probe for proper levels.

SUB TEST 8 - ADDRESS TOGGLE TEST
Each address line is toggled in sequence and verified with the probe for proper levels.

SUB TEST 9 - DATA TOGGLE TEST
Each data line is toggled in sequence and verified with the probe for proper levels.

SUB TEST 10 - BUS TEST
A bus test is executed.

SUB TEST 11 - READ DATA TEST
Data is read at address FFFF and checked for FF.
Data is read at address 0002 (ROM) and checked for 00.

SUB TEST 12 - SIGNATURE STABILITY TEST
Data is ramped at FFFF and a signature is gathered at data line A0.
The test is repeated three times to verify a stable signature of 96EC.

SUB TEST 13 - TEST FIXTURE ROM TEST (8085 FIXTURE ROM VER 1.1)
A ROM test is executed from 0 - 47F and signature 3D1A is verified.
At the completion of sub test 13 the test menu is displayed again.

RUN UUT TEST

The 'RUN UUT ' test executes a program in the fixture ROM that toggles a certain set of address lines. Each of the five interrupt lines are asserted in sequence, causing an additional address line to toggle. The address line is probed to verify the interrupt took place. Finally the HOLD function is tested. Refer to the fixture theory of operation for description of the ROM program.

The 'RUN UUT' test is divided into 4 sub tests. No provision is made to branch to a particular sub test because the outcome of some tests are dependent on previous test conditions.

SUB TEST 1 - CONTROL TESTS

The 9010A program places the pod in the 'RUN UUT' mode. A reset is performed and the fixture ROM executes the program at address 0. All control lines are probed for proper activity.

SUB TEST 2 - ADDRESS TESTS

All address lines are probed for proper activity as defined by the fixture ROM program.

NOTE: An 8085 'Run UUT' data test is not necessary because the data and low order address lines are multiplexed.

SUB TEST 3 - INTERRUPT TESTS

The INT line is tied high while INTA is probed for an acknowledge. Address line A12 is then probed for proper activity as defined by the fixture ROM program. The TRAP, RST 5.5, RST 6.5, and RST 7.5 interrupts are performed in sequence and verified by probing A11, A13, and A14 respectively for proper activity as defined by the fixture ROM program. The SOD line is then jumpered to the SID line and a RST 7.5 interrupt is performed. A15 will toggle and the SOD line will toggle if the serial lines are functioning.

SUB TEST 4 - HOLD TEST

The HOLD line is tied high and HLDA is probed for an acknowledge. At the completion of sub test 4 the test menu is displayed again.

FIXTURE THEORY OF OPERATION

The test fixture receives power and clock signals from the pod self test socket. No other connections to the self test socket are made. A divider and switch for the supply allows low line fault testing. S1 reduces the + 5 volt supply to + 4.5 volts. TP 41 is tied to + 5 volts through a 20 ohm resistor to provide a logic high level for stimulus of other test points through a jumper. TP 20 is used to tie other test points low.

Test points 1 - 40 allow access to all lines of the pod UUT cable for probing or stimulus as required. All status lines are tied to their inactive state with Z1, R4 and R5. Because the 8085 data and eight low order address lines are multiplexed, octal latch U2 is provided to latch the eight low order address bits for ROM U1 when the ALE line is high. U1 is a tristate octal buffer to place an RST2 (D7) interrupt vector address on the data bus to cause program execution to start at location 10 when an interrupt is received (INTA low). Address lines A0 - A10 are used to address the ROM U1. A11 must be low to select the ROM.

ROM U1 contains a program to test the 'RUN UUT' function. A low on the RESET IN line will cause the program to start at address location 0, set the restart mask, enable interrupts, and loop on address line A10.

A high on each interrupt line will cause a different routine to execute that toggles a different address line. Activity on the address line is checked with the probe to verify the interrupt took place.

A high on the INT line will cause the INTA line to go low, dumping the interrupt vector address onto the data bus, and vector the 8085 to a routine toggling address line A12.

A high on the TRAP line causes address line A11 to toggle.

A high on the RST 5.5 line causes address line A13 to toggle.

A high on the RST 6.5 line causes address line A14 to toggle.

A high on the RST 7.5 line causes address line A15 to toggle,
and the SOD line will toggle if connected to the SID line.

SOFTWARE DESCRIPTION

The test software consists of 18 programs, 2 of which are the 'NORMAL' and 'RUN UUT' tests for a particular pod. The remaining 16 programs are subroutines common to all fixtures. The program functions are outlined below. Refer to the program listings for detailed descriptions.

PROGRAM 0 is a menu to select either the 'NORMAL' or 'RUN UUT' tests.

PROGRAM 1 performs a read probe.

PROGRAM 2 toggles the address bit specified in REG D four times and performs a read probe.

PROGRAM 3 toggles the data bit specified in REG D four times and performs a read probe.

PROGRAM 4 toggles the control bit specified in REG D four times and performs a read probe.

PROGRAM 5 performs a read probe after a 1/4 second delay.

PROGRAM 6 ramps data at FFFF and performs a read probe. The signature is compared to the expected (REG A). This is performed three times to verify stable signatures.

PROGRAM 90 performs a read operation at the location specified in REG 3. Expected data is specified in REG 2. Program exits if expected data equals the actual, else the operator may branch to a loop - on - fail routine.

PROGRAM 91 performs a read status and displays the actual (REG C) and expected (REG A) levels.

PROGRAM 92 performs a status read operation at the test point specified in REG 9. Operator is instructed to place jumpers or press buttons as specified in REG 8. Program exits if expected status equals the actual, else the operator may branch to a loop-on-fail routine.

PROGRAM 93 calls program 1 to perform a read probe, then decodes the the probe history in REG C into level, count, or signature information as specified in REG 8. The expected and decoded probe history is displayed.

PROGRAM 94 selects the sync mode specified in REG 8 and calls PROG 93 to perform a read probe and display the history at the test point specified in REG 9. The program exits if expected history equals the actual, else the operator may branch to a loop-on-fail routine.

PROGRAM 95 detects when the probe has been removed from the test point.

PROGRAM 96 detects when the probe has been placed on a test point. If a valid level is not detected within 4 seconds the program will timeout.

PROGRAM 97 provides a one second delay.

PROGRAM 98 provides a 1/4 second delay.

