# MODULE REPAIR GUIDE NO. 3

EDITED BY CUSTOMER ENGINEERING DIVISION

January 13, 1978

MODEL 2236 SYSTEM REPAIR

## 1. INTRODUCTION

This guide covers the repair procedure and troubleshooting aids for the 2236 Interactive Terminal, and the 2236 MXC controller board. Repair of the 210-7292 and the controller (210-7290 and 210-7291) is facilitated by using the 928 Debug Tester as an aid. Sections 2 and 3 deal with the installation and operational procedures of the tester, and Section 4 contains the Debug PROM(s) microprogram and explanation. Schematics for the tester, terminal and controller are included at the end of this repair guide.

## 2. INSTALLATION

### 2.1 EQUIPMENT REQUIRED

| QTY. | PART # | DESCRIPTION |
|---|---|---|
| 1 | 270-0350 | 928 Debug Chassis |
| 1 | 220-0153 | 928 Debug Cable |
| 1 | 210-7257 | Interface and Timing PCB |
| 1 | 378-2135 | 2236 Terminal Debug PROM |
| 1 | 378-2136 | 2236 MXC Debug PROM #1 |
| 1 | 378-2155 | 2236 MXC Debug PROM #2 |
| 1 | 701-2291 | Microcode Interface Program Diskette |

### 2.2 PROCEDURE

Install the 210-7257 PCB into the HALT/STEP LOGIC position in the debug chassis (the position closest to the front of the chassis). Reference Figure 1. Connect the debug cable between the 7257 and connector 4 of the 7292 PCB or J1 of the 7291 PCB. Pin 1 of both ends of the cable must plug into or onto pin 1 of the appropriate connector.

Printed in U.S.A.

WANG LABORATORIES, INC.

ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

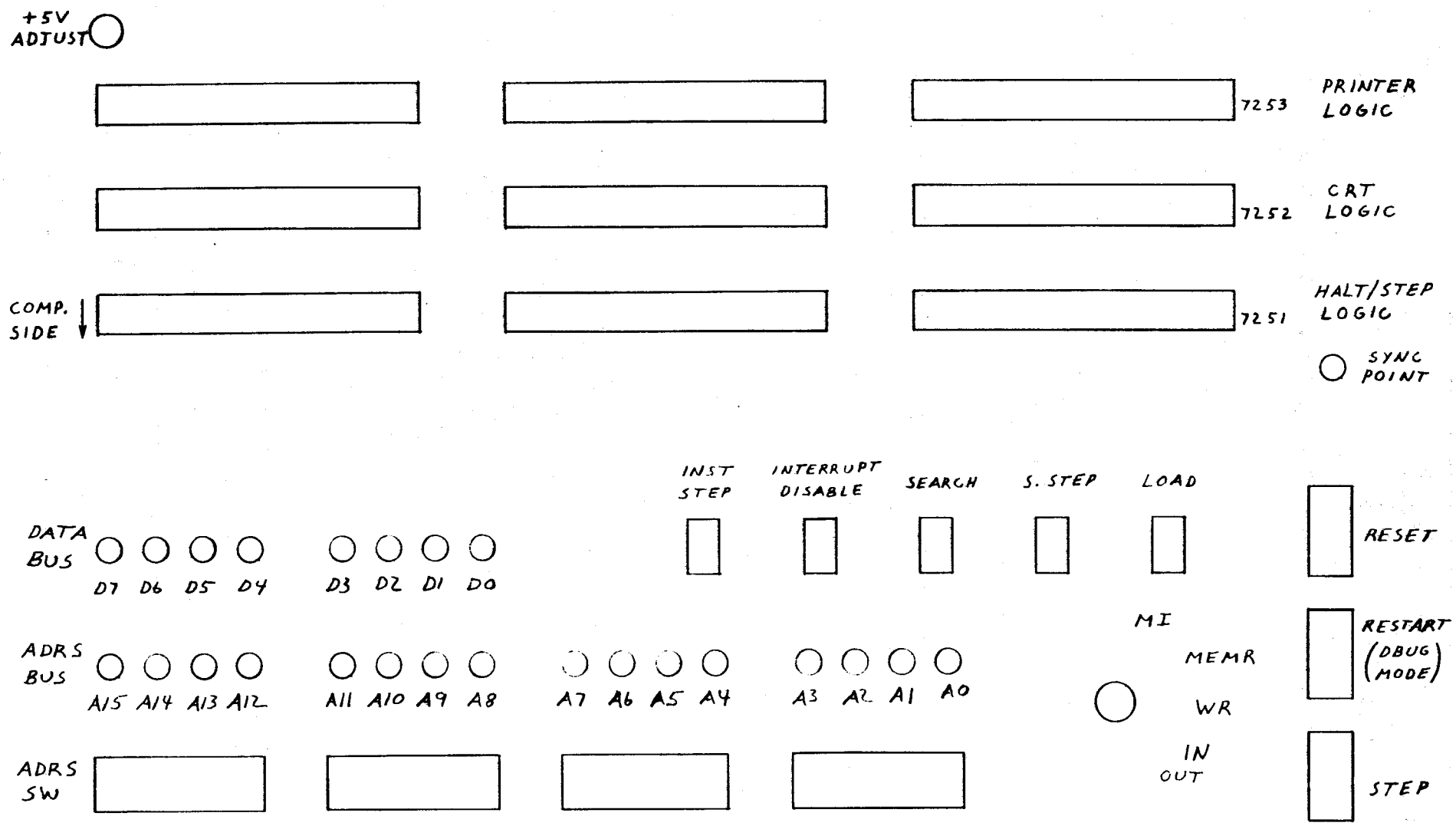

+5V ADJUST
7253
PRINTER LOGIC
7252
CRT LOGIC
COMP. SIDE
7251
HALT/STEP LOGIC
SYNC POINT
INST STEP
INTERRUPT DISABLE
SEARCH
S. STEP
LOAD
RESET
DATA BUS
D7 D6 D5 D4 D3 D2 D1 D0
MI
MEMR
WR
IN
OUT
RESTART (DBUG MODE)
ADRS BUS
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
ADRS SW
STEP

FIGURE 1

For normal operation of the 2236 Terminal or MXC, all the toggle switches on the debug chassis should be down (off). Switch A and B on the 7257 should be up and switch C must be down (reference 7257 schematic). These switches will be explained in the next section.

## 3. OPERATION

### 3.1 DATA/ADDRESS SELECT

The DATA/ADDRESS SELECT switches are the three locking pushbutton switches located at the top of the 210-7257 PCB. These switches are used to select the correct polarities of the data and address busses that are appropriate for the specific system that the debug tester is being used with. Switch C (on the right) selects bar address and true data which is needed for correct operation with a 2236 Terminal or MXC.

### 3.2 LIGHT INDICATORS

There are two sets of light indicators on the chassis. They represent the outputs of the following:

1) 8080 Data Bus ($D_7$-$D_0$).
2) 8080 Address Bus ($A_{15}$-$A_0$).

### 3.3 INSTRUCTION STEP

The instruction step switch (INST. STEP) is used in conjunction with the STEP pushbutton. When it is on (up), the clock is disabled and one complete instruction cycle will be executed only when the STEP pushbutton is depressed. The instruction may be anywhere from 1 to 3 bytes long.

The 5-position rotary switch (instruction select switch) must be at M1 for the instruction step switch to work properly. Reference section 3.8 for explanation of the instruction select switch.

NOTE:

If it is desired to return to the automatic cycle after having disabled the clock, the STEP pushbutton must be depressed one more time to initiate the free-running clock cycle again.

## 3.4 SINGLE STEP

When the SINGLE STEP switch is on (up), the tester will allow the 8080 to execute only one byte of an instruction, whether the instruction be 1, 2 or 3 bytes long, every time the STEP pushbutton is depressed.

When in the single step mode and an instruction involving memory or the stack is performed (CALL, PUSH, MOV A, M, etc.), the memory or stack location that is being accessed will be displayed on the address bus indicators. The data at that location or data being transferred to that location will be displayed on the data bus after the last byte of the instruction has been executed. If an IN or OUT is performed, the appropriate IN or OUT address (reference section 3.9, examples 4 and 5) will be displayed on the address bus and the data coming in or going out will be on the data bus indicators after the last byte of the instruction has been executed.

NOTE:

This switch also disables the clock, therefore, the step pushbutton must be depressed once after the switch is turned off (down) to start the clock again.

## 3.5 INTERRUPT DISABLE

The INTERRUPT DISABLE switch is not applicable to the 2236 Terminal and should be left off (down). The disable switch can however be used to disable any 8080 interrupt when the tester is being used on the 2236 MXC.

## 3.6 LOAD

The LOAD switch is not applicable to the 2236 Terminal or MXC and should be left off (down).

## 3.7 RESTART (DEBUG MODE)

The RESTART pushbutton switch is not applicable to the 2236 Terminal or MXC.

## 3.8 INSTRUCTION SELECT

One 5-position rotary switch is available to select the desired instruction type to be stopped on when in the SEARCH mode. The options are:

1) M1 - Instruction Fetch.
2) MEMR - Memory Read.
3) WR - Memory Write.
4) IN - 8080 Input.
5) OUT - 8080 Output.

## 3.9 SEARCH MODE

The SEARCH switch is used in conjunction with the address switches (ADRS. SW.), the rotary instruction select switch and the STEP pushbutton. Once the instruction type has been selected (rotary switch) and the appropriate address switches set, the SEARCH switch may be turned on (up) allowing the tester to stop the microprogram when the designated instruction is reached. The instruction byte that the microprogram is stopped at will not be executed until the step pushbutton is depressed. The search feature allows for internal observation at a certain point in the program without having to step through the preceding instructions of the program.

EXAMPLE 1: Instruction select set at M1 (instruction fetch), address switches set for 00A7.

When SEARCH is enabled, the tester will stop the microprogram at address (program step) 00A7 if and only if 00A7 is an instruction fetch (the first byte of every instruction). The program may be continued by depressing the step pushbutton and it will only stop if instruction fetch at 00A7 is accessed again (e.g., if the microprogram loops back).

EXAMPLE 2: Instruction select set at MEMR (memory read), address switches set for 013B.

NOTE:

Memory read allows the program to be stopped at any point since every byte of every instruction is read from PROM.

Enabling the search mode will stop the program at memory read 013B. Depressing the step pushbutton allows the program to continue and it will only stop if memory read at 013B is accessed again.

Memory read may also be used to monitor any location in RAM in order to observe the data that is in the location.

EXAMPLE 3: Instruction select set at WR (memory write), address switches set for 203E.

When the search mode is enabled, the tester will monitor for a write into memory (RAM) at location 203E. When data is written into that location, the program will be stopped with the memory location displayed on the address indicators and the data that was written into that location on the data bus indicators. Depressing the step pushbutton will continue the program and it will only stop if RAM location 203E is written into again.

EXAMPLE 4: Instruction select set at IN (8080 input).

The address switches should be set according to what type of IN it is desired to stop on. The settings (in HEX) are as follows:

2236 TERMINAL:

IN X'00' (UART data) - 0000
IN X'01' (Keyboard data) - 0101
IN X'02' (Status latch data) - 0202
IN X'03' (Four level switch status) - 0303

2236 MXC:

IN X'00' (Modem ready status) - 0000
IN X'01' (2200 interface status) - 0101
IN X'02' (2200 data) - 0202
IN X'03' (Address bits) - 0303
IN X'04' (8251 receiver ready status) - 0404
IN X'06' (USART data) - 0606
IN X'0E' (USART status) - 0E0E

When the search switch is turned on, the program will stop at the first available IN of the desired type. The data coming in will be displayed on the data bus indicators.

If the step pushbutton is depressed, the program will continue until it finds the next appropriate IN.

EXAMPLE 5: Instruction select set at OUT (8080 output).

The address switches should be set according to what type of OUT it is desired to stop on. The settings (in HEX) are as follows:

2236 TERMINAL:

OUT X'00' (UART data) - 0000
OUT X'01' (Rate latch data) - 0101
OUT X'02' (Command latch data) - 0202
OUT X'03' (Output latch data) - 0303

2236 MXC:

OUT X'00' (Clear reset status bit) - 0000
OUT X'01' (Data to 2200) - 0101
OUT X'11' (Data to 2200 with ENDI) - 1111
OUT X'02' (Reset to 2200) - 0202
OUT X'03' (HALT/STEP to 2200) - 0303
OUT X'05' (Select USART chip) - 0505
OUT X'06' (Data to USART) - 0606
OUT X'07' (Ready/Busy status) - 0707
OUT X'0E' (Mode/control data to USART) - 0E0E

If search is initiated, the program will stop at the first available OUT of the desired type. The data being sent out will be displayed on the data bus indicators. If the step pushbutton is depressed, the program will continue until it finds the next appropriate OUT.

### 3.10 SYNC POINT

Above the reset pushbutton is a solder point into which a heavy gauge wire may be placed. This point is labeled SYNC POINT and is used in conjunction with the address toggle switches and the rotary instruction select switch. The sync point is normally high and will go low only when the program reaches the step (address) that is set on the address switches as long as the addressed byte is of the type set on the rotary instruction select switch. The sync point can be used to trigger an oscilloscope on the selected step in the program or it can be observed to see if that address is ever reached.

## 4. DEBUG MICROPROGRAMS

### 4.1 GENERAL

Due to the length, complexity and interaction of the two micro-programs that normally control the 2236 Terminal and MXC, two programs have been written which will allow for isolated testing and trouble-

shooting of each. The program for the terminal is contained in a single PROM (378-2135), while the program for the MXC is contained in two PROMs (378-2136, 378-2155) and operates in conjunction with a 2200 BASIC program (701-2291).

NOTE:

The 2236 MXC microprogram and the 2200 BASIC program will operate correctly in a 2200T or a 2200VP.

