LPA11 LPA/DMC-11 TEST II AH-B056B-MC
CRLPMB0 FICHE 1 OF 1
FEB 1981

## IDENTIFICATION

| | |
|---|---|
| PRODUCT CODE: | AC-B055B-MC |
| DIAGNOSTIC CODE: | MAINDEC-11-CRLPM-B-D |
| PRODUCT NAME: | CRLPMB0 LPA/DMC-11 TEST II |
| DATE: | DEC. 1980 |
| MAINTAINER: | DIAGNOSTIC ENG. |

1. ABSTRACT

This diagnostic is one of a series of diagnostics aimed at the the LPA-11X system. Please reference section 8.7 for a complete list.

The function of the M8200-YC diagnostics is to verify that the option operates according to specifications. The diagnostics verfiy that there are no malfunctions and the all operations of the M8200-YC are correct in its environment.

This diagnostic requires the user to recable the system, that is, the LPA-11X I/O bus must join the unibus.

Parameters must be set up to alert the diagnostics to the M8200-YC configuration. These parameters are contained in the STATUS TABLE and are generated in two ways: 1) Manual Input - the operator answers questions. 2) Autosizing - the program determines the parameters automatically.

It performs jump tests on the micro-processor and verifies the
M of the M8200-YC. BO this diagnostic will not run on
a KMC (M8204), however it is possible to load the KMC CRAM with the M8200-YC micro-code. See test 2 for details.

Currently there are two off line diagnostics that are to be run in sequence to insure that if an error should occur it will be detected at an early stage.

NOTE: Additional diagnostics may be added in the future.

The two diagnostics are:

1. CRLPL [REV] LPA/DMC-11 DIAGNOSTIC TST I
   (BASIC W/R AND MICRO-PROCESSOR TESTS)
2. CRLPM [REV] LPA/DMC-11 DIAGNOSTIC TST II
   (JUMP AND CROM TESTS)

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8k memory
CONSOLE I/O TERMINAL

2.2 STORAGE

Program will use all 3K of memory except where ABL and BOOTSTRAP LOADER reside. Locations 1500 thru 1640; contain the "STATUS TABLE" information which is generated at start of diagnostics by manual input (questions) or automatically (auto-sizing). This area is an overlay area and should not be altered by the operator.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK ,MAGTAPE,DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address *500

| MEMORY | * SIZE |
|---|---|
| 4k | 17 |
| 8k | 37 |
| 12k | 57 |
| 16k | 77 |
| 20k | 117 |
| 24k | 137 |
| 28k | 157 |

3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)

3.1.2 Depress 'LOAD ADDRESS' key on console and release.

3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

a. Set switch register to 000200
b. Depress 'LOAD ADDRESS' key and release
c. Set SWR to zero for 'AUTO SIZING' or SWR bit0=1 for manual input (questions) or SWR bit7=1 to use existing parameters set up by a previous start or a previously run M8200-YC diagnostic.
d. Depress 'START KEY' and release. The program will type Maindec Name and program name (if this was the first start up of the program) and also the following:

MAP OF M8200-YC status

| PC | CSR | STAT1 | STAT2 | STAT3 |
|---|---|---|---|---|
| 001500 | 160010 | 145310 | 177777 | 000000 |

The program will type 'R' and proceed to run the diagnostic. The above is only an example. This would indicate the status table starting at add. 1500 in the program. In this example the table contains the information and status of an M8200-YC. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

If the diagnostic was started with SW00=1 indicating manual parameter input then the following shows an example of the questions asked and some example answers:

```
HOW MANY M8200-YC'S TO BE TESTED?1

01
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
```

FOLLOWING THE QUESTIONS THE STATUS MAP IS PRINTED OUT AS DESCRIBED ABOVE, THE INFORMATION IN THE MAP REFLECTS THE ANSWERS TO THE QUESTIONS. IF THE DIAGNOSTIC WAS STARTED WITH SW00=0 and SW07=0 (AUTO-SIZING) then no questions are asked and only the status-map is printed out. If AUTO-SIZING is used the status information must be verified to be correct (match the hardware). if it does not match the hardware the diagnostic must be restarted with SW00=1 and the questions answered.

4.1 CONTROL SWITCH SETTINGS

SW 15 Set: Halt on error
SW 14 Set: Loop on current test
SW 13 Set: Inhibit error print out
SW 12 Set: Inhibit type out/bell on error.
SW 11 Set: Inhibit iterations. (quick pass)
SW 10 Set: Escape to next test on error
SW 09 Set: Loop with current data
SW 08 Set: Catch error and loop on it
SW 07 Set: Use previous status table.
SW 06 Set: Halt in ROMCLK routine before clocking micro-processor
SW 05 Set: Reserved
SW 04 Clear: Select V5 of M8200-YC's micro-code
SW 04 Set: Select V4 of M8200-YC's micro-code
SW 03 Set: Reselect M8200-YC's desired active
SW 02 Set: Lock on selected test
SW 01 Set: Restart program at selected test
SW 00 Set: Build new status table from questions. (If SW07=0 and SW00=0 a new status table is built by auto-sizing)

Switch 06 and 08-15 are dynamic and can be changed as needed while the diagnostic is running. Switches 00-03 and switch 07 are static, and are used only on starting or restarting the diagnostic.