PROGRAM 64 is the 'NORMAL' test for the 8085 pod. The starting sub test is selected and the program branches to the appropriate label. REG 8 is encoded with the test information as outlined in the REGISTER DECODING charts shown in the next section. The appropriate subroutine (program 90, 92, or 94) is called for read data, read status, or read probe operations respectively. Refer to the program listings for test discriptions.

PROGRAM 65 is the 'RUN UUT' test for the 8085 pod. The pod is placed in the 'RUN UUT' mode and a reset is performed to run the ROM program. REG 8 is encoded with test information as outlined in the REGISTER 8 DECODING charts shown in the next section. The appropriate subroutine (program 90,92, or 94) is called for read data, read status, or read probe operations respectively. Refer to the program listings for test discriptions.

REGISTER 8 ENCODING

(1) REGISTER 8 ENCODING FOR DATA READS - PROGRAM 90

```
                READ ADDRESS          DATA
                bits  23 - 8          7 - 0
           \____________________/  \__________/
0000 0000 XXXX XXXX XXXX XXXX       XXXX XXXX
               ( 0 - FFFF )         ( 0 - FF )
```

EXAMPLE : REG 8 = 00FFFFFF, CALL PROGRAM 90

```
PERFORM READ @ FFFF
EXPECTED DATA = FF
```

(2) REGISTER 8 ENCODING FOR STATUS READS - PROGRAM 92

```
            STATUS BIT MASK   PASS      SWITCH        TIE  TP       TEST POINT
              bits 23-12       11       10 - 9         8 - 7          5 - 0
           \________________/  \/        \__/           \__/        \________/
0000 0000 XXXX XXXX XXXX        X         XX             XX         0XX  XXXX
             ( 0 - 4095 )                                           ( 0 - 63 )

                             0 = LO   00 = NO PUSH   00 = DO NOT TIE TP
                             1 = HI   01 = PUSH S1   01 = TIE TP LOW
                                      10 = PUSH S2   11 = TIE TP HI
                                      11 = PUSH S3
```

EXAMPLE: REG8 = 00010999 , CALL PROG 92

```
Test point = 25
Tie TP 25 high
Do not push button
Pass if status reads high
Status bit mask = 000000010000
```

(3) REGISTER 8 ENCODING FOR PROBE HISTORY - PROGRAM 94

```
     Expected signature, count,       Sync &     Stimulus       Test point
     or level history.                read.      Program #

            bits  31 - 16             15 - 12     11 - 6          5 - 0
     \_________________________/     \____/     \________/     \________/
SIG  XXXX    XXXX    XXXX    XXXX      XXXX       XXXX  XX       XX  XXXX
                                                 ( 0 - 63 )     ( 0 - 63 )
HIST 0000    0000    0000    01xh

CONT 0XXX    XXXX    0XXX    XXXX      0001 = freerun - level
      \________/      \________/       0010 = freerun - count
      MIN COUNT       MAX COUNT        0100 = address - signature
      ( 0-127 )       ( 0-127 )        0101 = address - level
                                       0110 = address - count
                                       1000 = data - signature
                                       1001 = data - level
                                       1010 = data - count
```

EXAMPLE: REG8 = 00051081 , CALL PROGRAM 94

```
Test point = 1
Stimulus program = 2
Sync = freerun
Read = level history
Expected level history = LH
```

```
***********************************************************
***********************************************************
***                                                     ***
***   TITLE:    FLUKE 9000A 8085 INTERFACE POD TESTS    ***
***   VERSION:  REV 1.0     MAR 25 1982                 ***
***   AUTHOR:   ED FERGUSON                             ***
***             CUSTOMER SERVICE ENGINEERING            ***
***             JOHN FLUKE MFG. CO., INC.               ***
***                                                     ***
***********************************************************
***********************************************************
```

PROPRIETARY NOTICE

```
SET UP COMMANDS

   TRAP BAD PWR SUPPLY ? NO             TRAP ILLEGAL ADDR ? YES
   TRAP ACTIVE INTERRUPT ? NO           TRAP ACTIVE FORCE LINE ? YES
   TRAP CTL ERR ? YES                   TRAP ADDR ERR ? YES
   TRAP DATA ERR ? YES                  ENABLE READY ? NO
   ENABLE HOLD ? NO                     BUS TEST @ FFFF
   RUN UUT @ 0000                       TIMEOUT 200
   EXERCISE ERRORS ? YES                BEEP ON ERR TRANSITION ? YES
   STALL 13                             UNSTALL 11
   NEWLINE 00000D0A                     LINESIZE 79


PROGRAM 0  MENU

   DPY *** 8085 POD TESTS
   DPY-+ REV 1.0 ***#
   EXECUTE PROGRAM 97
   DPY- *** FLUKE CUSTOMER
   DPY-+ SERVICE ***#
   EXECUTE PROGRAM 97
0: LABEL 0
   DPY-TEST? 1-8085 NORM
   DPY-+ 2-8085 RUN UUT
1: LABEL 1
   DPY-+#
   REG1 = 40
   DPY-+%1
2: LABEL 2
   IF REG1 = 40 GOTO 2
   IF REG1 = 1 GOTO 3
   IF REG1 = 2 GOTO 4
   GOTO 1
3: LABEL 3
   EXECUTE PROGRAM 64
   GOTO 0
4: LABEL 4
   EXECUTE PROGRAM 65
   GOTO 0
```

PROGRAM 1 READ PROBE; NO DELAY

```
   READ PROBE                                    CLEAR PROBE
   READ PROBE                                    READ LOGIC HISTORY
   REGC = REGO                                   ASSIGN HISTORY TO GLOBAL REG C
```

PROGRAM 2 ADDRESS TOGGLE

```
   READ PROBE                                    CLEAR PROBE
   ATOG @ 0 BIT REGD REPT REPT REPT              TOGGLE ADDR BIT(REG D) 4 TIMES
   READ PROBE                                    READ LOGIC HISTORY
   REGC = REGO                                   ASSIGN HISTORY TO GLOBAL REG C
```

PROGRAM 3 DATA TOGGLE

```
   READ PROBE                                    CLEAR PROBE
   DTOG @ FFFF = FF BIT REGD REPT REPT REPT      TOGGLE DATA BIT(REG D) 4 TIMES
   READ PROBE                                    READ LOGIC HISTORY
   REGC = REGO                                   ASSIGN HISTORY TO GLOBAL REG C
```