Insert the PROM(s) as follows:

| PCB | LOCATION | PROM # |
|---|---|---|
| 7291 | L32 | 378-2136 (#1) |
| 7291 | L17 | 378-2155 (#2) |
| 7292 | L43 | 378-2135 |

The PROMs located at L15 and L16 of the 7291 PCB need not be removed.

An RS-232-C loop back connector must be made and installed in place of the normal cable. To make to loop back connector, wire a 25 pin RS-232-C male connector (350-1030) as follows:

| JUMPER PIN | TO PIN |
|---|---|
| 2 | 3 |
| 4 | 5 |
| 6 | 20 |

## 4.2 2236 TERMINAL DEBUG PROGRAM (Refer to Appendix A)

There are 6 basic tests in the 2236 Terminal program. The first is a general CPU check which initializes the terminal, prints "READY:" on the CRT and waits for the selection of a desired test. This test is initiated at power up.

The 5 remaining tests are selected via special function keys '01 to '04. To jump from one test to another, RESET must be keyed before selecting another special function key. All tests are continuous and stop on error (except keyboard test).

NOTE:

All of the tests rely on a portion of RAM (stack) to be good and available. If there is any doubt as to the condition of the RAM ICs, change them.

SF'00 - KEYBOARD TEST

The keyboard test allows for selection of only two characters. The first is a shifted "period" (keyboard code HEX(55)), the other is "<" (keyboard code HEX(AA)). The selected character is printed on the CRT for verification.

SF'01 - UART TEST

This test will send data to the UART, read it back and check for data integrity. The data sent starts with HEX (00) and is incremented by HEX (01) up to HEX (FF). The baud rate may be set at any of the possible selections (refer to Service Bulletin #80).

SF'02 - RAM TEST

Because the other tests rely on the STACK portion of RAM to be good and available, this test is mainly for detecting bad locations in RAM. Data is sent to RAM, read from it and checked for integrity. The data is incremented from HEX (00) to HEX (FF) as in the UART test.

SF'03 - CRT TEST

The CRT test is basically a transfer and buffer check for the 210-7158 PCB. The screen is flooded with the character whose HEX code is (10). The code is then incremented and the next character is

displayed. Hex codes (10) to (FF) are displayed. All of the control codes are not checked. HALT/STEP may be keyed to stop the test for observation of a possibly bad character. To continue the test, RESET must be keyed and the test restarted.

SF'04 - PRINTER TEST

This test will insure correct operation of the printer controller included on the 210-7158 PCB. HEX codes (20) to (7E) are sent to the 2221W or 2231W for printout.

4.3 2236 MXC DEBUG PROGRAM (Refer to Appendix B)

NOTE:

The controller address select switch (SW 1) must be set such that the board is enabled by an address of HEX (81) to (87). Switch #2 on the 5 bank switch must be on and all others off. Reference schematic of 210-7290.

The 2236 MXC program consists of a power up routine and 8 tests. The power up routine will not be completely executed until after the 2200 BASIC program is run. The routine waits for a CBS from the 2200, checks to make sure the controller is enabled and then waits for a test to be selected via a special function key. The tests should be selected in the order of the special function keys assigned to them (SF'00 first and SF'05 last).

If an error is detected in one of the tests or if it is desired to jump to another test, RESET should be keyed and the 2200 program rerun prior to selecting another special function key. A description of the 8 tests follows.

SF'00 - START TEST

The first test is a combination of a communication check between the 2200 and the MXC, and a STACK test. If this test will not run without error, the remaining tests will not function properly.

The communication check is divided into three parts. The first tests CR/B, IBS, OBS, CBS and general operation of the enable circuitry. The second part checks the CRT size selection circuitry that is connected to IB5. The last part insures the ENDI bit is operating.

The stack test is to insure that the last 20 locations of the 4K RAM (L36-L43) are good and can be used for storage. The remaining tests depend on the STACK and they will not run correctly if it is not working, therefore the START test should be run at least once prior to running the rest of the tests.

The next five tests may be run in two different modes; 1) stop on error, 2) loop on error. The first special function key that is listed is for the stop on error mode, the second is for the loop on error mode. When in the loop on error mode, the test will start from the beginning after the error is detected.

SF'01, SF'17 - INPUT BUS (IB) TEST

The IB tests sends data to the 2200 through the IB latches L28, L29, L40 and L41. The 2200 then verifies the data for accuracy. The data sent by the 8080 starts at HEX (00) and is incremented to HEX (FF).

SF'02, SF'18 - OUTPUT BUS (OB) TEST

In this test, the 2200 sends data to the 8080 through latches L38 and L50. The data is immediately echoed back to the 2200 where it is checked for errors. The data that the 8080 inputs starts at HEX (00) and increments to HEX (FF).

SF'03, SF'19 - ADDRESS BUS (AB) TEST

Hex addresses (81) to (87) are sent from the 2200 to enable the controller. The address is strobed through the input latches (L35, L36, L37 and L48) and checked by the 8080 for validity. If incorrect, an error message is sent to the 2200.

SF'04, SF'20 - RAM TEST

The RAM test is capable of testing either a 4K or 8K MXC. The program will ask what size is desired. Key in the appropriate number only (RETURN/EXEC does not have to be keyed).

The test itself floods the RAM with a code of HEX (00). The data is then read back and compared with the data sent. If there were no errors, the HEX code data is incremented and the pattern continues all the way to HEX (FF).

If an error occurs, the data sent and received along with the bad location's address will be displayed. RAM addresses 2000-2FFF are in L36-L43 and locations 3000-3FFF are in L22-L29.

SF'05, SF'21 - USART TEST

The USART test is set up such that the 210-7294 grandaughter board (terminals 5-8) can be tested along with the first 4 terminals on the 210-7290 PCB. The RS-232-C loop back connector (reference section 4.1) should be placed on the desired terminal's female connector. The remaining connectors may be left open.

NOTE:

The baud rate switches MUST be set for 9600. The other rates (L2-L19) should be checked with an oscilloscope.

The test will first ask for the terminal's number to be entered. RETURN/EXEC need not be keyed.

The USART test sends data equivalent to HEX (00) to HEX (FF) to the USART. The data is read back and checked for accuracy. Along with the USART, two other related latches are tested. They are L40 (receiver ready status) and L4 (data set ready status). These are checked prior to the data being read from the USART. The interrupt circuit (L32 and L34) is also monitored for the reception of data by the USART.

SF'15 - CYCLE TEST

This option will run the 5 major tests (IB, OB, AB, RAM, USART) one after the other automatically. It only operates in the stop on error mode. The program will ask for the RAM size and terminal number to be entered. RETURN/EXEC need not be keyed. Only one terminal (USART) may be checked at a time.

SF'31 - LONG RAM TEST

This RAM test is extremely long (approx. 30 min. for 4K and 1 hour for 8K). It operates in the loop on error mode. The correct RAM size will have to be entered at the beginning of the test.

Although the test is time consuming, it is an excellent check for the 4050 one bit RAMs that are used in the 2236 MXC. The RAM is flooded with "ones", then a "zero" is put into a single location. The entire RAM is read for data integrity and then the location that contains the zero is changed back to the one state and the next location is put to zero. This pattern continues until a zero has been placed in every location. The second half of the test is just the opposite. The RAM is flooded with zeroes and a one is placed into a single location.

NOTE:

Because it may be necessary to step through the MXC microprogram during a test while allowing the 2200 BASIC program to run free, the RBI (ready/busy) line on the controller had to be monitored. Therefore, if an error is detected or the program hangs up, it may not be solely due to a malfunction in the circuitry that the test is designed to check. The ready/busy circuit could be what is causing the problem.

```
                    2    *          2236 TERMINAL DEBUG PROGRAM


                    4    *          CPU TEST


     0000           6               ORG    X'0000'
0000 310021         7    POWERON    LXI    SP,X'2100' SET STACK POINTER TO 2100
0003 DB02           8               IN     X'02'      STATUS BYTE IN
0005 07             9               RLC                SHIFT DATA LEFT 3 BITS
0006 07            10               RLC
0007 07            11               RLC
0008 E607          12               ANI    X'07'      DETERMINE BAUD RATE
000A 21C303        13               LXI    H,RATETBLE HL = ROM ADDR OF RATE TABLE
000D 85            14               ADD    L          ADD A TO L RESULT TO A
000E 6F            15               MOV    L,A        HL = ROM ADDR OF SPECIFIED RATE
000F 7E            16               MOV    A,M        READ RATE
0010 D301          17               OUT    X'01'      BAUD RATE OUT
0012 3E19          18               MVI    A,X'19'    RESET KBD STATUS,ENABLE CRT
0014 D302          19               OUT    X'02'      CLEAR BYTE OUT
0016 3E03          20               MVI    A,X'03'    SET CLEAR SCREEN
0018 CDB602        21               CALL   OUTCRT
001B 215903        22    READY      LXI    H,BEGIN    HL = ROM ADDR OF "R E A D Y :"
001E CDC202        23               CALL   TITLE

0021 DB02          25               IN     X'02'      STATUS BYTE IN
0023 E604          26               ANI    X'04'      TEST FOR KEYBOARD DATA
0025 CA1B00        27               JZ     READY      NO,JUMP
0028 DB01          28               IN     X'01'      KEYBOARD DATA IN
002A EE81          29               XRI    X'81'      TEST FOR SF'0
002C CA4E00        30               JZ     KBDTEST    YES,JUMP
002F DB01          31               IN     X'01'      KEYBOARD DATA IN
0031 EE82          32               XRI    X'82'      TEST FOR SF'1
0033 CAA700        33               JZ     UARTTEST   YES,JUMP
0036 DB01          34               IN     X'01'      KEYBOARD DATA IN
0038 EE83          35               XRI    X'83'      TEST FOR SF'2
003A CA3101        36               JZ     RAMTEST    YES,JUMP
003D DB01          37               IN     X'01'      KEYBOARD DATA IN
003F EE84          38               XRI    X'84'      TEST FOR SF'3
0041 CA1502        39               JZ     CRTTEST    YES, JUMP
0044 DB01          40               IN     X'01'      KEYBOARD DATA IN
0046 EE85          41               XRI    X'85'      TEST FOR SF'4
0048 CA7702        42               JZ     PRINTER    YES, JUMP
004B C31B00        43               JMP    READY      JUMP TO START READY TEST
```

```
004E 21FB02  46  KBDTEST   LXI   H,KBD      HL = ROM ADDR OF KEYBOARD TITLE
0051 CDC202  47            CALL  TITLE
0054 3E0D    48            MVI   A,X'0D'    SET CARRIAGE RETURN
0056 CDB602  49            CALL  OUTCRT
0059 3E0A    50            MVI   A,X'0A'    SET LINE FEED
005B CDB602  51            CALL  OUTCRT
005E 0600    52  COUNT1     MVI   B,X'00'    CHARACTER COUNT = 0
0060 DB02    53  KBDTEST1  IN    X'02'      STATUS BYTE IN
0062 E604    54            ANI   X'04'      TEST FOR KEYBOARD DATA
0064 CA6000  55            JZ    KBDTEST1   NO,JUMP
0067 DB01    56            IN    X'01'      KEYBOARD DATA IN
0069 4F      57            MOV   C,A        SAVE KEYBOARD DATA
006A EEC9    58            XRI   X'C9'      TEST FOR RESET
006C CA0000  59            JZ    POWERON    YES,JUMP
006F 79      60            MOV   A,C        RETRIEVE KEYBOARD DATA
0070 EE49    61            XRI   X'49'      TEST FOR RESET(SHIFTED)
0072 CA0000  62            JZ    POWERON    YES,JUMP
0075 79      63            MOV   A,C        RETRIEVE KEYBOARD DATA
0076 EEAA    64            XRI   X'AA'      TEST FOR "<"
0078 CA8400  65            JZ    LESS       YES,JUMP
007B 79      66            MOV   A,C        RETRIEVE KEYBOARD DATA
007C EE55    67            XRI   X'55'      TEST FOR SHIFTED "."
007E CA8E00  68            JZ    PERIOD     YES,JUMP
0081 C36000  69            JMP   KBDTEST1
0084 21FA02  70  LESS      LXI   H,X'02FA'  H,L = CONVERSION TABLE
0087 7E      71            MOV   A,M        ASCII CODE -> A
0088 CDB602  72            CALL  OUTCRT
008B C39800  73            JMP   COUNTUP
008E 21F902  74  PERIOD    LXI   H,X'02F9'  H,L = CONVERSION TABLE
0091 7E      75            MOV   A,M        ASCII CODE -> A
0092 CDB602  76            CALL  OUTCRT
0095 C39800  77            JMP   COUNTUP
0098 04      78  COUNTUP    INR   B          ADD 1 TO CHARACTER COUNT
0099 3E50    79            MVI   A,X'50'    SET A = 80
009B B8      80            CMP   B          TEST FOR CHARACTER COUNT = 80
009C C26000  81            JNZ   KBDTEST1   NO,JUMP
009F 3E0A    82            MVI   A,X'0A'    SET LINE FEED
00A1 CDB602  83            CALL  OUTCRT
00A4 C35E00  84            JMP   COUNT1     JUMP TO START NEXT LINE
```

```
00A7 212E03   87  UARTTEST LXI   H,UART    HL = ROM ADD OF "UART TEST.....LOOP:"
00AA CDC202   88           CALL  TITLE
00AD 0600     89           MVI   B,X'00'   SET DATA = 00
00AF 210000   90           LXI   H,X'0000' RESET LOOP COUNTER
00B2 DB02     91  TRANSRDY IN    X'02'     STATUS BYTE IN
00B4 E601     92           ANI   X'01'     UART READY TO TRANSMIT?
00B6 CAB200   93           JZ    TRANSRDY  NO,JUMP
00B9 78       94           MOV   A,B       DATA -> A
00BA D300     95           OUT   X'00'     DATA TO UART
00BC DB02     96  RCVRRDY  IN    X'02'     STATUS BYTE IN
00BE E602     97           ANI   X'02'     TEST FOR UART RECEIVED DATA?
00C0 C2BC00   98           JNZ   RCVRRDY   NO,JUMP
00C3 DB00     99           IN    X'00'     UART DATA IN
00C5 B8       100          CMP   B         COMPARE DATA RECEIVED WITH DATA SENT
00C6 C2FB00   101          JNZ   ERROR     JUMP IF ERROR
00C9 04       102          INR   B         INCREMENT DATA
00CA 3EFF     103          MVI   A,X'FF'   SET A = FF
00CC B8       104          CMP   B         TEST FOR END OF LOOP
00CD CAE100   105          JZ    LOOP1     YES,JUMP
00D0 DB02     106          IN    X'02'     STATUS BYTE IN
00D2 E604     107          ANI   X'04'     TEST FOR KEYBOARD DATA
00D4 CAB200   108          JZ    TRANSRDY  NO, JUMP
00D7 DB01     109          IN    X'01'     KEYBOARD DATA IN?
00D9 EEC9     110          XRI   X'C9'     TEST FOR RESET
00DB CA0000   111          JZ    POWERON   YES,JUMP
00DE C3B200   112          JMP   TRANSRDY  SEND NEXT BYTE

00E1 23       114 LOOP1    INX   H         INCREMENT LOOP COUNTER
00E2 7C       115          MOV   A,H       H -> A
00E3 CDD802   116          CALL  HEXASCII
00E6 7D       117          MOV   A,L       L -> A
00E7 CDD802   118          CALL  HEXASCII
00EA 3E08     119          MVI   A,X'08'   BACKSPACE
00EC CDB602   120          CALL  OUTCRT
00EF CDB602   121          CALL  OUTCRT
00F2 CDB602   122          CALL  OUTCRT
00F5 CDB602   123          CALL  OUTCRT
00F8 C3B200   124          JMP   TRANSRDY  JUMP TO START NEXT LOOP

00FB 4F       126 ERROR    MOV   C,A       DATA RECEIVED -> C
00FC 216003   127          LXI   H,UARTERR HL = ROM ADDR OF ERROR LABEL
00FF CDCD02   128          CALL  ERRLABEL
0102 78       129          MOV   A,B       DATA SENT -> A
0103 C5       130          PUSH  B         SAVE DATA SENT AND RECEIVED
0104 CDD802   131          CALL  HEXASCII
0107 3E0A     132          MVI   A,X'0A'   LINE FEED
0109 CDB602   133          CALL  OUTCRT
010C 3E0A     134          MVI   A,X'0A'   LINE FEED
010E CDB602   135          CALL  OUTCRT
0111 3E08     136          MVI   A,X'08'   CURSOR LEFT
0113 CDB602   137          CALL  OUTCRT
0116 3E08     138          MVI   A,X'08'   CURSOR LEFT
0118 CDB602   139          CALL  OUTCRT
011B C1       140          POP   B         RETRIEVE DATA SENT AND RECEIVED
011C 79       141          MOV   A,C       DATA RECEIVED -> A
```

```
011D CDD802   142           CALL   HEXASCII
0120 DB02     143 KBDTEST2  IN     X'02'     STATUS BYTE IN
0122 E604     144           ANI    X'04'     TEST FOR KEYBOARD DATA
0124 CA2001   145           JZ     KBDTEST2  NO,JUMP
0127 DB01     146           IN     X'01'     KEYBOARD DATA IN
0129 EEC9     147           XRI    X'C9'     TEST FOR RESET
012B CA0000   148           JZ     POWERON
012E C32001   149           JMP    KBDTEST2
```

```
0131 214403   152 RAMTEST  LXI   H,RAM     HL = ROM ADDR OF "RAM TEST.....LOOP:"
0134 CDC202   153          CALL  TITLE
0137 010000   154          LXI   B,X'0000' DATA = 00
013A 110000   155          LXI   D,X'0000' CLEAR LOOP COUNTER
013D 310021   156 AGAIN    LXI   SP,X'2100' SET STACK POINTER TO 2100
0140 C5       157 LOOP     PUSH  B         DATA -> RAM
0141 21FFDF   158          LXI   H,X'DFFF' HL = DFFF
0144 39       159          DAD   SP        TEST FOR BEGINNING OF RAM
0145 DA4001   160          JC    LOOP      NO, JUMP
0148 78       161          MOV   A,B       DATA -> A
0149 C1       162 LOOP2    POP   B         RAM DATA -> B,C
014A B8       163          CMP   B         TEST FOR A = B
014B C26B01   164          JNZ   ERROR1    NO, JUMP
014E B9       165          CMP   C         TEST FOR A = C
014F C28501   166          JNZ   ERROR2    NO, JUMP
0152 2100DF   167          LXI   H,X'DF00' HL = DF00
0155 39       168          DAD   SP        TEST FOR END OF RAM
0156 D24901   169          JNC   LOOP2     NO, JUMP
0159 3EFF     170          MVI   A,X'FF'   FF -> A
015B B8       171          CMP   B         TEST FOR END OF LOOP
015C CACE01   172          JZ    WAIT2     YES, JUMP
015F DB01     173          IN    X'01'     KEYBOARD DATA IN
0161 EEC9     174          XRI   X'C9'     TEST FOR RESET
0163 CA0000   175          JZ    POWERON   YES, JUMP
0166 04       176          INR   B         INCREMENT DATA
0167 0C       177          INR   C         INCREMENT DATA
0168 C33D01   178          JMP   AGAIN     JUMP TO SEND NEXT DATA

016B 57       180 ERROR1   MOV   D,A       SAVE DATA SENT
016C 210000   181          LXI   H,X'0000' CLEAR H,L
016F 39       182          DAD   SP        SP -> HL
0170 3EF9     183          MVI   A,X'F9'   SET A = F9
0172 BD       184          CMP   L         COMPARE L AND A
0173 DAFF01   185          JC    BAD       IF ERROR ABOVE LOCATION 20F5,JUMP
0176 310021   186          LXI   SP,X'2100' SET STACK POINTER TO 2100
0179 218703   187          LXI   H,RAMERR HL = ROM ADDR OF ERROR LABEL
017C CDCD02   188          CALL  ERRLABEL
017F CDA001   189          CALL  PRINT
0182 C3BD01   190          JMP   WAIT

0185 57       192 ERROR2   MOV   D,A       SAVE DATA SENT
0186 210000   193          LXI   H,X'0000' CLEAR H,L
0189 39       194          DAD   SP        SP -> H,L
018A 3EF9     195          MVI   A,X'F9'   SET A = F9
018C BD       196          CMP   L         COMPARE L AND A
018D DAFF01   197          JC    BAD       IF ERROR ABOVE LOCATION 20F5,JUMP
0190 310021   198          LXI   SP,X'2100' SET STACK POINTER TO 2100
0193 218703   199          LXI   H,RAMERR HL = ROM ADDR OF ERROR LABEL
0196 CDCD02   200          CALL  ERRLABEL
0199 41       201          MOV   B,C       SAVE DATA RECEIVED
019A CDA001   202          CALL  PRINT
019D C3BD01   203          JMP   WAIT
01A0 7A       204 PRINT    MOV   A,D       RETRIEVED DATA SENT