4.1.2 SWITCH REGISTER OPTIONS (at start up)

SW 01 RESTART PROGRAM AT SELECTED TEST. It is strongly suggested that at least one pass has been made before trying to select a test, the reason being is that the program has to clear areas and set up parameters. When this switch is used the diagnostic will ask TEST NO.? Answer by typing the number of the test desired and carrige return to begin execution at the selected test.

SW 02 LOCK ON SELECTED TEST. This switch when used with SW01 will cause the program to constantly loop on the selected test. Hitting any key on the console will let it advance to the next test and loop until a key is hit again. If SW02=0 when SW01 is used. The program will begin at the selected test and continue normal operations.

SW 03 RESELECT M8200-YC's desired active. Please note that a message is typed out for setting the switch register equal to M8200-YC's active. this means if the system has four M8200-YC's; bits 00,01,02,03 will be set in loc 'DMACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four M8200-YC's are in the system ***DO NOT*** set switchs greater than SW 03 in the up position. This would be a fatal error. Do not select more active M8200-YC's than there is information on in the status table.

METHOD: A: Load address 200
B: Start with SW 00=1
C: Program will type message
D: Set a switch for each M8200-YC desired active.
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired. PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Goto beginning of the test(on error).
5. SW 10 Goto next test(on error).

SCOPE SWITCHES

1. SW06 Halt in ROMCLK routine before clocking micro-processor instruction. This allows the operator to scope a micro-processor instruction in the static state before it is clocked. Hit continue to resume running.
2. SW09 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 10 ) SW09 is incorporated in that test and therefore SW09 is usually the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a HARD error (constant); SW08 is best. (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the M8200-YC diagnostics. (See Section 4.0)

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after all available M8200-YC's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section 4.0 will be printed, and program will begin running the diagnostic

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

1. Halt on error (via SW 15=1) when ever an error occurs.
2. Clear SW 15.
3. Set SW 14: (loop on this test)
4. Set SW 13: (inhibit error print out)

The TEST NUMBER and PC will be typed out and possibily an error message (this depends on the test) to give the operator an idea as to the source of the problem. If it is necessary to know more information concerning the error report; LOOK IN THE LISTING for that TEST NUMBER which was typed out and then NOTE THE PC of thE ERROR REPORT this way the EXACT FUNCTION of the test CAN BE DETERMINED.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). in most cases additional information will be supplied in the the error message to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the M8200-YC should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should happen; look in location 'TSTNO' (address 1226)for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the M8200-YC was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified reguardless of how program was started. Also it is important to use this listing along with the information printed on the CONSOLE TERMINAL to completly isolate problems.

7.2 OPERATING RESTRICTIONS

The first time a M8200-YC diagnostic is loaded into core and run the STATUS TABLE must be set up. This is done by manual input (SW00=1) or by autosizing (SW00=0 and SW07=0). Thereafter however the status table need not be setup by subsequent restarts or even loading the next M8200-YC diagnostic because the STATUS TABLE is overlayed. The current parameters in the STATUS TABLE are used when SW07=1 on start up.

M8200-YC must be on the unibus.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

M8200-YC - Jumper W1 must be IN, and switch 7 of E76 must be in the OFF position.

KMC(M8204)- Jumper W1 must be IN.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All M8200-YC device diagnostics will give an 'END PASS' message (providing no errors and SW12=0) within 4 mins. This is assuming SW11=1 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration and the amount of memory in the system.

8.2 PASS COMPLETE

NOTE: EVERY time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO HARD ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all M8200-YC's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

END PASS CRLPMB CSR: 175000 VEC: 0300 PASSES: 000001
ERRORS: 000000

NOTE: The pass count and error counts are cummulitive for each M8200-YC that is running, and are set to zero only when the diagnostic is started. Therefore after an overnight run for example, the total passes and errors for each M8200-YC since the diagnostic was started are reflected in PASSES: and ERRORS:.

8.4 KEY LOCATIONS

RETURN (1214) Contains the address where program will return when iteration count is reached or if loop on test is asserted.

NEXT (1216) Contains the address of the next test to be peformed.

TSTNO (1226) Contains the number of the test now being peformed.

RUN (1316) The bit in 'RUN' always points to the M8200-YC currently being tested. EXAMPLE: (RUN) 1316/0000000001000000 Means that M8200-YC NO.06 is the M8200-YC now running.

DMCR00-DMCR17
DMST00-DMST17
(1500)-(1700)

These locations contain the information needed to test up to 16 (decimal) M8200-YC's sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each M8200-YC.

DMACTV (1306) Each bit set in this location indicates that the associated M8200-YC will be tested in turn. EXAMPLE: (DMACTV) 1306/0000000000011111 means that M8200-YC no. 00,01,02,03,04 will be tested. EXAMPLE: (DMACTV) 1306/0000000000010001 Means that M8200-YC no. 00,04 will be tested.

DMCSR (1404) Contains the CSR of the current M8200-YC under test.