PROGRAM 4 CONTROL TOGGLE

```
   SYNC FREE-RUN
   READ PROBE                                    CLEAR PROBE
   DTOG @ CTL = 00000000 BIT REGD REPT REPT REPT TOGGLE CTL BIT(REG D) 4 TIMES
   READ PROBE                                    READ LOGIC HISTORY
   REGC = REGO                                   ASSIGN HISTORY TO GLOBAL REG C
```

PROGRAM 5 READ PROBE; 1/4 SECOND DELAY

```
   READ PROBE                                    CLEAR PROBE
   EXECUTE PROGRAM 98                            DELAY 1/4 SECOND
   READ PROBE                                    READ LOGIC HISTORY
   REGC = REG 0                                  ASSIGN HISTORY TO GLOBAL REG C
```

PROGRAM 6 SIGNATURE STABILITY

```
   REG1 = 3                                      INITIAIZE LOOP COUNTER
0: LABEL 0                                       BEGIN LOOP
   READ PROBE                                    CLEAR PROBE
   RAMP @ FFFF                                   RAMP DATA AT ADDRESS FFFF
   READ PROBE                                    READ PROBE HISTORY
   REGC = REGO                                   ASSIGN HISTORY TO GLOBAL REG C
   REGO = REGO SHR SHR SHR SHR
   REGO = REGO SHR SHR SHR SHR AND FFFF          ACTUAL SIGNATURE
   IF REGA = REGO GOTO 1                         EXPECTED = ACTUAL; LOOP AGAIN
   GOTO 2                                        EXPECTED <> ACTUAL; EXIT
1: LABEL 1
   DEC REG 1                                     DECREMENT LOOP COUNTER
   IF REG1 > 0 GOTO 0                            3 LOOPS
2: LABEL 2                                       EXIT
```

```
PROGRAM 90   DATA TEST

   REG2 = REG8 AND FF                   EXPECTED DATA (REG 2)
   REG3 = REG8 SHR SHR SHR SHR
   REG3 = REG3 SHR SHR SHR SHR
   REG3 = REG3 AND FFFF                 READ ADDRESS (REG 3)
   READ @ REG3                          READ DATA
   DPY-READ DATA $2=$E                  EXPECTED DATA = ACTUAL DATA
   IF REG2 = REGE GOTO 6                BRANCH PASS
   DPY-+ FAIL LOOP?#                    FAIL;LOOP?
   EXECUTE PROGRAM 98                   DELAY
   DPY-+#                               BEEP
   EXECUTE PROGRAM 98                   DELAY
0: LABEL 0
   DPY-+#                               BEEP
   REG1 = 40                            NO KEYS THIS VALUE
   DPY-+%1                              ENABLE INPUT
1: LABEL 1                              SELECT OPTION ENTRY
   IF REG1 = 40 GOTO 1                  LOOP UNTIL INPUT
   IF REG1 = 1C GOTO 2                  PRESSED 'YES'
   IF REG1 = 27 GOTO 2                  PRESSED 'LOOP'
   IF REG1 = 1D GOTO F                  PRESSED 'NO'
   IF REG1 = 25 GOTO F                  PRESSED 'CONTINUE'
   GOTO 0                               PRESSED INVALID KEY
2: LABEL 2                              LOOP ENTRY
   REGB = 40                            NO KEYS THIS VALUE
   DPY-+%B                              ENABLE INPUT
3: LABEL 3
   READ @ REG3                          READ DATA
   DPY-READ DATA $2=$E                  EXPECTED DATA = ACTUAL DATA
   IF REG2 = REGE GOTO 4                BRANCH PASS
   DPY-+ FAIL                           FAIL
   GOTO 5                               BRANCH CHECK KEY
4: LABEL 4                              PASS ENTRY
   DPY-+ PASS#                          PASS
5: LABEL 5                              CHECK KEY
   IF REGB = 40 GOTO 3                  LOOP UNTILL CONT PRESSED
   IF REGB = 25 GOTO F                  PRESSED CONT;BRANCH EXIT
   DPY-+#                               BEEP
   GOTO 2                               PRESSED INVALID KEY
6: LABEL 6                              PASS ENTRY
   DPY-+ PASS#                          PASS
   EXECUTE PROGRAM 98                   DELAY
F: LABEL F                              EXIT
```

PROGRAM 91   STATUS READER

```
   READ @ STS REPT                      READ STATUS
   REGC = REGC AND FFF                  ACTUAL STATUS 12 LINES (REG C)
   REGA = REG8 SHR SHR SHR SHR
   REGA = REGA SHR SHR SHR SHR
   REGA = REGA SHR SHR SHR SHR
   REGA = REGA AND FFF                  EXPECTED STATUS (REG A)
   IF REG9 > 0 GOTO 0                   BRANCH DISPLAY TEST POINT
   DPY-POWER                            POWER SUPPLY STATUS
   GOTO 1                               BRANCH DISPLAY STATUS
0: LABEL 0                              TEST POINT ENTRY
   DPY-TP@9                             DISPLAY TEST POINT (REG 9)
1: LABEL 1                              DISPLAY STATUS ENTRY
   IF REG8 AND 800 = 800 GOTO 2         EXPECTING HIGH STATUS
   CPL REGC                             EXPECTING LOW ;COMPLEMENT ACTUAL STATUS
   REGC = REGC AND FFF                  12 STATUS LINES (REG C)
   DPY-+ STATUS LOW=                    EXPECTING LOW STATUS
   GOTO 3                               BRANCH DISPLAY ACTUAL STATUS
2: LABEL 2                              EXPECTING HIGH STATUS ENTRY
   DPY-+ STATUS HIGH=                   EXPECTING HIGH STATUS
3: LABEL 3                              DISPLAY ACTUAL STATUS ENTRY
   IF REGA AND REGC = REGA GOTO 5       EXPECTED STATUS=ACTUAL; BRANCH PASS
   IF REG8 AND 800 = 800 GOTO 4         EXPECTED HIGH STATUS; BRANCH FAIL LOW
   DPY-+HIGH FAIL                       EXPECTED LOW STATUS; FAIL HIGH
   GOTO F                               BRANCH EXIT
4: LABEL 4                              FAIL LOW ENTRY
   DPY-+LOW FAIL                        FAIL LOW STATUS
   GOTO F                               BRANCH EXIT
5: LABEL 5                              PASS STATUS ENTRY
   IF REG8 AND 800 = 800 GOTO 6         BRANCH;EXPECTED A HIGH
   DPY-+LOW PASS#                       PASS LOW
   GOTO F                               BRANCH EXIT
6: LABEL 6                              PASS HIGH ENTRY
   DPY-+HIGH PASS#                      PASS HIGH
F: LABEL F                              EXIT
```