01A1 CDD802   205          CALL  HEXASCII
01A4 3E0A     206          MVI   A,X'0A'   LINE FEED
```

```
01A6 CDB602   207            CALL   OUTCRT
01A9 3E0A     208            MVI    A,X'0A'   LINE FEED
01AB CDB602   209            CALL   OUTCRT
01AE 3E08     210            MVI    A,X'08'   BACKSPACE
01B0 CDB602   211            CALL   OUTCRT
01B3 3E08     212            MVI    A,X'08'   BACKSPACE
01B5 CDB602   213            CALL   OUTCRT
01B8 78       214            MOV    A,B       DATA RECEIVED -> A
01B9 CDD802   215            CALL   HEXASCII
01BC C9       216            RET

01BD DB02     218 WAIT       IN     X'02'     STATUS BYTE IN
01BF E604     219            ANI    X'04'     TEST FOR KEYBOARD DATA
01C1 CABD01   220            JZ     WAIT      NO,JUMP
01C4 DB01     221            IN     X'01'     KEYBOARD DATA IN
01C6 EEC9     222            XRI    X'C9'     TEST FOR RESET
01C8 CA0000   223            JZ     POWERON   YES,JUMP
01CB C3BD01   224            JMP    WAIT
01CE DB02     225 WAIT2      IN     X'02'     STATUS BYTE IN
01D0 E604     226            ANI    X'04'     TEST FOR KEYBOARD DATA
01D2 C2F501   227            JNZ    YES       YES,JUMP
01D5 13       228 COUNT2     INX    D         INCREMENT LOOP COUNTER
01D6 D5       229            PUSH   D         SAVE LOOP COUNT
01D7 7A       230            MOV    A,D       D -> A
01D8 CDD802   231            CALL   HEXASCII
01DB D1       232            POP    D         RETRIEVE LOOP COUNT
01DC 7B       233            MOV    A,E       E -> A
01DD D5       234            PUSH   D         SAVE LOOP COUNT
01DE CDD802   235            CALL   HEXASCII
01E1 3E08     236            MVI    A,X'08'   BACKSPACE
01E3 CDB602   237            CALL   OUTCRT
01E6 CDB602   238            CALL   OUTCRT
01E9 CDB602   239            CALL   OUTCRT
01EC CDB602   240            CALL   OUTCRT
01EF D1       241            POP    D         RETRIEVE LOOP COUNT
01F0 04       242            INR    B         INCREMENT DATA
01F1 48       243            MOV    C,B       NEW DATA -> C
01F2 C33D01   244            JMP    AGAIN
01F5 DB01     245 YES        IN     X'01'     KEYBOARD DATA IN
01F7 EEC9     246            XRI    X'C9'     TEST FOR RESET
01F9 CA0000   247            JZ     POWERON   YES,JUMP
01FC C3D501   248            JMP    COUNT2

01FF 215509   250 BAD        LXI    H,X'0955' HL = ROM ADDR OF ERROR LABEL - 1
0202 23       251 INCREMNT   INX    H         HL = HL+1
0203 DB02     252 CRTCHECK   IN     X'02'     STATUS BYTE IN
0205 E610     253            ANI    X'10'     TEST FOR CRT READY
0207 C20302   254            JNZ    CRTCHECK  NO,JUMP
020A 7E       255            MOV    A,M       DATA -> A
020B D303     256            OUT    X'03'
020D EE21     257            XRI    X'21'     TEST FOR !
020F C20202   258            JNZ    INCREMNT  NO,JUMP
0212 C3BD01   259            JMP    WAIT
              260 *
```

```
0215 1610     3   CRTTEST   MVI   D,X'10'  SET DATA = 10
0217 010000   4   LOOPAGN   LXI   B,X'0000' CLEAR COUNTERS
021A 3E03     5             MVI   A,X'03'  SET CLEAR SCREEN
021C CDB602   6             CALL  OUTCRT
021F 7A       7   AGAIN1    MOV   A,D      DATA -> A
0220 CDB602   8             CALL  OUTCRT
0223 04       9             INR   B        INCREMENT CHARACTER COUNT
0224 3E50     10            MVI   A,X'50'  SET A = 80
0226 B8       11            CMP   B        TEST FOR CHARACTER COUNT = 80
0227 C21F02   12            JNZ   AGAIN1   NO, JUMP
022A 3E0A     13            MVI   A,X'0A'  SET LINE FEED
022C CDB602   14            CALL  OUTCRT
022F 0600     15            MVI   B,X'00'  RESET CHARACTER COUNT
0231 0C       16            INR   C        INCREMENT LINE COUNT
0232 3E18     17            MVI   A,X'18'  SET A = 24
0234 B9       18            CMP   C        TEST FOR LINE COUNT = 24
0235 C21F02   19            JNZ   AGAIN1   NO, JUMP
0238 7A       20  AGAIN2    MOV   A,D      DATA -> A
0239 CDB602   21            CALL  OUTCRT
023C 04       22            INR   B        INCREMENT CHARACTER COUNT
023D 3E4F     23            MVI   A,X'4F'  SET A = 79
023F B8       24            CMP   B        TEST FOR CHARACTER COUNT = 79
0240 C23802   25            JNZ   AGAIN2   NO, JUMP
0243 010000   26            LXI   B,X'0000' CLEAR DELAY COUNTER
0246 210080   27  DELAY     LXI   H,X'8000' SET DELAY COMPARE BYTES
0249 03       28            INX   B        INCREMENT DELAY COUNTER
024A 09       29            DAD   B        TEST FOR END OF DELAY
024B D24602   30            JNC   DELAY    NO, JUMP
024E DB01     31            IN    X'01'    KEYBOARD DATA IN
0250 EEC9     32            XRI   X'C9'    TEST FOR RESET
0252 CA0000   33            JZ    POWERON  YES, JUMP
0255 DB01     34            IN    X'01'    KEYBOARD DATA IN
0257 EE9E     35            XRI   X'9E'    TEST FOR HALT/STEP
0259 C26D02   36            JNZ   NEXTDATA NO, JUMP
025C DB02     37  KBDTEST3  IN    X'02'    STATUS BYTE IN
025E E604     38            ANI   X'04'    TEST FOR KEYBOARD DATA
0260 CA5C02   39            JZ    KBDTEST3 NO, JUMP
0263 DB01     40            IN    X'01'    KEYBOARD DATA IN
0265 EEC9     41            XRI   X'C9'    TEST FOR RESET
0267 CA0000   42            JZ    POWERON  YES, JUMP
026A C35C02   43            JMP   KBDTEST3 JUMP TO WAIT FOR RESET
026D 14       44  NEXTDATA  INR   D        INCREMENT DATA
026E 3E00     45            MVI   A,X'00'  00 -> A
0270 BA       46            CMP   D        TEST FOR LAST DATA
0271 CA1502   47            JZ    CRTTEST  YES, JUMP
0274 C31702   48            JMP   LOOPAGN  JUMP TO CONTINUE TEST
```

```
0277 3E1A     51  PRINTER  MVI   A,X'1A'   SET PRINTER ENABLE
0279 D302     52           OUT   X'02'     ENABLE PRINTER
027B 0620     53           MVI   B,X'20'   SET DATA = 20
027D 0E00     54           MVI   C,X'00'   CLEAR COUNTER
027F CDAB02   55  PRINT1   CALL  PRTRDY
0282 DB01     56           IN    X'01'     KEYBOARD DATA IN
0284 EEC9     57           XRI   X'C9'     TEST FOR RESET
0286 CA0000   58           JZ    POWERON   YES, JUMP
0289 04       59           INR   B         INCREMENT DATA
028A 0C       60           INR   C         INCREMENT COUNT
028B 3E7F     61           MVI   A,X'7F'   7F -> A
028D B8       62           CMP   B         TEST FOR END OF LOOP
028E CAA302   63           JZ    NEXTPRT   YES, JUMP
0291 3E50     64           MVI   A,X'50'   SET A = 80
0293 B9       65           CMP   C         TEST FOR END OF LINE
0294 C27F02   66           JNZ   PRINT1    NO, JUMP
0297 0E00     67           MVI   C,X'00'   RESET COUNTER
0299 50       68           MOV   D,B       SAVE DATA
029A 060D     69           MVI   B,X'0D'   SET CARRIAGE RETURN
029C CDAB02   70           CALL  PRTRDY
029F 42       71           MOV   B,D       RETRIEVE DATA
02A0 C37F02   72           JMP   PRINT1    JUMP TO PRINT NEXT CHARACTER
02A3 060D     73  NEXTPRT  MVI   B,X'0D'   SET CARRIAGE RETURN
02A5 CDAB02   74           CALL  PRTRDY
02A8 C37702   75           JMP   PRINTER   JUMP TO START TEST OVER

02AB DB02     77  PRTRDY   IN    X'02'     STATUS BYTE IN
02AD E608     78           ANI   X'08'     TEST FOR PRINTER READY
02AF C2AB02   79           JNZ   PRTRDY    NO, JUMP
02B2 78       80           MOV   A,B       DATA -> A
02B3 D303     81           OUT   X'03'
02B5 C9       82           RET
```

```
02B6 5F       85  OUTCRT   MOV   E,A       SAVE DATA IN A
02B7 DB02     86  OUTCRT1  IN    X'02'     STATUS BYTE IN
02B9 E610     87           ANI   X'10'     TEST FOR CRT READY
02BB C2B702   88           JNZ   OUTCRT1   NO,JUMP
02BE 7B       89           MOV   A,E       RETRIEVE DATA
02BF D303     90           OUT   X'03'
02C1 C9       91           RET


02C2 7E       93  TITLE    MOV   A,M       TITLE -> A
02C3 CDB602   94           CALL  OUTCRT
02C6 EE3A     95           XRI   X'3A'     TEST FOR COLON
02C8 23       96           INX   H         HL = HL+1
02C9 C2C202   97           JNZ   TITLE     NO COLON,JUMP
02CC C9       98           RET


02CD 7E      100  ERRLABEL MOV   A,M       LABEL -> A
02CE CDB602  101           CALL  OUTCRT
02D1 EE07    102           XRI   X'07'     TEST FOR BELL
02D3 23      103           INX   H         HL = HL+1
02D4 C2CD02  104           JNZ   ERRLABEL  NO BELL,JUMP
02D7 C9      105           RET


02D8 0E00    107  HEXASCII MVI   C,X'00'   CLEAR C
02DA 57      108           MOV   D,A       A -> D
02DB 07      109           RLC             ROTATE DATA IN A LEFT
02DC 07      110           RLC
02DD 07      111           RLC
02DE 07      112           RLC
02DF E60F    113  MASK     ANI   X'0F'     MASK OUT HIGH ORDER
02E1 F630    114           ORI   X'30'
02E3 FE3A    115           CPI   X'3A'     TEST FOR A > HEX(39)
02E5 DAEA02  116           JC    OUT1      A = OR < HEX(39),JUMP
02E8 C607    117           ADI   X'07'     A = A+7
02EA CDB602  118  OUT1     CALL  OUTCRT
02ED 0C      119           INR   C         C = C+1
02EE 3E02    120           MVI   A,X'02'
02F0 B9      121           CMP   C         TEST FOR COMPLETE
02F1 CAF802  122           JZ    RETURN
02F4 7A      123           MOV   A,D       D -> A
02F5 C3DF02  124           JMP   MASK
02F8 C9      125  RETURN   RET
             126  *
             127  *
             128  *
```

```
                        3   *          KEYBOARD TO ASCII TABLE

02F9 2E3C               5              DC     X'2E3C'

                        7   *          TITLE TABLE

02FB 034B45             9   KBD        DC     X'034B4559424F41524409'
     59424F
     415244
     09
0305 544553             10             DC     X'5445535409E2E2E2E2E2'
     535409E
     2E2E2E2
     2E
030F 09454E             11             DC     X'09454E5445522053484849'
     544552
     205348
     49
0319 465445             12             DC     X'4654454420222E22204F'
     442022
     2E2220
     4F
0323 522022             13             DC     X'5220223C22204F4E4C59'
     3C2220
     4F4E4C
     59
032D 3A                 14             DC     X'3A'

032E 035541             16  UART       DC     X'0355415254095445535409'
     525409
     544553
     54
0338 09E2E2             17             DC     X'09E2E2F2E2F094C4F4F'
     2E2E2E
     094C4F
     4F
0342 503A               18             DC     X'503A'


0344 035241             20  RAM        DC     X'0352414D09544553540909'
     4D0954
     455354
     09
034E 2E2E2E             21             DC     X'2E2E2E2E2E094C4F4F50'
     2E2E09
     4C4F4F
     50
0358 3A                 22             DC     X'3A'
```

```
0355 015245 414459 3A      24  BEGIN    DC    X'015245414459 3A'

                           26  *          ERROR LABEL

0360 034552 524F52 210D0A 0A 28  UARTERR  DC    X'034552524F52210D0A0A'
036A 444154 410953 454E54 09 29           DC    X'44415441095345 4E5409'
0374 3D0D0A 0A4441 544109 52 30           DC    X'3D0D0A0A444154410952'
037E 435644 093D0C 0C0907  31             DC    X'435644093D0C0C0907'

0387 034552 524F52 210D0A 0A 33  RAMERR   DC    X'034552524F52210D0A0A'
0391 444154 412049 4E544F 20 34           DC    X'444154412049 4E544F20'
039B 52414D 203D0D 0A0A44 41 35           DC    X'52414D203D0D0A0A4441'
03A5 544120 46524F 4D2052 41 36           DC    X'5441204652 4F4D205241'
03AF 4D203D 200C0C 07      37             DC    X'4D203D200C0C07'
03B6 035245 504C41 434520 52 38           DC    X'035245504C4143452052'
03C0 414D21                39             DC    X'414D21'

                           41  *          BAUD RATE TABLE

03C3 2292C9 E5F3FA         43  RATETBLE DC    X'2292C9E5F3FA'
                           44  *
```