8.4A 'STATUS TABLE' (1500-1700)

The table is filled by AUTO SIZING or by the manual parameter input (questions) as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

The example status map shown below contains information for two M8200-YC'S. the table can contain up to 16 M8200-YC'S. Following the map is a description of the bits for each map entry

MAP OF M8200-YC STATUS

| PC | CSR | STAT1 | STAT2 | STAT3 |
|---|---|---|---|---|
| 001500 | 160010 | 145310 | 177777 | 000000 |
| 001510 | 160020 | 016320 | 000000 | 000000 |

Each map entry contains 4 words which contain the status information for 1 M8200-YC. The PC shows where in core memory the first of the 4 words is. In the example above the first M8200-YC's status is in locations, 1500, 1502, 1504, and 1506. The second M8200-YC status is located at 1510, 1512, 1514, and 1516. The information contained in each 4 word entry is defined as follows:

CSR: Contains M8200-YC CSR address

STAT1: BITS 00-08 IS M8200-YC VECTOR ADDRESS
BIT15=1 MICRO-PROCESSOR HAS CRAM
BIT15=0 MICRO-PROCESSOR HAS CROM
BIT14=1 TURNAROUND CONNECTOR IS ON
BIT14=0 NO TURNAROUND CONNECTOR
BITS 09-11 IS M8200-YC BR PRIORITY LEVEL

STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3: BIT0=1 PERFORM FREE RUNNING TESTS ON KMC

## 8.5 METHOD OF AUTO SIZING

### 8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a M8200-YC as follows: It starts at address 170440 and tests all address in increments of 10 up to and including address 170500. If the address does not time out, the following is done, the first CROM address is written to a 125252 then it is read back. If it contains a -1 or 125252 or 456 or 16520 a M8200-YC or KMC11 has been found, if not, the address is updated by 10 and the search continues. a 125252 indicates a KMC11 with CRAM. a 456 indicates a M8200-YC. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. All M8200-YC's in the system will be found by the auto-sizer. If it does not find a M8200-YC the diagnostic must be restarted and the questions answered.

### 8.5.2 FINDING THE VECTOR AND BR LEVEL

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). The processor status is started at 7 and the DMC is programmed to interrupt. The PS is lowered by 1 until the DMC interrupts, a delay is made and if no interupt occures at PS level 3 (because of a bad M8200-YC) the program assumes vector address 300 at BR level 5 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interupt occured; the address to which the M8200-YC interupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

## 8.6 SOFTWARE SWITCH REGISTER

If the diagnostic is run on an 11/04 or other CPU without a switch register then a software switch register is used to allow user the same switch options as described previously. If the hardware switch register does not exist or if one does and it contains all ones (177777) this software switch register is used.

Control:

To obtain control at any allowable time during execution of the diagnostic the operator types a CTRL G on the console terminal keyboard. As soon as the CTRL G is recognized, by the diagnostic, the following message will be displayed:

SWR=XXXXXX NEW?

Where XXXXXX is the current contents of the software switch register in octal. The software control routine will then await operator action. At which time the operator is required to type one or more of the legal characters: 1) 0 - 7, 2) line feed(<LF>), 3) carriage return(<CR>), or 4) control-U (CTRL U). No check is made for legality. If the input character is not a <LF>, <CR>, or CTRL U it is assumed to be an octal digit.

To change the contents of the SSR the operator simply types the new desired value in octal - leading zeros need not be typed. And terminates the input string with a <CR> or <LF> depending on the program action desired as described below. The input value will be truncated to the last 6 digits typed. At least one digit must be typed on any given input string prior to the terminator before a change to the SSR will occur.

When the input string is terminated with a <CR> the diagnostic will continue execution from the point at which it was interrupted. If a <CR> is the only thing typed the program will continue without changing the SSR. The <LF> differs from the <CR> by restarting the program as if it were restarted at address 200.

If a CTRL U is typed at any point in the input string prior to the terminator the input value will be disregarded and the prompt displayed (SWR = XXXXXX NEW?).

To set the SSR for the starting switches, first load the diagnostic, then hit CTRL G, then start the diagnostic.

## 8.7 LPA-11 (SYSTEM) DIAGNOSTIC SUMMARY

Diagnostics for the LPA-11 are written at three levels: (1) total PDP-11 system, (2) LPA-11 system; and, (3) LPA-11 options.

Level 1, is designed to isolate a failure to the LPA-11 system. All options on the PDP-11 are exercised.

Level 2 diagnostics isolate a failure to the individual option within the LPA-11. The level 2 diagnostic is "CRLPA". When the user runs CRLPA he can generally tell which option diagnostic (level 3) to run next. M8254 and M8200-YC errors may "look" alike and "CRLPA" may not be able to distinguish between them. Arbitration errors will not be detected by this diagnostic.

Level three diagnostics aid in determining if the error was in fact on the option the "CRLPA specified. The user may "loop" on the error. Within level three, there are two groups of diagnostics. The first group requires no "extra" work by the user in order to run. Group "A" diagnostics do not check arbitration, and require extra time for execution. The second group (group "B") requires that the user reconfigure the PDP-11 system. This reconfiguration involves cabling the unibus to the LPA's I/O bus.

The diagnostic for the M8254 falls into the group "B" catagory.