PROGRAM 92   STATUS TEST

```
   REG9 = REG8 AND 3F                    TEST POINT (REG 9)
   IF REG8 AND 80 = 0 GOTO 3             BRANCH PRESS SWITCH
0: LABEL 0                               TEST POINT ENTRY
   DPY-JUMPER TP@9                       JUMPER TEST POINT (REG 9)
   IF REG8 AND 100 = 100 GOTO 1          BRANCH JUMPER TEST POINT HIGH
   DPY-+ LOW                             JUMPER TEST POINT LOW
   GOTO 2                                BRANCH WAIT FOR CONTINUE
1: LABEL 1                               JUMPER TEST POINT HIGH ENTRY
   DPY-+ HIGH                            JUMPER TEST POINT HIGH
2: LABEL 2                               WAIT FOR CONTINUE ENTRY
   DPY-+ THEN PRESS CONT#                PRESS CONTINUE KEY
   STOP                                  WAIT FOR CONTINUE
3: LABEL 3                               PRESS SWITCH ENTRY
   IF REG8 AND 600 = 0 GOTO 4            NO SWITCH; BRANCH READ STATUS
   REGD = REG8 SHR SHR SHR SHR
   REGD = REGD SHR SHR SHR SHR
   REGD = REGD SHR AND 3                 SWITCH NUMBER (REG D)
   DPY-HOLD SWITCH                       HOLD SWITCH DOWN
   DPY-+@D THEN PRESS CONT#              PRESS CONTINUE KEY
   STOP                                  WAIT FOR CONTINUE
4: LABEL 4                               READ STATUS ENTRY
   EXECUTE PROGRAM 91                    STATUS READER
   IF REGA AND REGC = REGA GOTO B        EXPECTED=ACTUAL; BRANCH PASS
   DPY-+ LOOP?#                          FAIL; LOOP?
   EXECUTE PROGRAM 98                    DELAY
   DPY-+#                                BEEP
   EXECUTE PROGRAM 98                    DELAY
5: LABEL 5                               ENABLE INPUT ENTRY
   DPY-+#                                BEEP
   REG1 = 40                             NO KEYS THIS VALUE
   DPY-+%1                               ENABLE INPUT
6: LABEL 6                               SELECT OPTION ENTRY
   IF REG1 = 40 GOTO 6                   LOOP UNTIL INPUT
   IF REG1 = 1C GOTO 7                   PRESSED 'YES'
   IF REG1 = 27 GOTO 7                   PRESSED 'LOOP'
   IF REG1 = 1D GOTO B                   PRESSED 'NO'
   IF REG1 = 25 GOTO B                   PRESSED 'CONTINUE'
   GOTO 5                                PRESSED INVALID KEY
7: LABEL 7                               LOOP ENTRY
   REGB = 40                             NO KEYS THIS VALUE
   DPY-+%B                               ENABLE INPUT
8: LABEL 8
   EXECUTE PROGRAM 91                    STATUS READER
   IF REGA AND REGC = REGA GOTO 9        EXPECTED=ACTUAL; BRANCH PASS
   GOTO A                                EXPECTED<>ACTUAL;BRANCH CHECK KEY
9: LABEL 9                               PASS ENTRY
   DPY-+#                                BEEP
A: LABEL A                               CHECK KEY ENTRY
   IF REGB = 40 GOTO 8                   LOOP UNTIL CONT PRESSED
   IF REGB = 25 GOTO B                   PRESSED CONT;BRANCH EXIT
   DPY-+#                                BEEP
   GOTO 7                                PRESSED INVALID KEY
B: LABEL B                               PASS ENTRY
   EXECUTE PROGRAM 98                    DELAY
C: LABEL C                               EXIT LOOP ENTRY
   IF REG8 AND 80 = 80 GOTO D            BRANCH REMOVE JUMPER
   IF REG8 AND 600 > 0 GOTO E           BRANCH RELEASE SWITCH
   GOTO F                               BRANCH EXIT
D: LABEL D                              REMOVE JUMPER ENTRY
   DPY-REMOVE JUMPER                    REMOVE JUMPER
   DPY-+ THEN PRESS CONT#               PRESS CONTINUE
   STOP                                 WAIT FOR CONTINUE
   GOTO F                               BRANCH EXIT
E: LABEL E                              RELEASE SWITCH ENTRY
   DPY-RELEASE SW@D                     RELEASE SWITCH
   DPY-+ THEN PRESS CONT#               PRESS CONTINUE
   STOP                                 WAIT FOR CONTINUE
F: LABEL F                              EXIT
```