REFERENCES

```
  AGAIN     (015D) --0168. 01F2
  AGAIN1    (021F) --0227. 023E
  AGAIN2    (0238) --0240
  BAD       (01FF) --0173. 018D
  BEGIN     (0359) --001B
  COUNT1    (005E) --00A4
  COUNT2    (01D5) --01FC
  COUNTUP   (0098) --008B. 0093
  CRTCHECK  (0203) --0207
  CRTTEST   (0215) --0041. 0271
  DELAY     (0246) --024B
  ERRLABEL  (02CD) --00FF. 017C. 0196. 02D4
  ERROR     (00FB) --00C6
  ERROR1    (016B) --0142
  ERROR2    (0185) --014F
  HEXASCII  (02D8) --00E3. 00E7. 0104. 011D. 01A1. 0189. 01D8. 01DE
  INCREMNT  (0202) --020F
  KBD       (02FB) --004E
  KBDTEST   (004E) --002C
  KBDTEST1  (0060) --0064. 0081. 009C
  KBDTEST2  (0120) --0124. 012E
  KBDTEST3  (025C) --0260. 026A
  LESS      (0084) --0078
  LOOP      (0140) --0145
  LOOP1     (00E1) --00CD
```

```
                    2   *          2236 MXC DEBUG PROGRAM


                    4   *          POWERON ROUTINE

      0000          6              ORG   X'0000'
0000  F3            7              DI                 DISABLE INTERRUPTS
0001  DB02          8              IN    X'02'        CLEAR CBS, OBS
0003  DB01          9   CBSCHEK    IN    X'01'        2200 INTERFACE STATUS IN
0005  E602         10              ANI   X'02'        TEST FOR CBS
0007  CA0300       11              JZ    CBSCHEK      NO, JUMP
000A  DB02         12              IN    X'02'        CLEAR CBS
000C  DB01         13   START      IN    X'01'        2200 INTERFACE STATUS IN
000E  E610         14              ANI   X'10'        TEST FOR CONTROLLER CARD ENABLED
0010  CA0C00       15              JZ    START        NO, JUMP


0013  D300         17   SFKEY      OUT   X'00'        CLEAR RESET STATUS BIT
0015  DB01         18              IN    X'01'        2200 INTERFACE STATUS IN
0017  E601         19              ANI   X'01'        TEST FOR OBS
0019  CA1300       20              JZ    SFKEY        NO, JUMP
001C  DB02         21              IN    X'02'        2200 DATA IN
001E  0600         22              MVI   B,X'00'      00 -> B
0020  B8           23              CMP   B            TEST FOR COMMUNICATION, STACK TEST
0021  CA5D00       24              JZ    STARTEST     YES, JUMP
0024  0603         25              MVI   B,X'03'      03 -> B
0026  B8           26              CMP   B            TEST FOR IB TEST
0027  CAFB00       27              JZ    IBTEST       YES, JUMP
002A  06FF         28              MVI   B,X'FF'      FF -> B
002C  B8           29              CMP   B            TEST FOR OB TEST
002D  CA2501       30              JZ    OBTEST       YES, JUMP
0030  06AA         31              MVI   B,X'AA'      AA -> B
0032  B8           32              CMP   B            TEST FOR AB TEST
0033  C33F00       33              JMP   SKIP         JUMP OVER INTERRUPT TRAP
0036  00           34              NOP
0037  00           35              NOP
0038  F3           36              DI                 DISABLE INTERRUPT
0039  310030       37              LXI   SP,X'3000'   RESET STACK POINTER
003C  C3C402       38              JMP   DSR          JUMP TO CONT. USART TEST
003F  CA4901       39   SKIP       JZ    ABTEST       YES, JUMP
0042  0644         40              MVI   B,X'44'      44 -> B
0044  B8           41              CMP   B            TEST FOR LONG 4K RAM TEST
0045  CA4C03       42              JZ    LRAMTST4     YES, JUMP
0048  0688         43              MVI   B,X'88'      88 -> B
004A  B8           44              CMP   B            TEST FOR LONG 8K RAM TEST
004B  CA5403       45              JZ    LRAMTST8     YES, JUMP
004E  06BB         46              MVI   B,X'BB'      BB -> B
0050  B8           47              CMP   B            TEST FOR SHORT 4K RAM TEST
0051  CADC01       48              JZ    RAM4K        YES, JUMP
0054  0677         49              MVI   B,X'77'      77 -> B
0056  B8           50              CMP   B            TEST FOR SHORT 8K RAM TEST
0057  CAE401       51              JZ    RAM8K        YES, JUMP
005A  C37302       52              JMP   USARTEST     JUMP TO TEST USART
```

```
                    54  *           COMMUNICATION CHECK

005D DB01           56  STARTEST IN     X'01'     2200 INTERFACE STATUS IN
005F E602           57           ANI    X'02'     TEST FOR CBS
0061 CA5D00         58           JZ     STARTEST  NO, JUMP
0064 DB02           59           IN     X'02'     CLEAR CBS
0066 DB01           60  READY    IN     X'01'     2200 INTERFACE STATUS IN
0068 E608           61           ANI    X'08'     TEST FOR CPU READY
006A CA6600         62           JZ     READY     NO, JUMP
006D 3EFF           63           MVI    A,X'FF'   SET BUSY BYTE
006F D307           64           OUT    X'07'     SET CONT. BUSY
0071 D301           65           OUT    X'01'     SEND IBS
0073 3E00           66           MVI    A,X'00'   SET READY BYTE
0075 D307           67           OUT    X'07'     SET CONT. READY
0077 DB01           68  CBSCHEK1 IN     X'01'     2200 INTERFACE STATUS IN
0079 E602           69           ANI    X'02'     TEST FOR CBS
007B CA7700         70           JZ     CBSCHEK1  NO, JUMP
007E DB02           71           IN     X'02'     CLEAR CBS
0080 DB01           72  CPURDY1  IN     X'01'     2200 INTERFACE STATUS IN
0082 E608           73           ANI    X'08'     TEST FOR CPU READY
0084 CA8000         74           JZ     CPURDY1   NO, JUMP
0087 3EFF           75           MVI    A,X'FF'   SET BUSY BYTE
0089 D307           76           OUT    X'07'     SET CONT. BUSY
008B D311           77           OUT    X'11'     SEND IBS WITH ENDI
008D 3E00           78           MVI    A,X'00'   SET READY BYTE
008F D307           79           OUT    X'07'     SET CONT. READY

                    81  *           STACK TEST

0091 310030         83           LXI    SP,X'3000' SET STACK POINTER
0094 1600           84           MVI    D,X'00'   CLEAR COUNT
0096 01AAAA         85  BEGIN    LXI    B,X'AAAA' DATA = AA
0099 C5             86           PUSH   B         DATA -> RAM
009A 14             87           INR    D         INCREMENT COUNT
009B 3E0A           88           MVI    A,X'0A'   0A -> A
009D BA             89           CMP    D         TEST FOR COUNT = 0A
009E C29600         90           JNZ    BEGIN     NO, JUMP
00A1 C1             91  POP      POP    B         RAM DATA -> B,C
00A2 3EAA           92           MVI    A,X'AA'   AA -> A
00A4 B8             93           CMP    B         TEST FOR RAM DATA OK
00A5 CAAC00         94           JZ     TESTC     YES, JUMP
00A8 60             95           MOV    H,B       SAVE DATA RECEIVED
00A9 C3FA00         96           JMP    HALT      JUMP TO SEND ERROR
00AC B9             97  TESTC    CMP    C         TEST FOR RAM DATA OK
00AD CAB400         98           JZ     COUNTCHK  YES, JUMP
00B0 61             99           MOV    H,C       SAVE DATA RECEIVED
00B1 C3FA00         100          JMP    HALT      JUMP TO SEND ERROR
00B4 14             101 COUNTCHK INR    D         INCREMENT COUNT
00B5 3E14           102          MVI    A,X'14'   14 -> A
00B7 BA             103          CMP    D         TEST FOR COUNT = 14
00B8 CABE00         104          JZ     STACK55   YES, JUMP
00BB C3A100         105          JMP    POP       JUMP TO READ NEXT LOCATIONS
```

```
LOOP2     (0149) --0156

LOOPAGN   (0217) --0274

MASK      (02DF) --02F5

NEXTDATA  (026D) --0259

NEXTPRT   (02A3) --028E

OUT1      (02CA) --02E5

OUTCRT    (02B6) --0018, 0056, 005B, 0088, 0092, 00A1, 00EC, 00EF, 00F2, 00F5,
                 --0109, 010E, 0113, 0118, 01A6, 01AB, 01B0, 01B5, 01E3, 01E6,
                 --01E9, 01EC, 021C, 0220, 022C, 0239, 02C3, 02CE, 02EA

OUTCRT1   (02B7) --02BB

PERIOD    (008E) --007E

POWERON   (0000) --006C, 0072, 00D8, 012B, 0163, 01C8, 01F9, 0252, 0267, 0286

PRINT     (01AC) --017F, 019A

PRINT1    (027F) --0294, 02A0

PRINTER   (0277) --0048, 02A8

PRTRDY    (02AB) --027F, 029C, 02A5, 02AF

RAM       (0344) --0131

RAMERR    (0387) --0179, 0193

RAMTEST   (0131) --003A

RATETBLE  (03C3) --000A

RCVRRDY   (00BC) --00C0

READY     (001B) --0025, 004B

RETURN    (02F8) --02F1

TITLE     (02C2) --001E, 0051, 00AA, 0134, 02C9

TRANSRDY  (00B2) --00B6, 00D4, 00DE, 00F8

UART      (032E) --00A7

UARTERR   (0360) --00FC

UARTTEST  (00A7) --0033
```

```
WAIT     (01BD) --0182, 019D, 01C1, 01CB, 0212
WAIT2.   (01CE) --015C
YES      (01F5) --01D2


NUMBER OF ERRORS IN ASSEMBLY  0
```

```
00BE 015555   107 STACK55  LXI   B,X'5555' DATA = 55
00C1 C5       108          PUSH  B         DATA -> RAM
00C2 14       109          INR   D         INCREMENT COUNT
00C3 3E1E     110          MVI   A,X'1E'   1E -> A
00C5 BA       111          CMP   D         TEST FOR COUNT = 1E
00C6 C2BE00   112          JNZ   STACK55   NO, JUMP
00C9 C1       113 POP1     POP   B         RAM DATA -> B,C
00CA 3E55     114          MVI   A,X'55'   55 -> A
00CC B8       115          CMP   B         TEST FOR RAM DATA OK
00CD CAD400   116          JZ    TESTC1    YES, JUMP
00D0 60       117          MOV   H,B       SAVE DATA RECEIVED
00D1 C3FA00   118          JMP   HALT      JUMP TO SEND ERROR


00D4 B9       120 TESTC1   CMP   C         TEST FOR RAM DATA OK
00D5 CADC00   121          JZ    CHKCOUNT  YES, JUMP
00D8 61       122          MOV   H,C       SAVE DATA RECEIVED
00D9 C3FA00   123          JMP   HALT      JUMP TO SEND ERROR
00DC 14       124 CHKCOUNT INR   D         INCREMENT COUNT
00DD 3E28     125          MVI   A,X'28'   28 -> A
00DF BA       126          CMP   D         TEST FOR COUNT = 28
00E0 CAE600   127          JZ    PRINT     YES, JUMP
00E3 C3C900   128          JMP   POP1      JUMP TO READ NEXT LOCATIONS

00E6 310030   130 PRINT    LXI   SP,X'3000' SET STACK POINTER
00E9 CDF704   131          CALL  CBSCHECK  WAIT FOR CBS
00EC DB02     132          IN    X'02'     CLEAR CBS
00EE DB01     133 CPURDY2  IN    X'01'     2200 INTERFACE STATUS IN
00F0 E608     134          ANI   X'08'     TEST FOR CPU READY
00F2 CAEE00   135          JZ    CPURDY2   NO, JUMP
00F5 D301     136          OUT   X'01'     SEND IBS
00F7 C31300   137          JMP   SFKEY     JUMP TO LOOK FOR SPECIAL FUNCTION

00FA 76       139 HALT     HLT             STOP
```

```
00FB 310030   142 IBTEST    LXI   SP,X'3000' SET STACK POINTER
00FE 0600     143           MVI   B,X'00'  SET DATA = 00
0100 3EFF     144 SENDIB    MVI   A,X'FF'  SET BUSY BYTE
0102 D307     145           OUT   X'07'    SET CONT. BUSY
0104 78       146           MOV   A,B      DATA -> A
0105 CD0A05   147           CALL  OUTCPU   DATA -> CPU
0108 CDF704   148           CALL  CBSCHECK WAIT FOR CBS
010B 04       149           INR   B        INCREMENT DATA
010C DB02     150           IN    X'02'    2200 DATA IN
010E 0E00     151           MVI   C,X'00'  00 -> C
0110 B9       152           CMP   C        TEST FOR ERROR IN LOOP MODE
0111 C21C01   153           JNZ   NEXTTEST NO, JUMP
0114 CDF704   154           CALL  CBSCHECK WAIT FOR CBS
0117 DB02     155           IN    X'02'    CLEAR CBS
0119 C3F900   156           JMP   IBTEST   JUMP TO START TEST OVER
011C 0EFF     157 NEXTTEST  MVI   C,X'FF'  FF -> C
011E B9       158           CMP   C        TEST FOR JUMP TO OB TEST
011F CA2501   159           JZ    OBTEST   YES, JUMP
0122 C30001   160           JMP   SENDIB   JUMP TO SEND NEXT DATA
```

```
0125 310030   163 OBTEST   LXI   SP,X'3000' SET STACK POINTER
0128 DB01     164          IN    X'01'      2200 INTERFACE STATUS IN
012A E608     165          ANI   X'08'      TEST FOR CPU READY
012C CA2501   166          JZ    OBTEST     NO, JUMP
012F 3EFF     167          MVI   A,X'FF'    SET BUSY BYTE
0131 D307     168          OUT   X'07'      SET CONT. BUSY
0133 DB02     169          IN    X'02'      2200 DATA IN
0135 D301     170          OUT   X'01'      2200 DATA ECHOED BACK
0137 3E00     171          MVI   A,X'00'    SET READY BYTE
0139 D307     172          OUT   X'07'      SET CONT. READY
013B CDF704   173          CALL  CBSCHECK   WAIT FOR CBS
013E DB02     174          IN    X'02'      2200 DATA IN
0140 06AA     175          MVI   B,X'AA'    AA -> A
0142 B8       176          CMP   B          TEST FOR JUMP TO AB TEST
0143 CA4901   177          JZ    ABTEST     YES. JUMP
0146 C32501   178          JMP   OBTEST     JUMP TO WAIT FOR NEXT DATA
```

```
0149 310030   181 ABTEST    LXI   SP,X'3000' SET STACK POINTER
014C 0630     182           MVI   B,X'30'  SET AB, ENABLE COMPARE BYTE
014E 0E20     183           MVI   C,X'20'  SET AB COMPARE BYTE
0150 CDF704   184 ABAGAIN1  CALL  CBSCHECK WAIT FOR CBS
0153 DB02     185           IN    X'02'    CLEAR CBS
0155 DB01     186           IN    X'01'    2200 INTERFACE STATUS IN
0157 E6F0     187           ANI   X'F0'    MASK OUT LOW ORDER
0159 58       188           CMP   B        TEST FOR ENABLE AND AB COMPARE
015A C28B01   189           JNZ   ABERROR1 NO, JUMP
015D DB03     190           IN    X'03'    ADDRESS BUS IN
015F E6E0     191           ANI   X'E0'    MASK OUT LOW ORDER, HIGH ORDER 1 BIT
0161 B9       192           CMP   C        TEST FOR AB COMPARE
0162 C2C101   193           JNZ   ABERROR2 NO, JUMP
0165 3EFF     194           MVI   A,X'FF'  SET 2200 CONTROL BYTE
0167 D307     195           OUT   X'07'    SEET CONT. BUSY
0169 CD0A05   196           CALL  OUTCPU   SEND IBS
016C CDF704   197           CALL  CBSCHECK
016F DB02     198           IN    X'02'    CLEAR CBS
0171 3EFF     199           MVI   A,X'FF'  SET BUSY BYTE
0173 D307     200           OUT   X'07'    SET CONT. BUSY
0175 CD0A05   201           CALL  OUTCPU   SEND IBS
0178 CDF704   202 CONT1     CALL  CBSCHECK WAIT FOR CBS
017B DB02     203           IN    X'02'    2200 DATA IN
017D 16FF     204           MVI   D,X'FF'  FF -> D
017F BA       205           CMP   D        TEST FOR ERROR IN LOOP MODE
0180 C28D01   206           JNZ   CONT     NO, JUMP
0183 3EFF     207           MVI   A,X'FF'  SET BUSY BYTE
0185 D307     208           OUT   X'07'    SET CONT. BUSY
0187 CD0A05   209           CALL  OUTCPU   SEND IBS
018A C34901   210           JMP   ABTEST   JUMP TO START TEST OVER
018D 1644     211 CONT      MVI   D,X'44'  44 -> D
018F BA       212           CMP   D        TEST FOR JUMP TO SHORT 4K RAM TEST
0190 CADC01   213           JZ    RAM4K    YES, JUMP
0193 1688     214           MVI   D,X'88'  88 -> D