THE LPA-11XX DIAGNOSTIC KIT WILL INCLUDE:

| OPTION | GROUP | DIAG. # | DIAG. TITLE |
|---|---|---|---|
| LPA-11XX | LEVEL 2 | MD-11-CRLPA | LPA-11 SYSTEM EXER. |
| M8254 | "B" | MD-11-CRLPN | M8254 (IPBM) FIELD DIAG. |
| AA11K | A | MD-11-CRLPB | LPA/AA11K DIAG. |
| | B | MD-11-DZAAC | AA11-K DIAG. |
| AR11K | A | MD-11-CRLPC | LPA/AR11 DIAG. #1 |
| | A | MD-11-CRLPD | LPA/AR11 DIAG. #2 |
| | A | MD-11-CRLPE | LPA/AR11 DIAG. #3 |
| | B | MD-11-DZARA | AR11 DIAG. #1 |
| | B | MD-11-DZARB | AR11 DIAG. #2 |
| | B | MD-11-DZARC | AR11 DIAG. #3 |

| | | | |
|---|---|---|---|
| DR11K | A | MD-11-CRLPF | LPA/DR11K DIAG. |
| | B | MD-11-DZDRG | DR11K DIAG. |
| KW11K | A | MD-11-CRLPG | LPA/KW11K DIAG. |
| | B | MD-11-DZKWK | KW11K DIAG. |
| LPS-11 | A | MD-11-CRLPH | LPA/LPS-11 DIAG. #1 |
| | A | MD-11-CRLPI | LPA/LPS-11 DIAG. #2 |
| | A | MD-11-CRLPJ | LPA/LPS-11 DIAG. #3 |
| | B | MD-11-DZLPC | LPS-11 DIAG. #1 |
| | B | MD-11-DZLPD | LPS-11 DIAG. #2 |
| | B | MD-11-DZLPI | LPS-11 DIAG. #3 |
| AD11K | A | MD-11-CRLPK | LPA/AD11K DIAG. |
| | B | MD-11-DZADL | AD11K DIAG. |
| M8200-YC | B | MD-11-CRLPL | LPA/DMC-11 DIAG. TEST I |
| | B | MD-11-CRLPM | LPA/DMC-11 DIAG. TEST II |