PROGRAM 93   PROBE HISTORY READER

```
    IF REG8 AND 2000 = 2000 GOTO 1              BRANCH EVENTS
    IF REG8 AND 1000 = 1000 GOTO 5              BRANCH HISTORY
0:  LABEL 0                                     SIGNATURE ENTRY
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR AND FFFF        ACTUAL SIGNATURE (REG C)
    DPY-TP@9 SIG $A=$C                          EXPECTED SIG = ACTUAL
    GOTO F                                      BRANCH EXIT
1:  LABEL 1                                     EVENTS ENTRY
    REGC = REGC AND 7F                          ACTUAL COUNT
    REG2 = REGA AND 7F                          MAX COUNT EXPECTED
    REG1 = REGA SHR SHR SHR SHR SHR             MIN COUNT EXPECTED
    REGA = REGA SHR SHR SHR AND 7F
    IF REG1 > REG2 GOTO 2                       BRANCH COUNT WRAP
    IF REGC > REG2 GOTO 3                       BRANCH >MAX FAIL
    IF REG1 > REGC GOTO 3                       BRANCH < MIN FAIL
    GOTO 4                                      BRANCH PASS
2:  LABEL 2                                     COUNT WRAP ENTRY
    IF REG2 >= REGC GOTO 4                      BRANCH PASS
    IF REGC >= REG1 GOTO 4                      BRANCH PASS
3:  LABEL 3                                     FAIL COUNT ENTRY
    DPY-TP@9 COUNT @1-@2 =@C                    MIN-MAX=ACTUAL
    GOTO F                                      BRANCH EXIT
4:  LABEL 4                                     PASS ENTRY
    DPY-TP@9 COUNT @1-@2 =@C                    MIN-MAX=ACTUAL
    REGC = REGA                                 FORCE A PASS;COUNTS IN RANGE
    GOTO F                                      BRANCH EXIT
5:  LABEL 5                                     HISTORY ENTRY
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR
    REGC = REGC SHR SHR SHR SHR                 LOGIC LEVEL HISTORY (REG C)
    DPY-TP@9 LOGIC LVL                          TEST POINT (REG 9)
    IF REGA AND 1 = 0 GOTO 6                    BRANCH NOT HIGH
    DPY-+H                                      EXPECTED HIGH
6:  LABEL 6
    IF REGA AND 2 = 0 GOTO 7                    BRANCH NOT TRI
    DPY-+X                                      EXPECTED TRISTATE
7:  LABEL 7
    IF REGA AND 4 = 0 GOTO 8                    BRANCH NOT LOW
    DPY-+L                                      EXPECTED LOW
8:  LABEL 8
    DPY-+=                                      EQUALS
9:  LABEL 9
    IF REGC AND 1 = 0 GOTO A                    BRANCH NOT HIGH
    DPY-+H                                      READ HIGH
A:  LABEL A
    IF REGC AND 2 = 0 GOTO B                    BRANCH NOT TRISTATE
    DPY-+X                                      READ TRISTATE
B:  LABEL B
    IF REGC AND 4 = 0 GOTO C                    BRANCH NOT LOW
    DPY-+L                                      READ LOW
C:  LABEL C
    IF REGC > 0 GOTO F                          BRANCH NOT TRISTATE
    DPY-+X                                      READ TRISTATE
F:  LABEL F                                     EXIT
```

PROGRAM 94   PROBE HISTORY TEST

```
   REG9 = REG8 AND 3F                                    TEST POINT (REG 9)
   REGA = REG8 SHR SHR SHR SHR SHR SHR SHR SHR
   REGA = REGA SHR SHR SHR SHR SHR SHR SHR SHR           EXPECTED PROBE READING
   DPY-PROBE TP@9                                        TEST POINT (REG 9)
   EXECUTE PROGRAM 96                                    PLACE PROBE
   SYNC FREE-RUN
   IF REG8 AND C000 = 0 GOTO 0                           SYNC FREE RUN
   SYNC ADDRESS
   IF REG8 AND 4000 > 0 GOTO 0                           SYNC ADDRESS
   SYNC DATA                                             SYNC DATA
0: LABEL 0
   REG2 = REG8 SHR SHR SHR SHR SHR SHR AND 3F
   EXECUTE PROGRAM REG2                                  TEST PROGRAM (REG 2)
   EXECUTE PROGRAM 93                                    PROBE HISTORY READER
   IF REGA = REGC GOTO 7                                 EXPECTED=PROBE READING
   DPY-+ FAIL LOOP?#                                     FAIL; LOOP ?
   EXECUTE PROGRAM 98                                    DELAY
   DPY-+#                                                BEEP
   EXECUTE PROGRAM 98                                    DELAY
1: LABEL 1                                               ENABLE INPUT ENTRY
   DPY-+#                                                BEEP
   REG1 = 40                                             NO KEYS THIS VALUE
   DPY-+%1                                               ENABLE INPUT
2: LABEL 2                                               SELECT OPTION ENTRY
   IF REG1 = 40 GOTO 2                                   LOOP UNTIL INPUT
   IF REG1 = 1C GOTO 3                                   PRESSED 'YES'
   IF REG1 = 27 GOTO 3                                   PRESSED 'LOOP'
   IF REG1 = 1D GOTO 8                                   PRESSED 'NO'
   IF REG1 = 25 GOTO 8                                   PRESSED 'CONTINUE'
   GOTO 1                                                PRESSED INVALID KEY
3: LABEL 3                                               LOOP ENTRY
   REGB = 40                                             NO KEYS THIS VALUE
   DPY-+%B                                               ENABLE INPUT
4: LABEL 4
   REG2 = REG8 SHR SHR SHR SHR SHR SHR AND 3F
   EXECUTE PROGRAM REG2                                  TEST PROGRAM (REG 2)
   EXECUTE PROGRAM 93                                    PROBE HISTORY READER
   IF REGA = REGC GOTO 5                                 EXPECTED=ACTUAL;PASS
   DPY-+ FAIL                                            FAIL
   GOTO 6                                                BRANCH CHECK KEY
5: LABEL 5                                               PASS ENTRY
   DPY-+ PASS#                                           PASS
6: LABEL 6                                               CHECK KEY ENTRY
   IF REGB = 40 GOTO 4                                   LOOP UNTILL CONT PRESS
   IF REGB = 25 GOTO 8                                   PRESSED CONT;EXIT
   DPY-+#                                                BEEP
   GOTO 3                                                PRESSED INVALID KEY
7: LABEL 7                                               PASS ENTRY
   DPY-+ PASS#                                           PASS
   EXECUTE PROGRAM 98                                    DELAY
8: LABEL 8                                               EXIT LOOP ENTRY
   EXECUTE PROGRAM 95                                    REMOVE PROBE
```

PROGRAM 95   REMOVE PROBE

```
   SYNC FREE-RUN                         FREE RUN PROBE
0: LABEL 0                               BEGIN PASS COUNT ENTRY
   REG1 = 4                              INITIALIZE PASS COUNTER
1: LABEL 1                               BEGIN HISTORY LOOP
   READ PROBE                            READ PROBE HISTORY
   IF REG0 AND 5000000 = 0 GOTO 2        BRANCH; NOT HIGH OR LOW
   DPY-REMOVE PROBE                      HIGH OR LOW DETECTED
   GOTO 0                                START OVER
2: LABEL 2                               TRI-STATE ENTRY
   DEC REG1                              DECREMENT PASS COUNTER
   IF REG1 > 0 GOTO 1                    LOOP 4 TIMES
F: LABEL F                               EXIT WHEN 4 CONSECITIVE
                                         READS ARE TRISTATE.
```