0195 BA       215           CMP   D        TEST FOR JUMP TO SHORT 8K RAM TEST
0196 CAE401   216           JZ    RAM8K    YES, JUMP
0199 1655     217           MVI   D,X'55'  SET COMPARE BYTE
019B BA       218           CMP   D        TEST FOR END OF LOOP
019C C2A901   219           JNZ   CONT2    NO, JUMP
019F 3EFF     220           MVI   A,X'FF'  SET BUSY BYTE
01A1 D307     221           OUT   X'07'    SET CONT. BUSY
01A3 CD0A05   222           CALL  OUTCPU   SEND IBS
01A6 C34901   223           JMP   ABTEST   JUMP TO START TEST OVER
01A9 3E20     224 CONT2     MVI   A,X'20'  20 -> A
01AB 80       225           ADD   B        INCREMENT COMPARE BYTE
01AC 47       226           MOV   B,A      NEW COMPARE BYTE -> B
01AD 3E20     227           MVI   A,X'20'  20 -> A
01AF 81       228           ADD   C        INCREMENT COMPARE BYTE
01B0 4F       229           MOV   C,A      NEW COMPARE BYTE -> C
01B1 3EFF     230           MVI   A,X'FF'  SET BUSY BYTE
01B3 D307     231           OUT   X'07'    SET CONT. BUSY
01B5 CD0A05   232           CALL  OUTCPU   SEND IBS
01B8 C35001   233           JMP   ABAGAIN1 JUMP TO INPUT NEXT ADDRESS

01BB 5F       235 ABERROR1  MOV   E,A      SAVE ADDRESS DATA
01BC 3E0F     236           MVI   A,X'0F'  SET BUSY + CONTROL BYTE
```

```
01BE C3C701   237            JMP   EROUT
01C1 5F       238 ABERROR2   MOV   E,A       SAVE ADDRESS DATA
01C2 3EF0     239            MVI   A,X'F0'   SET BUSY + CONTROL BYTE
01C4 C3C701   240            JMP   EROUT
01C7 D307     241 EROUT      OUT   X'07'     SET CONT. BUSY
01C9 CD0A05   242            CALL  OUTCPU    SEND CONTROL BYTE
01CC CDF704   243            CALL  CBSCHECK
01CF DB02     244            IN    X'02'     CLEAR CBS
01D1 3EFF     245            MVI   A,X'FF'   SET BUSY BYTE
01D3 D307     246            OUT   X'07'     SET CONT. BUSY
01D5 7B       247            MOV   A,E       RETRIEVE ADDRESS DATA
01D6 CD0A05   248            CALL  OUTCPU    SEND DATA
01D9 C37801   249            JMP   CONT1     JUMP TO CONTINUE
              250 *
              251 *
              252 *
```

```
01DC 310030   3   RAM4K    LXI   SP,X'3000' SET STACK POINTER
01DF 1630     4             MVI   D,X'30'  SET RAM SIZE ID BYTE
01E1 C3E901   5             JMP   TESTRAM  JUMP TO START TEST
01E4 310040   6   RAM8K    LXI   SP,X'4000' SET STACK POINTER
01E7 1640     7             MVI   D,X'40'  SET RAM SIZE ID BYTE
01E9 010000   8   TESTRAM   LXI   B,X'0000' SET DATA = 00
01EC 58       9             MOV   E,B      DATA -> E
01ED 21FFDF   10  PUSH1     LXI   H,X'DFFF' DFFF -> H,L
01F0 C5       11            PUSH  B        DATA -> RAM
01F1 39       12            DAD   SP       TEST FOR BEGINNING OF RAM
01F2 DAED01   13            JC    PUSH1    NO, JUMP
01F5 7B       14  LOOPS4    MOV   A,E      DATA -> A
01F6 C1       15            POP   B        RAM DATA -> B,C
01F7 B8       16            CMP   B        TEST FOR DATA CORRECT
01F8 C23C02   17            JNZ   ERROR1   NO, JUMP
01FB B9       18            CMP   C        TEST FOR DATA CORRECT
01FC C24502   19            JNZ   ERROR2   NO. JUMP
01FF 3E30     20            MVI   A,X'30'  30 -> A
0201 BA       21            CMP   D        TEST FOR 4K RAM
0202 CA0B02   22            JZ    SET4K1   YES, JUMP
0205 2100C0   23            LXI   H,X'C000' C000 -> H,L
0208 C30E02   24            JMP   COMPARE2 JUMP TO TEST END
020B 2100D0   25  SET4K1    LXI   H,X'D000' D000 -> H,L
020E 39       26  COMPARE2  DAD   SP       TEST FOR END OF RAM
020F D2F501   27            JNC   LOOPS4   NO, JUMP
0212 3EFF     28            MVI   A,X'FF'  FF -> A
0214 BB       29            CMP   E        TEST FOR END OF LOOP
0215 CA2502   30            JZ    LOOPOUT  YES, JUMP
0218 DB01     31            IN    X'01'    2200 INTERFACE STATUS IN
021A E604     32            ANI   X'04'    TEST FOR RESET
021C C21300   33            JNZ   SFKEY    YES, JUMP
021F 1C       34            INR   E        INCREMENT DATA
0220 43       35            MOV   B,E      NEW DATA -> B
0221 4B       36            MOV   C,E      NEW DATA -> C
0222 C3ED01   37            JMP   PUSH1    JUMP TO WRITE NEXT DATA
0225 3EFF     38  LOOPOUT   MVI   A,X'FF'  BUSY BYTE
0227 D307     39            OUT   X'07'    SET CONT. BUSY
0229 3E00     40            MVI   A,X'00'  SET LOOP CONTROL BYTE
022B CD0A05   41            CALL  OUTCPU   SEND IBS
022E CDF704   42            CALL  CBSCHECK WAIT FOR CBS
0231 DB02     43            IN    X'02'    2200 DATA IN
0233 06AA     44            MVI   B,X'AA'  AA -> B
0235 B8       45            CMP   B        TEST FOR JUMP TO USART TEST
0236 C27302   46            JNZ   USARTEST YES, JUMP
0239 C3E901   47            JMP   TESTRAM  JUMP TO START NEXT LOOP
023C 3B       48  ERROR1    DCX   SP       DECREMENT STACK POINTER
023D 3B       49            DCX   SP       DECREMENT STACK POINTER
023E 210000   50            LXI   H,X'0000' CLEAR H,L
0241 39       51            DAD   SP       SP -> H,L
0242 C34B02   52            JMP   ERROUT   JUMP TO SEND ERROR
0245 3B       53  ERROR2    DCX   SP       DECREMENT STACK POINTER
0246 210000   54            LXI   H,X'0000' CLEAR H,L
0249 39       55            DAD   SP       SP -> H,L
024A 41       56            MOV   B,C      DATA RECEIVED -> B
024B 310030   57  ERROUT    LXI   SP,X'3000' SET STACK POINTER
024E 3EFF     58            MVI   A,X'FF'  SET ERROR CONTROL BYTE
```

```
0250 D307     59          OUT   X'07'     SET CONT. BUSY
0252 CD1D05   60          CALL  ERROROUT
0255 78       61          MOV   A,E       DATA SENT -> A
0256 CD1D05   62          CALL  ERROROUT
0259 78       63          MOV   A,B       DATA RECEIVED -> A
025A CD1D05   64          CALL  ERROROUT
025D 7C       65          MOV   A,H       FIRST BYTE OF RAM ADDR -> A
025E CD1D05   66          CALL  ERROROUT
0261 7D       67          MOV   A,L       SECOND BYTE OF RAM ADDR -> A
0262 CD0A05   68          CALL  OUTCPU    SEND IBS
0265 CDF704   69          CALL  CBSCHECK  WAIT FOR CBS
0268 DB02     70          IN    X'02'     CLEAR CBS
026A 3E30     71          MVI   A,X'30'   30 -> A
026C BA       72          CMP   D         TEST FOR 4K RAM
026D CADC01   73          JZ    RAM4K     YES. JUMP TO START TEST OVER
0270 C3E401   74          JMP   RAM8K     JUMP TO START TEST OVER
```

```
0273 4F       77  USARTEST MOV   C,A        SAVE SELECTED USART
0274 79       78  AGAIN    MOV   A,C        SELECTED USART -> A
0275 47       79           MOV   B,A        SELECTED USART -> B
0276 D305     80  SELECT   OUT   X'05'      SELECT USART
0278 3E00     81           MVI   A,X'00'    ZERO ACCUMULATOR
027A D30E     82           OUT   X'0E'      SET USART TO COMMAND MODE
027C D30E     83           OUT   X'0E'      SET USART TO COMMAND MODE
027E 3E40     84           MVI   A,X'40'    SET INTERNAL RESET BIT
0280 D30E     85           OUT   X'0E'      CONTROL BYTE -> USART
0282 3E5E     86           MVI   A,X'5E'    SET MODE(1 STOP,ODD PAR.,8 DATA,/16)
0284 D30E     87           OUT   X'0E'      MODE BYTE -> USART
0286 78       88           MOV   A,B        SELECTED USART -> A
0287 07       89           RLC              SHIFT DATA LEFT 1 BIT
0288 B9       90           CMP   C          TEST FOR EVERY USART RESET
0289 CA9002   91           JZ    LOOPAGN    YES, JUMP
028C 47       92           MOV   B,A        NEXT USART -> B
028D C37602   93           JMP   SELECT     JUMP TO RESET NEXT USART
0290 310030   94  LOOPAGN  LXI   SP,X'3000' SET STACK POINTER
0293 0600     95           MVI   B,X'00'    DATA = 00
0295 110000   96           LXI   D,X'0000'  CLEAR LOOP COUNTER
0298 79       97           MOV   A,C        SELECTED USART -> A
0299 D305     98           OUT   X'05'      SELECT USART
029B 3E37     99           MVI   A,X'37'    SET CONTROL(RTS,RES ERR,ENBL RCV+XMIT
029D D30E     100          OUT   X'0E'      CONTROL BYTE TO USART
029F DB0E     101 XMITRDY  IN    X'0E'      USART STATUS BYTE IN
02A1 E605     102          ANI   X'05'      TEST FOR XMITR EMPTY AND READY
02A3 CA9F02   103          JZ    XMITRDY    NO, JUMP
02A6 DB06     104          IN    X'06'      CLEAR RCVR DATA BUFFER
02A8 FB       105          EI               ENABLE INTERRUPT
02A9 78       106          MOV   A,B        DATA -> A
02AA D306     107          OUT   X'06'      DATA -> USART
02AC 1E01     108          MVI   E,X'01'    START DELAY COUNT
02AE 1C       109 DELAY    INR   E          INCREMENT COUNT
02AF C2AE02   110          JNZ   DELAY      TEST FOR END OF DELAY
02B2 F3       111          DI               DISABLE INTERRUPT
02B3 3EFF     112          MVI   A,X'FF'    SET BUSY BYTE
02B5 D307     113          OUT   X'07'      SET CONT. BUSY
02B7 3EAA     114          MVI   A,X'AA'    SET INTERRUPT ERROR CONTROL BYTE
02B9 CD0A05   115          CALL  OUTCPU     SEND IBS
02BC CDF704   116          CALL  CBSCHECK   WAIT FOR CBS
02BF DB02     117          IN    X'02'      CLEAR CBS
02C1 C37302   118          JMP   USARTEST   JUMP TO START TEST OVER
02C4 DB00     119 DSR      IN    X'00'      MODEM STATUS IN
02C6 5F       120          MOV   E,A        SAVE STATUS
02C7 79       121          MOV   A,C        SELECTED USART -> A
02C8 C6F0     122          ADI   X'F0'      TEST FOR 8 PORT
02CA DAD302   123          JC    PORT8      YES, JUMP
02CD 7B       124          MOV   A,E        RETRIEVE MODEM STATUS
02CE E60F     125          ANI   X'0F'      MASK OUT HIGH ORDER
02D0 C3D402   126          JMP   TESTDSR    JUMP TO TEST DSR
02D3 7B       127 PORT8    MOV   A,E        RETRIEVE MODEM STATUS
02D4 B9       128 TESTDSR  CMP   C          TEST FOR DATA SET READY = SEL. USART
02D5 C22C03   129          JNZ   DSRERROR   NO, JUMP
02D8 DB04     130          IN    X'04'      RECEIVER READY STATUS IN
02DA 5F       131          MOV   E,A        SAVE STATUS
02DB 79       132          MOV   A,C        SELECTED USART -> A
```

```
02DC C6F0     133           ADI   X'F0'     TEST FOR 8 PORT
02DE DAE702   134           JC    PORT8#1   YES, JUMP
02E1 7B       135           MOV   A,E       RETRIEVE RCVR READY STATUS
02E2 E60F     136           ANI   X'0F'     MASK OUT HIGH ORDER
02E4 C3E802   137           JMP   TESTRXRY  JUMP TO TEST RECEIVER READY
02E7 7B       138 PORT8#1   MOV   A,E       RETRIEVE RCVR READY STATUS
02E8 B9       139 TESTRXRY  CMP   C         TEST FOR RCVR READY = SELECTED USART
02E9 C23603   140           JNZ   RCVRDYER  NO, JUMP
02EC DB06     141           IN    X'06'     USART DATA IN
02EE B8       142           CMP   B         TEST FOR DATA SENT = DATA RECEIVED
02EF C21303   143           JNZ   USARTERR  NO, JUMP
02F2 04       144           INR   B         INCREMENT DATA
02F3 3EFF     145           MVI   A,X'FF'   FF -> A
02F5 B8       146           CMP   B         TEST FOR END OF LOOP
02F6 CAFC02   147           JZ    LOOP#1    YES, JUMP
02F9 C39F02   148           JMP   XMITRDY   JUMP TO SEND NEXT BYTE

02FC 3EFF     150 LOOP#1    MVI   A,X'FF'   SET BUSY BYTE
02FE D307     151           OUT   X'07'     SET CONT. BUSY
0300 3E00     152           MVI   A,X'00'   SET LOOP CONTROL BYTE
0302 CD0A05   153           CALL  OUTCPU    SEND IBS
0305 CDF704   154           CALL  CBSCHECK  WAIT FOR CBS
0308 DB02     155           IN    X'02'     2200 DATA IN
030A 1E55     156           MVI   E,X'55'   55 -> E
030C BB       157           CMP   E         TEST FOR JUMP TO IB TEST
030D CAFB00   158           JZ    IBTEST    YES, JUMP
0310 C39002   159           JMP   LOOPAGN   JUMP TO START NEXT LOOP

0313 60       161 USARTERR  MOV   H,B       DATA SENT -> H
0314 6F       162           MOV   L,A       DATA RECEIVED -> L
0315 3EFF     163           MVI   A,X'FF'   SET ERROR CONTROL BYTE
0317 D307     164           OUT   X'07'     SET CONT. BUSY
0319 CD1D05   165           CALL  ERROROUT  SEND IBS
031C 7C       166           MOV   A,H       DATA SENT -> A
031D CD1D05   167           CALL  ERROROUT  SEND IBS
0320 7D       168           MOV   A,L       DATA RECEIVED -> A
0321 CD0A05   169           CALL  OUTCPU    SEND IBS
0324 CDF704   170           CALL  CBSCHECK  WAIT FOR CBS
0327 DB02     171           IN    X'02'     CLEAR CBS
0329 C37402   172           JMP   AGAIN     JUMP TO START TEST OVER
032C 5F       173 DSRERROR  MOV   E,A       SAVE MODEM STATUS
032D 3EFF     174           MVI   A,X'FF'   SET BUSY BYTE
032F D307     175           OUT   X'07'     SET CONT. BUSY
0331 3EF0     176           MVI   A,X'F0'   SET DSR ERROR CONTROL BYTE
0333 C33D03   177           JMP   SENDERR
0336 5F       178 RCVRDYER  MOV   E,A       SAVE RECEIVER READY STATUS
0337 3EFF     179           MVI   A,X'FF'   SET BUSY BYTE
0339 D307     180           OUT   X'07'     SET CONT. BUSY
033B 3E0F     181           MVI   A,X'0F'   SET RCVR READY ERROR CONTROL BYTE
033D CD1D05   182 SENDERR   CALL  ERROROUT  SEND IBS
0340 7B       183           MOV   A,E       RETRIEVE ERROR DATA
0341 CD0A05   184           CALL  OUTCPU    SEND IBS
0344 CDF704   185           CALL  CBSCHECK  WAIT FOR CBS
0347 DB02     186           IN    X'02'     CLEAR CBS
0349 C37402   187           JMP   AGAIN     JUMP TO START TEST OVER
              188 *
```

```
034C 310030   3   LRAMTST4 LXI   SP,X'3000' SET STACK POINTER
034F 1630     4           MVI   D,X'30'  SET RAM SIZE ID BYTE
0351 C35903   5           JMP   RAMTEST  JUMP TO START TEST
0354 310040   6   LRAMTST8 LXI   SP,X'4000' SET STACK POINTER
0357 1640     7           MVI   D,X'40'  SET RAM SIZE ID BYTE
0359 01FFFF   8   RAMTEST  LXI   B,X'FFFF' DATA = FF
035C 21FFDF   9   PUSH     LXI   H,X'DFFF' DFFF -> H,L
035F C5       10          PUSH  B        DATA -> RAM
0360 39       11          DAD   SP       TEST FOR BEGINNING OF RAM
0361 DA5C03   12          JC    PUSH     NO, JUMP
0364 4A       13          MOV   C,D      SAVE ID BYTE
0365 110020   14          LXI   D,X'2000' D,E = 00 BYTE LOCATION
0368 210020   15          LXI   H,X'2000' SET RAM ADDRESS = 2000
036B 3E00     16  WRITE00  MVI   A,X'00'  DATA = 00
036D 77       17          MOV   M,A      DATA -> RAM
036E 210020   18          LXI   H,X'2000' SET RAM ADDRESS = 2000
0371 3E30     19  CHECK    MVI   A,X'30'  30 -> A
0373 BC       20          CMP   H        TEST FOR H = 30
0374 CA9603   21          JZ    OUT      YES, JUMP
0377 46       22          MOV   B,M      RAM DATA -> B