HISTORY FILE OF CRLPM



```
*****************************************
VERSION "B" WAS CREATED BECAUSE OF A CHANGE TO THE LPA
MICRO CODE. THIS PROGRAM DID NOT REALLY CHANGE MUCH.
THE MAIN CHANGE WAS INSTALLING THE NEW VERSION OF
M8200-YC ROM CODE IN THE .P11 FILE AND USING SWR BIT 4 TO
INDICATE V4 OR V5 MICROCODE BEING VERIFIED.
THE .RND FILE WAS CLEANED UP FOR UPPER AND LOWER CASE.

NOTE:    THE FILES CRLPLB.MAC AND CRLPMB.MAC ARE THE SAME.
         THE FILES CRLPLB.RND AND CRLPMB.RND ARE ALMOST THE SAME.
         THE .HST FILE WAS CREATED
         THE .CTL FILE WAS CREATED
         THE .OPR FILE WAS CORRECTED

                  R. SHOOP
*****************************************
```

```
                        ;*MAINDEC-11-CRLPM-B   LPA-DMC-11 DIAGNOSTIC TST II
                        ;*COPYRIGHT 1980, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*-------------------------------------------------------------------

                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;SWR=0  AUTOSIZE M8200-YC
                        ;SW07=1  USE CURRENT M8200-YC PARAMETERS
                        ;SW00=1  INPUT NEW M8200-YC PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-CRLPM-B   LPA-DMC-11 DIAGNOSTIC TST II"
                        ;PROGRAM WILL TYPE STATUS MAP
                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                        ;AND THEN RESUME TESTING
                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE



                        ;SWITCH REGISTER OPTIONS
                        ;----------------------

  100000                SW15=100000             ;=1,HALT ON ERROR
  040000                SW14=40000              ;=1,LOOP ON CURRENT TEST
  020000                SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
  010000                SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
  004000                SW11=4000               ;=1,INHIBIT ITERATIONS
  002000                SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
  001000                SW09=1000               ;=1,LOOP WITH CURRENT DATA
  000400                SW08=400                ;=1,LOOP ON ERROR
  000200                SW07=200                ;=1,USE CURRENT M8200-YC PARAMETERS, =0,AUTOSIZE M8200-YC
  000100                SW06=100                ;=1, HALT BEFORE CLOCKING MICRO-PROCESSOR INSTRUCTION
  000040                SW05=40
  000020                SW04=20                 ;=1, USE V4 IN PLACE OF V5 MICRO-CODE ON M8200-YC
  000010                SW03=10                 ;RESELECT M8200-YC'S TO BE TESTED (ACTIVE)
  000004                SW02=4                  ;LOCK ON TEST SELECT
  000002                SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
  000001                SW00=1                  ;INPUT M8200-YC PARAMETERS
```

```
                                ;REGISTER DEFINITIONS
                                ;--------------------

000000                          R0=%0           ;GENERAL REGISTER
000001                          R1=%1           ;GENERAL REGISTER
000002                          R2=%2           ;GENERAL REGISTER
000003                          R3=%3           ;GENERAL REGISTER
000004                          R4=%4           ;GENERAL REGISTER
000005                          R5=%5           ;GENERAL REGISTER
000006                          SP=%6           ;PROCESSOR STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER

                                ;LOCATION EQUIVALENCIES
                                ;----------------------

177776                          PS=177776       ;PROCESSOR STATUS WORD
001200                          STACK=1200      ;START OF PROCESSOR STACK

                                ;INSTRUCTION DEFINITIONS
                                ;-----------------------

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                          PUSHR0=10046    ;SAVE R0 ON STACK
012600                          POPR0=12600     ;RESTORE R0 FROM STACK
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                          POP2SP=22626    ;INCREMENT STACK TWICE
                                .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL

                                ;BIT DEFINITIONS
                                ;---------------

100000                          BIT15=100000
040000                          BIT14=40000
020000                          BIT13=20000
010000                          BIT12=10000
004000                          BIT11=4000
002000                          BIT10=2000
001000                          BIT9=1000
000400                          BIT8=400
000200                          BIT7=200
000100                          BIT6=100
000040                          BIT5=40
000020                          BIT4=20
000010                          BIT3=10
000004                          BIT2=4
000002                          BIT1=2
000001                          BIT0=1
```

```
                                ;:**********************************************************************
                                ;----------------------------------------------------------------------
                                        ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                                        ;THE STANDARD "TRAP CATCHER" IS PLACED
                                        ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                        ;IT LOOKS LIKE "PC+2 HALT".
                                ;----------------------------------------------------------------------
                                ;:**********************************************************************

000000                          .=0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

000024                          .=24
000024  005346                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  004760                          .HLT                    ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  004726                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
000040                          .=40
000040  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000042  000000                          0                       ;RETURN ADDRESS IF UNDER ACT-11 OR XXDP
000044  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000046  003532                          $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
000052                          .=52
000052  000000                          0                       ;ACT-11 PROGRAM CHARACTERISTICS

000174                          .=174
000174  000000                 DISPREG:0                       ;SOFTWARE DISPLAY REGISTER
000176  000000                 SWREG:  0                       ;SOFTWARE SWITCH REGISTER

000200                          .=200
000200  000137  002002                  JMP     .START          ;GO TO START OF PROGRAM


001000                          .=1000
001000  005377  040515  047111 MTITLE: .ASCII  <377><12>/MAINDEC-11-CRLPM-B/<377>
(2) 001025  000114  040520  042055          .ASCIZ  /LPA-DMC-11 DIAGNOSTIC TST II/<377>
(2)
001200                          .=1200

                                ;INDIRECT POINTERS TO SWITCH REGISTER AND LIGHT DISPLAY
                                ;-----------------------------------------------------

001200  177570                 DISPLAY:177570
001202  177570                 SWR:    177570
```

```
                                ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                ;--------------------------------------------------

001204  177560                  TKCSR:  177560          ;TELETYPE KEYBOARD CONTROL REGISTER
001206  177562                  TKDBR:  177562          ;TELETYPE KEYBOARD DATA BUFFER
001210  177564                  TPCSR:  177564          ;TELEPRINTER CONTROL REGISTER
001212  177566                  TPDBR:  177566          ;TELEPRINTER DATA BUFFER

                                ;PROGRAM CONTROL PARAMETERS
                                ;--------------------------

001214  000000                  RETURN: 0               ;SCOPE ADDRESS FOR LOOP ON TEST
001216  000000                  NEXT:   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
001220  000000                  LOCK:   0               ;ADDRESS FOR LOCK ON CURRENT DATA
001222  000003                  ICOUNT: 3               ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
001224  000000                  LPCNT:  0               ;NUMBER OF ITEREATIONS COMPLETED
001226  000000                  TSTNO:  0               ;NUMBER OF TEST IN PROGRESS
001230  000000                  PASCNT: 0               ;NUMBER OF PASSES COMPLETED
001232  000000                  ERRCNT: 0               ;TOTAL NUMBER OF ERRORS
001234  000000                  LSTERR: 0               ;PC OF LAST ERROR CALL

                                ;PROGRAM VARIABLES
                                ;-----------------

001236  000000                  STRTSW: 0               ;SWITCHES AT START OF PROGRAM
001240  000000                  STAT:   0               ;DM STATUS WORD STORAGE
001242  000000                  CLKX:   0
001244  000000                  MASKX:  0
001246  000000                  TEMP1:  0               ;TEMPORARY STORAGE
001250  000000                  TEMP2:  0               ;TEMPORARY STORAGE
001252  000000                  TEMP3:  0               ;TEMPORARY STORAGE
001254  000000                  TEMP4:  0               ;TEMPORARY STORAGE
001256  000000                  TEMP5:  0               ;TEMPORARY STORAGE
001260  000000                  SAVR0:  0               ;R0 STORAGE
001262  000000                  SAVR1:  0               ;R1 STORAGE
001264  000000                  SAVR2:  0               ;R2 STORAGE
001266  000000                  SAVR3:  0               ;R3 STORAGE
001270  000000                  SAVR4:  0               ;R4 STORAGE
001272  000000                  SAVR5:  0               ;R5 STORAGE
001274  000000                  SAVSP:  0               ;STACK POINTER STORAGE
001276  000000                  SAVPC:  0               ;PROGRAM COUNTER STORAGE
001300  000000                  ZERO:   0
001302  000001                  ONE:    1
001304  000000                  MEMLIM: 0               ;HIGHEST LOCATION FOR NPR'S
001306  000001                  DMACTV: .BLKW 1         ;M8200-YC'S SELECTED ACTIVE.
001310  000001                  DMNUM:  .BLKW 1         ;OCTAL NUMBER OF M8200-YC'S.
001312  000001                  SAVACT: .BLKW 1         ;ORIGINAL ACTV  DEVICES
001314  000001                  SAVNUM: .BLKW 1         ;WORKABLE NUMBER
001316  000000                  RUN:    0               ;POINTER TO RUNNING DEVICE.
                                .EVEN
001320  001472                  CREAM:  DM.MAP-6        ;TABLE POINTER.
001322  001676                  MILK:   CNT.MAP-4       ;TABLE POINTER
```

```
                                ;PROGRAM CONTROL FLAGS
                                ;---------------------

001324  000                     INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001325  000                     ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
001326  000                     LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
001327  000                     QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG.
                                                                ;ON FIRST PASS OF EACH M8200-YC ITERATIONS WILL BE SUPPR

                                .EVEN

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                                        ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

                                ;:*****************************************************************************
                                ;-----------------------------------------------------------------------------
001330                          .TRPTAB:
        104400                  SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001330  003606                          .SCOPE
        104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001332  003746                          .SCOP1
        104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001334  003776                          .TYPE
        104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001336  004060                          .INSTR
        104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001340  004164                          .INSTER
        104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001342  004204                          .PARAM
        104406                  SAV05=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
001344  004404                          .SAV05
        104407                  RES05=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001346  004444                          .RES05
        104410                  CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001350  004476                          .CONVRT
        104411                  CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001352  004502                          .CNVRT
        104412                  MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001354  005476                          .MSTCLR
        104413                  DELAY=TRAP+13           ;CALL TO DELAY
001356  005446                          .DELAY
        104414                  ROMCLK=TRAP+14          ;CALL TO CLOCK ROM ONCE
001360  005514                          .ROMCLK
        104415                  DATACLK=TRAP+15         ;CALL TO CLK DATA
001362  005562                          .DATACLK
        104416                  TIMER=TRAP+16           ;CALL TO DELAY A CLOCK TICK
001364  005626                          .TIMER

                                ;-----------------------------------------------------------------------------
                                ;:*****************************************************************************
```