PROGRAM 96   PLACE PROBE

```
   SYNC FREE-RUN                         FREE RUN PROBE
   REG1 = 6F                             INITIALIZE TIME OUT COUNTER
0: LABEL 0                               BEGIN PASS COUNT ENTRY
   DEC REG1                              DECREMENT TIME OUT COUNTER
   IF REG1 = 0 GOTO F                    BRANCH TIME OUT
   REG2 = 4                              INITIALIZE PASS COUNTER
1: LABEL 1                               BEGIN HISTORY LOOP
   READ PROBE                            READ PROBE HISTORY
   IF REG0 AND 5000000 = 0 GOTO 0        BRANCH NOT HIGH OR LOW
   DEC REG2                              DECREMENT PASS COUNTER
   IF REG2 > 0 GOTO 1                    BRANCH READ AGAIN
F: LABEL F                               EXIT WHEN 4 CONSECITIVE READS
                                         ARE NON-TRISTATE, OR AFTER A
                                         4 SECOND TIMEOUT.
```

PROGRAM 97   1 SECOND DELAY

```
0: LABEL 0
   INC REG 1
   IF 4F > REG1 GOTO 0
```

PROGRAM 98   1/4 SECOND DELAY

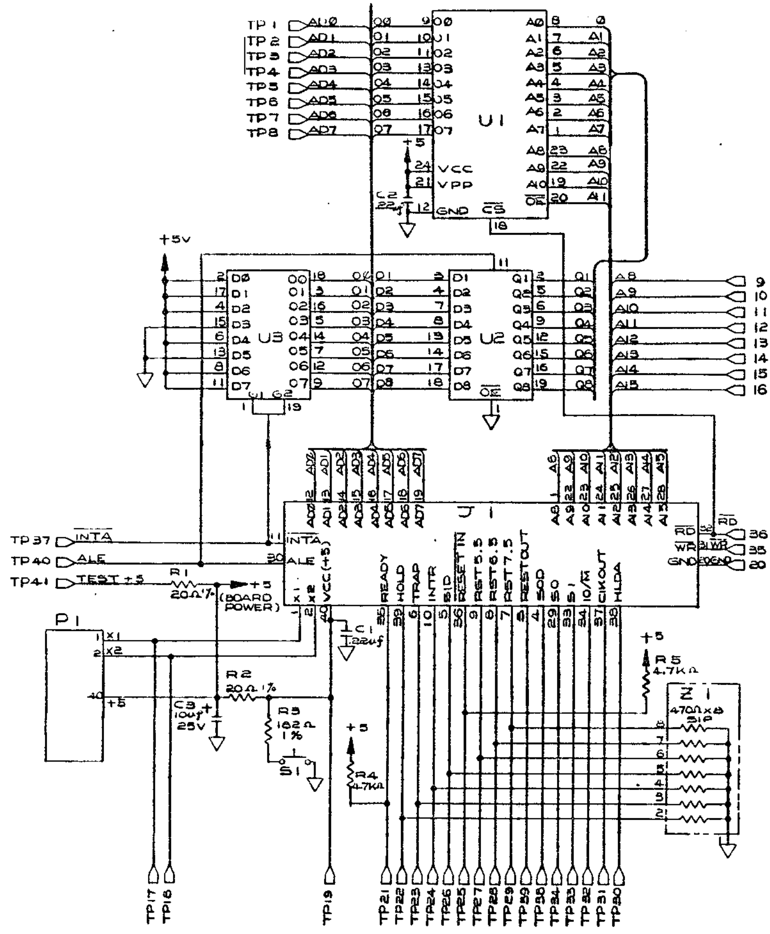
```
0: LABEL 0
   INC REG1
   IF F > REG1 GOTO 0
```

```
PROGRAM 64  8085 POD TESTS

0: LABEL 0
   DPY-ENTER STARTING TEST 1-13 ?
   DPY-+\1
   IF REG1 = 1 GOTO 1                   POWER SUPPLY CHECK
   IF REG1 = 2 GOTO 2                   CLOCK CHECK
   IF REG1 = 3 GOTO 3                   STATUS CHECK
   IF REG1 = 4 GOTO 4                   READ STATUS TEST
   IF REG1 = 5 GOTO 5                   POWER SUPPLY STATUS TEST
   IF REG1 = 6 GOTO 6                   CONTROL CHECK
   IF REG1 = 7 GOTO 7                   WRITE CONTROL TEST
   IF REG1 = 8 GOTO 8                   ADDRESS TOGGLE TEST
   IF REG1 = 9 GOTO A                   DATA TOGGLE TEST
   IF REG1 = A GOTO C                   BUS TEST
   IF REG1 = B GOTO D                   READ DATA TEST
   IF REG1 = C GOTO E                   FIXTURE ROM TEST
   IF REG1 = D GOTO F                   SIGNATURE STABILITY TEST
   GOTO 0
1: LABEL 1                              *** POWER SUPPLY CHECK ***
   DPY-POWER SUPPLY CHECK#
   EXECUTE PROGRAM 97
   REG8 = 00041054                      GROUND
   EXECUTE PROGRAM 94
   REG8 = 00011053                      +5 VOLT
   EXECUTE PROGRAM 94
2: LABEL 2                              *** CLOCK CHECK ***
   DPY-CLOCK CHECK#
   EXECUTE PROGRAM 97
   REG8 = 00051051                      X1
   EXECUTE PROGRAM 94
   REG8 = 00051052                      X2
   EXECUTE PROGRAM 94
3: LABEL 3                              *** STATUS CHECK ***
   DPY-STATUS CHECK#
   EXECUTE PROGRAM 97
   REG8 = 00011055                      READY
   EXECUTE PROGRAM 94
   REG8 = 00041056                      HOLD
   EXECUTE PROGRAM 94
   REG8 = 00041057                      TRAP
   EXECUTE PROGRAM 94
   REG8 = 00041058                      INTR
   EXECUTE PROGRAM 94
   REG8 = 00011059                      RESET - IN
   EXECUTE PROGRAM 94
   REG8 = 0004105A                      SID
   EXECUTE PROGRAM 94
   REG8 = 0004105B                      RST 5.5
   EXECUTE PROGRAM 94
   REG8 = 0004105C                      RST 6.5
   EXECUTE PROGRAM 94
   REG8 = 0004105D                      RST 7.5
   EXECUTE PROGRAM 94
4: LABEL 4                              *** READ STATUS TEST ***
   DPY-READ STATUS TEST-WAIT#
   EXECUTE PROGRAM 97
   REG8 = 00001815                      READY
   EXECUTE PROGRAM 92
   REG8 = 00002016                      HOLD
   EXECUTE PROGRAM 92
   REG8 = 00004017                      TRAP
   EXECUTE PROGRAM 92
   REG8 = 00008018                      INTR
   EXECUTE PROGRAM 92
   REG8 = 00010819                      RESET - IN
   EXECUTE PROGRAM 92
   REG8 = 0002001A                      SID
   EXECUTE PROGRAM 92
   REG8 = 0010001B                      RST 5.5
   EXECUTE PROGRAM 92
   REG8 = 0020001C                      RST 6.5
   EXECUTE PROGRAM 92
   REG8 = 0040001D                      RST 7.5
   EXECUTE PROGRAM 92
   REG8 = 00001095                      JUMPER READY LOW
   EXECUTE PROGRAM 92
   REG8 = 00002996                      JUMPER HOLD HIGH
   EXECUTE PROGRAM 92
   REG8 = 00004997                      JUMPER TRAP HIGH
   EXECUTE PROGRAM 92
   REG8 = 00008998                      JUMPER INTR HIGH
   EXECUTE PROGRAM 92
   REG8 = 00010099                      JUMPER RESET - IN  LOW
   EXECUTE PROGRAM 92
   REG8 = 0002099A                      JUMPER SID HIGH
   EXECUTE PROGRAM 92
   REG8 = 0010099B                      JUMPER RST 5.5 HIGH
   EXECUTE PROGRAM 92
   REG8 = 0020099C                      JUMPER RST 6.5 HIGH
   EXECUTE PROGRAM 92
   REG8 = 0040099D                      JUMPER RST 7.5 HIGH
   EXECUTE PROGRAM 92
5: LABEL 5                              *** POWER SUPPLY STATUS TEST ***
   DPY-POWER SUPPLY STATUS TEST#
   EXECUTE PROGRAM 97
   REG8 = 00080000                      NO FAULT
   EXECUTE PROGRAM 92
   REG8 = 00080A00                      + 5  VOLT FAULT
   EXECUTE PROGRAM 92
6: LABEL 6                              *** CONTROL CHECK ***
   DPY-CONTROL CHECK#
   EXECUTE PROGRAM 97
   REG8 = 0004105E                      HLDA
   EXECUTE PROGRAM 94
   REG8 = 0005105F                      CLOCK OUT
   EXECUTE PROGRAM 94
   REG8 = 00041060                      IO/M
   EXECUTE PROGRAM 94
   REG8 = 00011061                      S1
   EXECUTE PROGRAM 94
   REG8 = 00051062                      SO
   EXECUTE PROGRAM 94
   REG8 = 00011063                      WR
   EXECUTE PROGRAM 94
   REG8 = 00051064                      RD
   EXECUTE PROGRAM 94
   REG8 = 00011065                      INTA
   EXECUTE PROGRAM 94
   REG8 = 00041066                      SOD
   EXECUTE PROGRAM 94
   REG8 = 00041067                      RESET OUT
   EXECUTE PROGRAM 94