0378 7A       23          MOV   A,D      FIRST BYTE OF 00 LOACATIN -> A
0379 BC       24          CMP   H        TEST FOR H = D
037A C28C03   25          JNZ   SETFF    NO,JUMP
037D 7B       26          MOV   A,E      SECOND BYTE OF 00 LOCATION -> A
037E BD       27          CMP   L        TEST FOR LOCATION READ = 00 BYTE LOC.
037F C28C03   28          JNZ   SETFF    NO, JUMP
0382 3E00     29          MVI   A,X'00'  00 -> A
0384 B8       30          CMP   B        TEST FOR DATA READ = 00
0385 C2CC04   31          JNZ   RAMERROR NO, JUMP
0388 23       32          INX   H        INCREMENT RAM ADDRESS
0389 C37103   33          JMP   CHECK    JUMP TO READ NEXT LOCATION
038C 3EFF     34  SETFF    MVI   A,X'FF'  FF -> A
038E B8       35          CMP   B        TEST FOR DATA READ = FF
038F C2CC04   36          JNZ   RAMERROR NO, JUMP
0392 23       37          INX   H        INCREMENT RAM ADDRESS
0393 C37103   38          JMP   CHECK    JUMP TO READ NEXT LOCATION
0396 3E30     39  OUT      MVI   A,X'30'  30 -> A
0398 B9       40          CMP   C        TEST FOR 4K RAM
0399 CAD203   41          JZ    MOVE     YES, JUMP
039C 62       42          MOV   H,D      FIRST BYTE OF 00 BYTE LOCATION -> H
039D 6B       43          MOV   L,E      SECOND BYTE OF 00 BYTE LOCATION -> L
039E 3EFF     44          MVI   A,X'FF'  FF -> A
03A0 77       45          MOV   M,A      FF -> 00 BYTE LOCATION
03A1 3E10     46          MVI   A,X'10'  10 -> A
03A3 B2       47          ORA   D        SET 00 BYTE LOCATION
03A4 57       48          MOV   D,A      D,E = 00 BYTE LOCATION
03A5 62       49          MOV   H,D      SET RAM ADDRESS = 00 BYTE LOCATION
03A6 6B       50          MOV   L,E      SET RAM ADDRESS = 00 BYTE LOCATION
03A7 3E00     51          MVI   A,X'00'  DATA = 00
03A9 77       52          MOV   M,A      DATA -> RAM
03AA 210030   53          LXI   H,X'3000' SET RAM ADDRESS = 3000
03AD 3E40     54  CHECK2   MVI   A,X'40'  40 -> A
03AF BC       55          CMP   H        TEST FOR H = 40
03B0 CAD203   56          JZ    MOVE     YES, JUMP
03B3 46       57          MOV   B,M      RAM DATA -> B
03B4 7A       58          MOV   A,D      FIRST BYTE OF 00 LOCATION -> A
```

```
03B5 BC       59             CMP   H          TEST FOR H = D
03B6 C2C803   60             JNZ   SETFF1     NO, JUMP
03B9 7B       61             MOV   A,E        SECOND BYTE OF 00 LOCATION -> A
03BA BD       62             CMP   L          TEST FOR LOCATION READ = 00 BYTE LOC.
03BB C2C803   63             JNZ   SETFF1     NO, JUMP
03BE 3E00     64             MVI   A,X'00'    00 -> A
03C0 B8       65             CMP   B          TEST FOR DAT READ = 00
03C1 C2CC04   66             JNZ   RAMERROR   NO, JUMP
03C4 23       67             INX   H          INCREMENT RAM ADDRESS
03C5 C3AD03   68             JMP   CHECK2     JUMP TO READ NEXT LOCATION
03C8 3EFF     69  SETFF1     MVI   A,X'FF'    FF -> A
03CA B8       70             CMP   B          TEST FOR DATA READ = FF
03CB C2CC04   71             JNZ   RAMERROR   NO, JUMP
03CE 23       72             INX   H          INCREMENT RAM ADDRESS
03CF C3AD03   73             JMP   CHECK2     JUMP TO READ NEXT LOCATION
03D2 62       74  MOVE       MOV   H,D        FIRST BYTE OF 00 BYTE LOCATION -> H
03D3 6B       75             MOV   L,E        SECOND BYTE OF 00 BYTE LOCATION -> L
03D4 3EFF     76             MVI   A,X'FF'    FF -> A
03D6 77       77             MOV   M,A        FF -> 00 BYTE LOCATION
03D7 13       78             INX   D          INCREMENT 00 BYTE LOCATION
03D8 3E30     79             MVI   A,X'30'    30 -> A
03DA B9       80             CMP   C          TEST FOR 4K RAM TEST
03DB C2E103   81             JNZ   SET40      NO, JUMP
03DE C3E303   82             JMP   COMPARE
03E1 3E40     83  SET40      MVI   A,X'40'    40 -> A
03E3 BA       84  COMPARE    CMP   D          TEST FOR D = 30, 40
03E4 CAFD03   85             JZ    FLOOD00    YES, JUMP
03E7 DB01     86             IN    X'01'      2200 INTERFACE STATUS IN
03E9 E604     87             ANI   X'04'      TEST FOR RESET
03EB C21300   88             JNZ   SFKEY      YES, JUMP
03EE 3E30     89             MVI   A,X'30'    30 -> A
03F0 B9       90             CMP   C          TEST FOR 4K RAM
03F1 CAF803   91             JZ    OUT1CONT   YES, JUMP
03F4 3EEF     92             MVI   A,X'EF'    EF -> A
03F6 A2       93             ANA   D          MASK OUT HIGH ORDER 1 BIT
03F7 57       94             MOV   D,A        A -> D
03F8 62       95  OUT1CONT   MOV   H,D        FIRST BYTE OF 00 LOCATION -> H
03F9 6B       96             MOV   L,E        SECOND BYTE OF 00 LOCATION -> L
03FA C36B03   97             JMP   WRITE00    JUMP TO WRITE 00 INTO NEXT LOCATION


03FD 3E30     99  FLOOD00    MVI   A,X'30'    30 -> A
03FF B9       100            CMP   C          TEST FOR 4K RAM TEST
0400 C20A04   101            JNZ   SET8K      NO, JUMP
0403 51       102            MOV   D,C        SAVE RAM SIZE ID BYTE
0404 310030   103            LXI   SP,X'3000' SET STACK POINTER
0407 C30E04   104            JMP   START00    JUMP TO START TEST
040A 51       105 SET8K      MOV   D,C        SAVE RAM SIZE ID BYTE
040B 310040   106            LXI   SP,X'4000' SET STACK POINTER
040E 010000   107 START00    LXI   B,X'0000'  DATA = 00
0411 21FFDF   108 PUSH2      LXI   H,X'DFFF'  DFFF -> H,L
0414 C5       109            PUSH  B          DATA -> RAM
0415 39       110            DAD   SP         TEST FOR BEGINNING OF RAM
0416 DA1104   111            JC    PUSH2      NO, JUMP
0419 4A       112            MOV   C,D        SAVE ID BYTE
041A 110020   113            LXI   D,X'2000'  D,E = FF BYTE LOCATION
041D 210020   114            LXI   H,X'2000'  SET RAM ADDRESS = 2000
0420 3EFF     115 WRITEFF.   MVI   A,X'FF'    DATA = FF
```

```
0422 77       116           MOV   M,A       DATA -> RAM
0423 210020   117           LXI   H,X'2000' SET RAM ADDRESS = 2000
0426 3E30     118 CHECK1    MVI   A,X'30'   30 -> A
0428 BC       119           CMP   H         TEST FOR H = 30
0429 CA4B04   120           JZ    OUT2      YES, JUMP
042C 46       121           MOV   B,M       RAM DATA -> B
042D 7A       122           MOV   A,D       FIRST BYTE OF FF LOCATION -> A
042E BC       123           CMP   H         TEST FOR H = D
042F C24104   124           JNZ   SET00     NO, JUUMP
0432 7B       125           MOV   A,E       SECOND BYTE OF FF LOCATION -> A
0433 BD       126           CMP   L         TEST FOR LOCATION READ = FF BYTE LOC.
0434 C24104   127           JNZ   SET00     NO, JUMP
0437 3EFF     128           MVI   A,X'FF'   FF -> A
0439 B8       129           CMP   B         TEST FOR DATA READ = FF
043A C2CC04   130           JNZ   RAMERROR  NO, JUMP
043D 23       131           INX   H         INCREMENT RM ADDRESS
043E C32604   132           JMP   CHECK1    JUMP TO READ NEXT LOCATION
0441 3E00     133 SET00     MVI   A,X'00'   00 -> A
0443 B8       134           CMP   B         TEST FOR DATA READ = 00
0444 C2CC04   135           JNZ   RAMERROR  NO, JUMP
0447 23       136           INX   H         INCREMENT RAM ADDRESS
0448 C32604   137           JMP   CHECK1    JUMP TO READ NEXT LOCATION
044B 3E30     138 OUT2      MVI   A,X'30'   30 -> A
044D B9       139           CMP   C         TEST FOR 4K RAM
044E CA8704   140           JZ    MOVE1     YES, JUMP
0451 62       141           MOV   H,D       FIRST BYTE OF FF BYTE LOCATION -> H
0452 6B       142           MOV   L,E       SECOND BYTE OF FF BYTE LOCATION -> L
0453 3E00     143           MVI   A,X'00'   00 -> A
0455 77       144           MOV   M,A       00 -> FF BYTE LOCATION
0456 3E10     145           MVI   A,X'10'   10 -> A
0458 B2       146           ORA   D         SET FF BYTE LOCATION
0459 57       147           MOV   D,A       D,E = FF BYTE LOCATION
045A 62       148           MOV   H,D       SET RAM ADDRESS = FF BYTE LOCATION
045B 6B       149           MOV   L,E       SET RAM ADDRESS = FF BYTE LOCATION
045C 3EFF     150           MVI   A,X'FF'   DATA = FF
045E 77       151           MOV   M,A       DATA -> RAM
045F 210030   152           LXI   H,X'3000' SET RAM ADDRESS = 3000
0462 3E40     153 CHECK3    MVI   A,X'40'   40 -> A
0464 BC       154           CMP   H         TEST FOR H = 40
0465 CA8704   155           JZ    MOVE1     YES, JUMP
0468 46       156           MOV   B,M       RAM DATA -> B
0469 7A       157           MOV   A,D       FIRST BYTE OF FF LOCATION -> A
046A BC       158           CMP   H         TEST FOR H = D
046B C27D04   159           JNZ   SET00A    NO, JUMP
046E 7B       160           MOV   A,E       SECOND BYTE OF FF LOCATION -> A
046F BD       161           CMP   L         TEST FOR LOCATION READ = FF BYTE LOC.
0470 C27D04   162           JNZ   SET00A    NO, JUMP
0473 3EFF     163           MVI   A,X'FF'   FF -> A
0475 B8       164           CMP   B         TEST FOR DATA READ = FF
0476 C2CC04   165           JNZ   RAMERROR  NO, JUMP
0479 23       166           INX   H         INCREMENT RAM ADDRESS
047A C36204   167           JMP   CHECK3    JUMP TO READ NEXT LOCATION
047D 3E00     168 SET00A    MVI   A,X'00'   00 -> A
047F B8       169           CMP   B         TEST FOR DATA READ = 00
0480 C2CC04   170           JNZ   RAMERROR  NO, JUMP
0483 23       171           INX   H         INCREMENT RAM ADDRESS
0484 C36204   172           JMP   CHECK3    JUMP TO READ NEXT LOCATION
0487 62       173 MOVE1     MOV   H,D       FIRST BYTE OF FF BYTE LOCATION -> H
```

```
0488 6B       174            MOV   L,E       SECOND BYTE OF FF BYTE LOCATION -> L
0489 3E00     175            MVI   A,X'00'   00 -> A
048B 77       176            MOV   M,A       00 -> FF BYTE LOCATION
048C 13       177            INX   D         INCREMENT FF BYTE LOCATION
048D 3E30     178            MVI   A,X'30'   30 -> A
048F B9       179            CMP   C         TEST FOR 4K RAM TEST
0490 C29604   180            JNZ   SET#40    NO, JUMP
0493 C39804   181            JMP   COMPARE1
0496 3E40     182 SET#40     MVI   A,X'40'   40 -> A
0498 BA       183 COMPARE1   CMP   D         TEST FOR D = 30, 40
0499 CAB204   184            JZ    LOOP#     YES, JUMP
049C DB01     185            IN    X'01'     2200 INTERFACE STATUS IN
049E E604     186            ANI   X'04'     TEST FOR RESET
04A0 C21300   187            JNZ   SFKEY     YES, JUMP
04A3 3E30     188            MVI   A,X'30'   30 -> A
04A5 B9       189            CMP   C         TEST FOR 4K RAM
04A6 CAAD04   190            JZ    OUT3CONT  YES, JUMP
04A9 3EEF     191            MVI   A,X'EF'   EF -> A
04AB A2       192            ANA   D         MASK OUT HIGH ORDER 1 BIT
04AC 57       193            MOV   D,A       A -> D
04AD 62       194 OUT3CONT   MOV   H,D       FIRST BYTE OF FF LOCATION -> H
04AE 6B       195            MOV   L,E       SECOND BYTE OF FF LOCATION -> L
04AF C32004   196            JMP   WRITEFF   JUMP TO WRITE FF IN NEXT LOCATION

04B2 310030   198 LOOP#      LXI   SP,X'3000' SET STACK POINTER
04B5 3EFF     199            MVI   A,X'FF'   SET BUSY BYTE
04B7 D307     200            OUT   X'07'     SET CONT. BUSY
04B9 3E00     201            MVI   A,X'00'   SET LOOP CONTROL BYTE
04BB CD0A05   202            CALL  OUTCPU    SEND IBS
04BE CDF704   203            CALL  CBSCHECK  WAIT FOR CBS
04C1 DB02     204            IN    X'02'     CLEAR CBS
04C3 3E30     205            MVI   A,X'30'   30 -> A
04C5 B9       206            CMP   C         TEST FOR 4K RAM TEST
04C6 CA4C03   207            JZ    LRAMTST4  YES, JUMP
04C9 C35403   208            JMP   LRAMTST8  JUMP TO 8K RAM TEST

04CC 310030   210 RAMERROR   LXI   SP,X'3000' SET STACK POINTER
04CF EB       211            XCHG            RAM ADDRESS -> D,E
04D0 67       212            MOV   H,A       DATA SENT -> H
04D1 68       213            MOV   L,B       DATA RECEIVED -> L
04D2 3EFF     214            MVI   A,X'FF'   SET ERROR CONTROL BYTE
04D4 D307     215            OUT   X'07'     SET CONT. BUSY
04D6 CD1D05   216            CALL  ERROROUT  SEND DATA
04D9 7C       217            MOV   A,H       DATA SENT -> A
04DA CD1D05   218            CALL  ERROROUT  SEND DATA
04DD 7D       219            MOV   A,L       DATA RECEIVED -> A
04DE CD1D05   220            CALL  ERROROUT  SEND DATA
04E1 7A       221            MOV   A,D       HIGH ORDER BYTE OF RAM ADDR -> A
04E2 CD1D05   222            CALL  ERROROUT  SEND DATA
04E5 7B       223            MOV   A,E       LOW ORDER BYTE OF RAM ADDR -> A
04E6 CD0A05   224            CALL  OUTCPU    SEND DATA
04E9 CDF704   225            CALL  CBSCHECK  WAIT FOR CBS
04EC DB02     226            IN    X'02'     CLEAR CBS
04EE 3E30     227            MVI   A,X'30'   30 -> A
04F0 B9       228            CMP   C         TEST FOR 4K RAM TEST
04F1 CA4C03   229            JZ    LRAMTST4  YES, JUMP
04F4 C35403   230            JMP   LRAMTST8  JUMP TO 8K RAM TEST
```

```
04F7 CD0205   233 CBSCHECK CALL  RESET     TEST FOR RESET
04FA DB01     234          IN    X'01'     2200 INTERFACE STATUS IN
04FC E602     235          ANI   X'02'     TEST FOR CBS
04FE CAF704   236          JZ    CBSCHECK  NO, JUMP
0501 C9       237          RET

0502 DB01     239 RESET    IN    X'01'     2200 INTERFACE STATUS IN
0504 E604     240          ANI   X'04'     TEST FOR RESET
0506 C21300   241          JNZ   SFKEY     YES, JUMP
0509 C9       242          RET

050A F5       244 OUTCPU   PUSH  PSW       SAVE A, AND STATUS
050B CD0205   245          CALL  RESET
050E DB01     246 CPURDY3  IN    X'01'     2200 INTERFACE STATUS IN
0510 E608     247          ANI   X'08'     TEST FOR CPU READY
0512 CA0E05   248          JZ    CPURDY3   NO, JUMP
0515 F1       249          POP   PSW       RETRIEVE A, AND STATUS
0516 D301     250          OUT   X'01'     DATA OUT
0518 3E00     251          MVI   A,X'00'   READY BYTE
051A D307     252          OUT   X'07'     SET CONT. READY
051C C9       253          RET

051D CD0A05   255 ERROROUT CALL  OUTCPU    SEND IBS
0520 CDF704   256          CALL  CBSCHECK  WAIT FOR CBS
0523 DB02     257          IN    X'02'     CLEAR CBS
0525 3EFF     258          MVI   A,X'FF'   SET BUSY BYTE
0527 D307     259          OUT   X'07'     SET CONT. BUSY
0529 C9       260          RET

              262 *
              263 *
              264 *
```