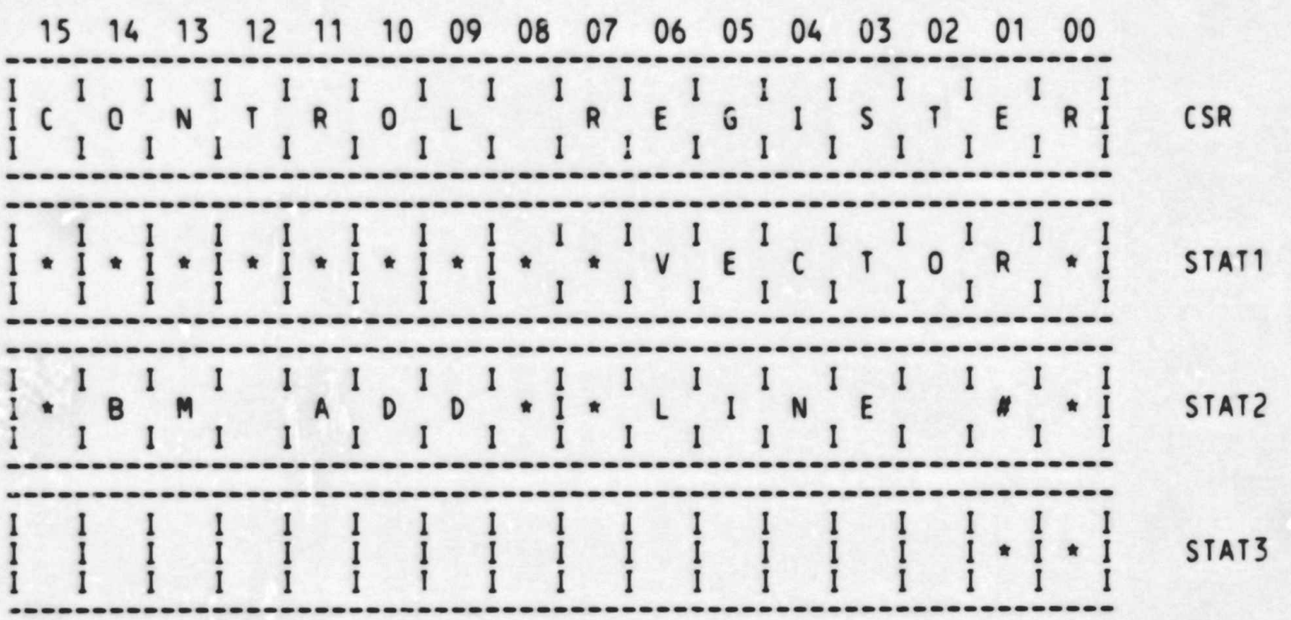
```
                                ;M8200-YC CONTROL INDICATORS FOR CURRENT M8200-YC UNDER TEST
                                ;-----------------------------------------------------
001366  000000                  STAT1:  0
001370  000000                  STAT2:  0
001372  000000                  STAT3:  0

                                ;M8200-YC VECTOR AND REGISTER INDIRECT POINTERS
                                ;---------------------------------------------

001374  000000                  DMRVEC: 0               ;POINTER TO M8200-YC RECEIVER INTERRUPT VECTOR
001376  000000                  DMRLVL: 0               ;POINTER TO M8200-YC RECEIVER INTERRUPT SERVICE PS
001400  000000                  DMTVEC: 0               ;POINTER TO M8200-YC TRANSMITTER INTERRUPT VECTOR
001402  000000                  DMTLVL: 0               ;POINTER TO M8200-YC TRANSMITTER INTERRUPT SERVICE PS
001404  000000                  DMCSR:  0               ;POINTER TO M8200-YC CONTROL STATUS REGISTER
001406  000000                  DMCSRH: 0               ;POINTER TO M8200-YC CONTROL STATUS REGISTER HIGH BYTE.
001410  000000                  DMCTL:  0               ;POINTER TO M8200-YC CONTOL OUT REGISTER
001412  000000                  DMP04:  0               ;POINTER TO M8200-YC PORT REGISTER(SEL 4)
001414  000000                  DMP06:  0               ;POINTER TO M8200-YC PORT REGISTER(SEL 6)

                                ;TEMP STORAGE
                                ;------------

001416  000000                  TEMP:   0
        001460                  .=.+40

                                ;M8200-YC STATUS TABLE AND ADDRESS ASSIGNMENTS
                                ;--------------------------------------------

        001500                  .=1500
001500                          DM.MAP:
001500  000001                  DMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 00
001502  000001                  DMS100: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 00
001504  000001                  DMS200: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 00
001506  000001                  DMS300: .BLKW   1               ;3RD STATUS WORD

001510  000001                  DMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 01
001512  000001                  DMS101: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 01
001514  000001                  DMS201: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 01
001516  000001                  DMS301: .BLKW   1               ;3RD STATUS WORD

001520  000001                  DMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 02
001522  000001                  DMS102: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 02
001524  000001                  DMS202: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 02
001526  000001                  DMS302: .BLKW   1               ;3RD STATUS WORD

001530  000001                  DMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 03
001532  000001                  DMS103: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 03
001534  000001                  DMS203: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 03
001536  000001                  DMS303: .BLKW   1               ;3RD STATUS WORD

001540  000001                  DMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 04
001542  000001                  DMS104: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 04
001544  000001                  DMS204: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 04
001546  000001                  DMS304: .BLKW   1               ;3RD STATUS WORD
```

```
    304  001550  000001                DMCR05: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 05
    305  001552  000001                DMS105: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 05
    306  001554  000001                DMS205: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 05
    307  001556  000001                DMS305: .BLKW   1               ;3RD STATUS WORD
    308
    309  001560  000001                DMCR06: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 06
    310  001562  000001                DMS106: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 06
    311  001564  000001                DMS206: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 06
    312  001566  000001                DMS306: .BLKW   1               ;3RD STATUS WORD
    313
    314  001570  000001                DMCR07: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 07
    315  001572  000001                DMS107: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 07
    316  001574  000001                DMS207: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 07
    317  001576  000001                DMS307: .BLKW   1               ;3RD STATUS WORD
    318
    319  001600  000001                DMCR10: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 10
    320  001602  000001                DMS110: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 10
    321  001604  000001                DMS210: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 10
    322  001606  000001                DMS310: .BLKW   1               ;3RD STATUS WORD
    323
    324  001610  000001                DMCR11: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 11
    325  001612  000001                DMS111: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 11
    326  001614  000001                DMS211: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 11
    327  001616  000001                DMS311: .BLKW   1               ;3RD STATUS WORD