   REG8 = 00051068                      ALE
   EXECUTE PROGRAM 94
7: LABEL 7                              *** WRITE CONTROL TEST ***
   DPY-WRITE CONTROL TEST#
   EXECUTE PROGRAM 97
   REGD = 0
   REG8 = 0005111E                      TOGGLE HLDA
   EXECUTE PROGRAM 94
   REGD = 2
   REG8 = 00051125                      TOGGLE INTA
   EXECUTE PROGRAM 94
   REGD = 3
   REG8 = 00051126                      TOGGLE SOD
   EXECUTE PROGRAM 94
   REGD = 1
   REG8 = 00071127                      TOGGLE RESET OUT
   EXECUTE PROGRAM 94
8: LABEL 8                              *** ADDRESS TOGGLE TEST ***
   DPY-ADDRESS TOGGLE TEST#             TOGGLE AD0-AD15
   EXECUTE PROGRAM 97
   REGD = 0
   REG8 = 00055081
9: LABEL 9
   EXECUTE PROGRAM 94
   INC REGD
   INC REG8
   IF 10 > REGD GOTO 9
A: LABEL A                              *** DATA TOGGLE TEST ***
   DPY-DATA TOGGLE TEST#                TOGGLE D0-D7
   EXECUTE PROGRAM 97
   REGD = 0
   REG8 = 00590C1
B: LABEL B
   EXECUTE PROGRAM 94
   INC REGD
   INC REG8
   IF 8 > REGD GOTO B
C: LABEL C                              *** BUS TEST ***
   DPY-BUS TEST#
   EXECUTE PROGRAM 97
   DPY-+-WAIT
   BUS TEST
D: LABEL D                              *** READ DATA TEST ***
   DPY-READ DATA TEST-WAIT#
   EXECUTE PROGRAM 97
   REG8 = FFFFFF                        READ @ FFFF = FF
   EXECUTE PROGRAM 90
   REG8 = 000200                        READ @ 0002 = 00
   EXECUTE PROGRAM 90
E: LABEL E                              *** SIGNATURE STABILITY TEST ***
   DPY-SIGNATURE STABILITY TEST#
   EXECUTE PROGRAM 97
   REG8 = 96EC8181                        RAMP @ FFFF; SIG @ A0 = 96EC
   EXECUTE PROGRAM 94
F: LABEL F                                *** FIXTURE ROM SIGNATURE TEST ***
   DPY-FIXTURE ROM TEST#
   EXECUTE PROGRAM 97
   DPY-+-WAIT
   ROM TEST @ 0 - 47F = SIG 3D1A
   DPY-*** NORMAL TEST
   DPY-+ COMPLETE ***#
   EXECUTE PROGRAM 97

PROGRAM 65  8085 POD "RUN UUT" TEST

   DPY-*** 8085 POD 'RUN UUT'
   DPY-+ TESTS ***#
   EXECUTE PROGRAM 97
0: LABEL 0                                *** 'RUN UUT' CONTROL TESTS ***
   DPY-'RUN UUT' CONTROL TESTS#
   EXECUTE PROGRAM 97
   RUN UUT @ 0
   DPY-TOUCH TP25 LOW                     RESET
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0004105E                        HLDA
   EXECUTE PROGRAM 94
   REG8 = 00041060                        IO/M
   EXECUTE PROGRAM 94
   REG8 = 00051061                        S1
   EXECUTE PROGRAM 94
   REG8 = 00051062                        S0
   EXECUTE PROGRAM 94
   REG8 = 00051063                        WR
   EXECUTE PROGRAM 94
   REG8 = 00051064                        RD
   EXECUTE PROGRAM 94
   REG8 = 00011065                        INTA
   EXECUTE PROGRAM 94
   REG8 = 00041066                        SOD
   EXECUTE PROGRAM 94
   REG8 = 00041067                        RESET OUT
   EXECUTE PROGRAM 94
   REG8 = 00051068                        ALE
   EXECUTE PROGRAM 94
1: LABEL 1                                *** 'RUN UUT' ADDRESS TESTS ***
   DPY-'RUN UUT' ADDRESS TESTS#                ---------------
   EXECUTE PROGRAM 97                          | AD0  TOGGLE |
   REG8 = 00051041                             | AD1  TOGGLE |
2: LABEL 2                                     | AD2  TOGGLE |
   EXECUTE PROGRAM 94                          | AD3  TOGGLE |
   INC REG8                                    | AD4  TOGGLE |
   IF 00051049 > REG8 GOTO 2                   | AD5  TOGGLE |
   REG8 = 00041049                             | AD6  TOGGLE |
   EXECUTE PROGRAM 94                          | AD7  TOGGLE |
   REG8 = 0004104A                             | A8   LOW    |
   EXECUTE PROGRAM 94                          | A9   LOW    |
   REG8 = 0005104B                             | A10  TOGGLE |
   EXECUTE PROGRAM 94                          | A11  LOW    |
   REG8 = 0004104C                           | A12  LOW     |
3: LABEL 3                                   | A13  LOW     |
   EXECUTE PROGRAM 94                        | A14  LOW     |
   INC REG8                                  | A15  LOW     |
   IF 00041051 > REG8 GOTO 3                 ---------------
4: LABEL 4                              *** 'RUN UUT' INTERRUPT TESTS ***
   DPY-'RUN UUT' INTERRUPT TESTS#
   EXECUTE PROGRAM 97
   DPY-JUMPER TP24 HIGH                 INT
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 00051065                      INTA
   EXECUTE PROGRAM 94
   DPY-REMOVE JUMPER
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0005104D                      A12
   EXECUTE PROGRAM 94
   DPY-TOUCH TP23 HIGH                  TRAP
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0005104C                      A11
   EXECUTE PROGRAM 94
   DPY-TOUCH TP27 HIGH                  RST 5.5
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0005104E                      A13
   EXECUTE PROGRAM 94
   DPY-TOUCH TP28 HIGH                  RST 6.5
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0005104F                      A14
   EXECUTE PROGRAM 94
   DPY-JUMP TP26 TO 38                  SID TO SOD
   DPY-+ THEN PRESS CONT#
   STOP
   DPY-TOUCH TP 29 HIGH                 RST 7.5
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 00051050                      A15
   EXECUTE PROGRAM 94
   REG8 = 00071066                      SOD
   EXECUTE PROGRAM 94
   DPY-REMOVE JUMPER
   DPY-+THEN PRESS CONT#
   STOP
5: LABEL 5                              *** HOLD TEST ***
   DPY-JUMPER TP22 HIGH                 HOLD
   DPY-+ THEN PRESS CONT#
   STOP
   REG8 = 0001105E                      HLDA
   EXECUTE PROGRAM 94
   DPY-REMOVE JUMPER
   DPY-+ THEN PRESS CONT#
   STOP
   DPY-*** RUN UUT TEST
   DPY-+ COMPLETE ***#
   EXECUTE PROGRAM 97
```