REFERENCES

ABAGAIN1 (0150) --01B8

ABERROR1 (01BB) --015A

ABERROR2 (01C1) --0162

ABTEST (0149) --003F, 0143, 018A, 01A6

AGAIN (0274) --0329, 0349

BEGIN (0096) --009E

CBSCHECK (04F7) --00E9, 0108, 0114, 013B, 0150, 016C, 0178, 01CC, 022E, 0265,
--02BC, 0305, 0324, 0344, 04BE, 04E9, 04FE, 0520

CBSCHEK (0003) --0007

CBSCHEK1 (0077) --007B

CHECK (0371) --0389, 0393

CHECK1 (0426) --043E, 0448

CHECK2 (03AD) --03C5, 03CF

CHECK3 (0462) --047A, 0484

CHKCOUNT (00DC) --00D5

COMPARE (03E3) --03DE

COMPARE1 (0498) --0493

COMPARE2 (020E) --0208

CONT (018D) --0180

CONT1 (0178) --01D9

CONT2 (01A9) --019C

COUNTCHK (00B4) --00AD

CPURDY1 (0080) --0084

CPURDY2 (00EE) --00F2

CPURDY3 (050E) --0512

NUMBER OF ERRORS IN ASSEMBLY 0

MODULE REPAIR GUIDE NO. 3.1
EDITED BY CUSTOMER ENGINEERING DIVISION

March 10, 1978

# 2236 SYSTEM REPAIR ADDENDUM

## 1. INTRODUCTION

This addendum contains information concerning the use of the Debug Tester for observation of the debug microprograms available for testing the 2236C(D) Terminal and 2236 MXC(D) Controller. The tester will not operate as described in MRG #3 when certain options (IN, OUT, WR) are selected. In the future, additional circuitry will be added to the 210-7292(-1) and all the options, as listed in MRG No. 3, will be available for use. The changes, however, will NOT be added to the 210-7291. The diagnostic PROM programs for both 7291 and 7292 will operate without the changes, but STEPPING through the microprogram will not produce the same results as described in MRG #3.

Section 2 of this addendum deals with the problems of using the tester with the 7291 PCB and Section 3 concerns the 7292(-1) PCB.

## 2. 210-7291

The IN position of the rotary switch is the only option that is not available for use with the 7291 PCB. Following is a list of jumper wires that may be incorporated to effect the necessary additions to the circuitry.

NOTE:

The additional circuitry as described below should ONLY be added if the IN option is an absolute necessity for accomplishing board repair. The changes should be removed after the PCB is repaired.

Printed in U.S.A.

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MASSACHUSETTS 01851, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

| JUMPER | FROM | TO |
|---|---|---|
| | L4, Pin 8 | J1, Pin N |
| | L4, Pin 11 | L3, Pin 8 |
| | L4, Pin 12 | L18, Pin 5 |
| | L4, Pin 13 | L9, Pin 1 |

3. 210-7292

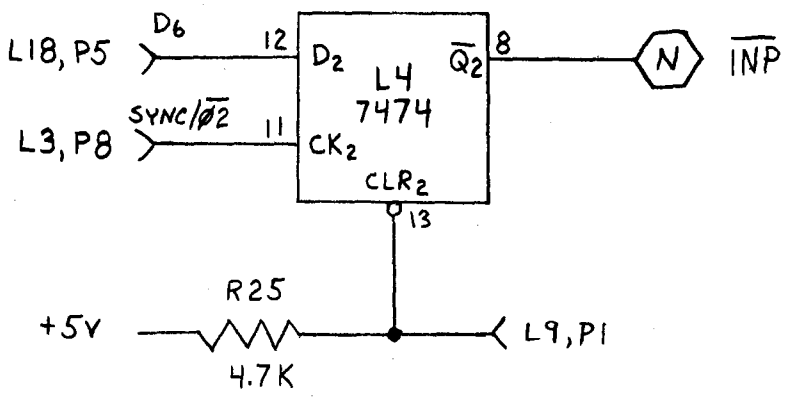
The only instruction types that may be used when operating the tester with the 7292(-1) PCB are M1 and MEMR. To aid in the repair of the 7292(-1), the following additions should be made (Reference ECN 8232):

NOTE:

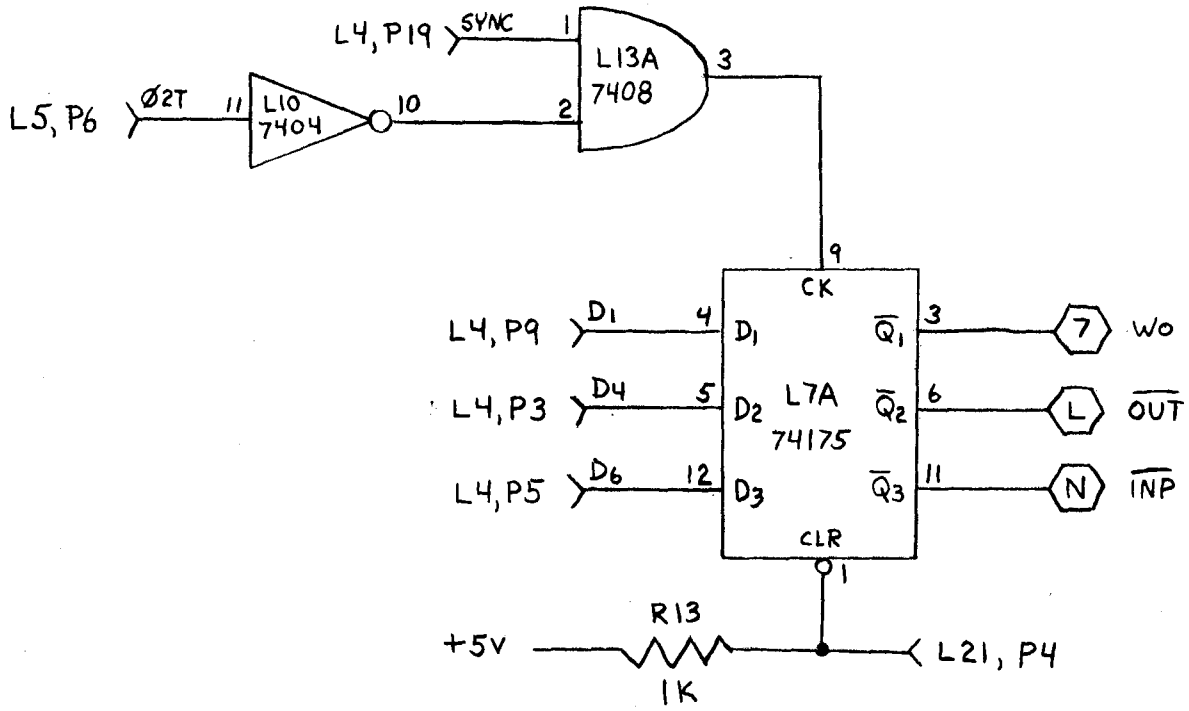
The additional circuitry as described below and in ECN 8232 should be added to provide the IN, OUT and WR options for accomplishing board repair. The changes should be left in after the PCB is repaired and the E-Rev should be incremented from 1 to 2. All future 7292 boards (after March 31, 1978) will have the ECN incorporated into the artwork.

1) Add (1) 74175 in the location to the left of L7 and designate this as L7A. Connect pin 8 to ground and pin 16 to +5V.
2) Add (1) 7408 in the location to the left of L13 and designate this as L13A. Connect pin 7 to ground and pin 14 to +5V. (This IC has already been added to the 7292-1.)

| JUMPER | FROM | TO |
|---|---|---|
| | L5, Pin 6 | L10, Pin 11 |
| | L4, Pin 19 | L13A, Pin 1 |
| | L10, Pin 10 | L13A, Pin 2 |
| | L13A, Pin 3 | L7A, Pin 9 |
| | L7A, Pin 1 | L21, Pin 4 |
| | L7A, Pin 3 | Conn. 4, Pin 7 |
| | L7A, Pin 6 | Conn. 4, Pin L |
| | L7A, Pin 11 | Conn. 4, Pin N |
| | L4, Pin 9 | L7A, Pin 4 |
| | L4, Pin 3 | L7A, Pin 5 |
| | L4, Pin 5 | L7A, Pin 12 |




210-7291 ECN




210-7292(-1) ECN