    328
    329  001620  000001                DMCR12: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 12
    330  001622  000001                DMS112: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 12
    331  001624  000001                DMS212: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 12
    332  001626  000001                DMS312: .BLKW   1               ;3RD STATUS WORD
    333
    334  001630  000001                DMCR13: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 13
    335  001632  000001                DMS113: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 13
    336  001634  000001                DMS213: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 13
    337  001636  000001                DMS313: .BLKW   1               ;3RD STATUS WORD
    338
    339  001640  000001                DMCR14: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 14
    340  001642  000001                DMS114: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 14
    341  001644  000001                DMS214: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 14
    342  001646  000001                DMS314: .BLKW   1               ;3RD STATUS WORD
    343
    344  001650  000001                DMCR15: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 15
    345  001652  000001                DMS115: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 15
    346  001654  000001                DMS215: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 15
    347  001656  000001                DMS315: .BLKW   1               ;3RD STATUS WORD
    348
    349  001660  000001                DMCR16: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 16
    350  001662  000001                DMS116: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 16
    351  001664  000001                DMS216: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 16
    352  001666  000001                DMS316: .BLKW   1               ;3RD STATUS WORD
    353
    354  001670  000001                DMCR17: .BLKW   1               ;CONTROL STATUS REGISTER FOR M8200-YC NUMBER 17
    355  001672  000001                DMS117: .BLKW   1               ;VECTOR FOR M8200-YC NUMBER 17
    356  001674  000001                DMS217: .BLKW   1               ;DDCMP LINE# FOR M8200-YC NUMBER 17
    357  001676  000001                DMS317: .BLKW   1               ;3RD STATUS WORD
    358
    359  001700  000000                DM.END: 000000
```

```
                                ;M8200-YC PASS COUNT AND ERROR COUNT TABLE
                                ;-----------------------------------------

001702                          CNT.MAP:
001702  000000                  PACT00: 0               ;PASS COUNT FOR M8200-YC NUMBER 00
001704  000000                  ERCT00: 0               ;ERROR COUNT FOR M8200-YC NUMBER 00

001706  000000                  PACT01: 0               ;PASS COUNT FOR M8200-YC NUMBER 01
001710  000000                  ERCT01: 0               ;ERROR COUNT FOR M8200-YC NUMBER 01

001712  000000                  PACT02: 0               ;PASS COUNT FOR M8200-YC NUMBER 02
001714  000000                  ERCT02: 0               ;ERROR COUNT FOR M8200-YC NUMBER 02

001716  000000                  PACT03: 0               ;PASS COUNT FOR M8200-YC NUMBER 03
001720  000000                  ERCT03: 0               ;ERROR COUNT FOR M8200-YC NUMBER 03

001722  000000                  PACT04: 0               ;PASS COUNT FOR M8200-YC NUMBER 04
001724  000000                  ERCT04: 0               ;ERROR COUNT FOR M8200-YC NUMBER 04

001726  000000                  PACT05: 0               ;PASS COUNT FOR M8200-YC NUMBER 05
001730  000000                  ERCT05: 0               ;ERROR COUNT FOR M8200-YC NUMBER 05

001732  000000                  PACT06: 0               ;PASS COUNT FOR M8200-YC NUMBER 06
001734  000000                  ERCT06: 0               ;ERROR COUNT FOR M8200-YC NUMBER 06

001736  000000                  PACT07: 0               ;PASS COUNT FOR M8200-YC NUMBER 07
001740  000000                  ERCT07: 0               ;ERROR COUNT FOR M8200-YC NUMBER 07

001742  000000                  PACT10: 0               ;PASS COUNT FOR M8200-YC NUMBER 10
001744  000000                  ERCT10: 0               ;ERROR COUNT FOR M8200-YC NUMBER 10

001746  000000                  PACT11: 0               ;PASS COUNT FOR M8200-YC NUMBER 11
001750  000000                  ERCT11: 0               ;ERROR COUNT FOR M8200-YC NUMBER 11

001752  000000                  PACT12: 0               ;PASS COUNT FOR M8200-YC NUMBER 12
001754  000000                  ERCT12: 0               ;ERROR COUNT FOR M8200-YC NUMBER 12

001756  000000                  PACT13: 0               ;PASS COUNT FOR M8200-YC NUMBER 13
001760  000000                  ERCT13: 0               ;ERROR COUNT FOR M8200-YC NUMBER 13

001762  000000                  PACT14: 0               ;PASS COUNT FOR M8200-YC NUMBER 14
001764  000000                  ERCT14: 0               ;ERROR COUNT FOR M8200-YC NUMBER 14

001766  000000                  PACT15: 0               ;PASS COUNT FOR M8200-YC NUMBER 15
001770  000000                  ERCT15: 0               ;ERROR COUNT FOR M8200-YC NUMBER 15

001772  000000                  PACT16: 0               ;PASS COUNT FOR M8200-YC NUMBER 16
001774  000000                  ERCT16: 0               ;ERROR COUNT FOR M8200-YC NUMBER 16

001776  000000                  PACT17: 0               ;PASS COUNT FOR M8200-YC NUMBER 17
002000  000000                  ERCT17: 0               ;ERROR COUNT FOR M8200-YC NUMBER 17
```