REF DES	DEVICE	+5	GND	
U1	2716-1	21.24	12	
U2	74LS573	20	10	
U3	74LS244	20	10	
Z1	470Ω SIP	—	1	

SIGNEL	VERIFIED NORMAL MODE	VERIFIED WITH RUN UUT
D0 - D7	PROBED (SYNC DATA)	BY ROM
A0 - A15	PROBED (SYNC ADDRESS)	A0-A10 BY ROM A11-A15 PROBED
RESET IN	PULSED LOW STATUS READ	PULSED LOW A10 TOGGLES
TRAP	PULSED HIGH STATUS READ	PULSED HIGH A11 TOGGLES
INTR	PULSED HIGH STATUS READ	PULSED HIGH A12 TOGGLES
RST 5.5	PULSED HIGH STATUS READ	PULSED HIGH A13 TOGGLES
RST 6.5	PULSED HIGH STATUS READ	PULSED HIGH A14 TOGGLES
RST 7.5	PULSED HIGH STATUS READ	PULSED HIGH A15 TOGGLES
READY	PULSED HIGH STATUS READ	PROBED
HOLD	PULSED HIGH STATUS READ	PULLED HIGH-VERIFY A11-A15 NOT ACTIVE
HLDA	WRITTEN FROM WRITE CTRL	PROBED DURING HOLD
SID	PULSED HIGH STATUS READ	PROBED
SOD	WRITTEN FROM WRITE CTRL	PROBED
S0	PROBED	PROBED
S1	PROBED	PROBED
IO/M	PROBED	PROBED
INTA	WRITTEN FROM WRITE CTRL	PROBED DURING INTR
ALE	PROBED	PROBED
X1	PROBED	——
X2	PROBED	——
VCC	PUSH BUTTON TO FAIL	——
RD	PROBED	PROBED
WR	PROBED	PROBED
RESET OUT	WRITTEN FROM WRITE CTRL	PROBED DURING RESET IN

REVISIONS

LTR	DESCRIPTION	EFF	APPROVAL / DATE

QTY	ITEM	PART NO.	DRAWING NO	DESCRIPTION	REF DES

PARTS LIST

PROPRIETARY NOTICE

[illegible]

FLUKE

89536

JOHN FLUKE MFG. CO., INC.
P.O. BOX 43210
MOUNTLAKE TERRACE, WA 98043

DO NOT SCALE DRAWING

UNLESS OTHERWISE SPECIFIED

TOLERANCE / DIMENSIONS IN INCHES

DWN BIBI. CAMPBELL 12-8-81
CHK 12-30-81
PROJ ENGR
MFG ENGR
FINISH

SCHEMATIC CUSTOMER SERVICE TEST FIXTURE

PART NO. | SIZE D | DRAWING NO 90004-8085-1073

SCALE NONE | SH 1 OF 1 | REV —

BASIC | DWG NO SUFFIX | DRAWING NO | PART NO. | FIRST USED ON | APPLICATION

90004-8085-1073 SH 1 OF 1 REV