413

FORMAT OF STATUS TABLE




DEFINITION OF FORMAT

CSR: CONTAINS M8200-YC CSR ADDRESS

STAT1: BITS 00-08 IS M8200-YC VECTOR ADDRESS
BIT15=1 MICRO-PROCESSOR HAS CRAM
BIT15=0 MICRO-PROCESSOR HAS CROM
BIT14=1 TURNAROUND CONNECTOR IS ON
BIT14=0 NO TURNAROUND CONNECTOR
BIT13=0 LINE UNIT IS AN M8201
BIT13=1 LINE UNIT IS AN M8202
BIT12=1 NO LINE UNIT
BITS 09-11 IS M8200-YC BR PRIORITY LEVEL

STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3: BIT0=1 DO FREE RUNNING TESTS ON KMC

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

002002  012737  000340  177776  .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002010  012706  001200                  MOV     #STACK,SP       ;SET UP STACK
002014  012737  005346  000024          MOV     #.PFAIL,@#24    ;SET UP POWER FAIL VECTOR
002022  013737  001310  001314          MOV     DMNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002030  005037  010062                  CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
002034  105037  001325                  CLRB    ERRFLG          ;CLEAR ERROR FLAG
002040  105037  001327                  CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
002044  012737  001470  001320          MOV     #DM.MAP-10,CREAM;GET MAP POINTER.
002052  012737  001676  001322          MOV     #CNT.MAP-4,MILK ;GET PASS COUNT MAP POINTER
002060  012737  100000  001316          MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
002066  012700  001702                  MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
002072  005020                  23$:    CLR     (R0)+           ;CLEAR TABLE
002074  022700  002002                  CMP     #CNT.MAP+100,R0 ;DONE YET?
002100  001374                          BNE     23$             ;KEEP GOING
002102  005037  001234                  CLR     LSTERR          ;CLEAR LAST ERROR POINTER
002106  012737  000001  001226          MOV     #1,TSTNO        ;SET UP FOR TEST 1
002114  012737  002002  001214          MOV     #.START,RETURN  ;SET UP FOR POWER FAIL BEFORE
                                        ;TESTING STARTS
002122  013746  000006                  MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
002126  013746  000004                  MOV     @#4,-(SP)       ;
002132  012737  002166  000004          MOV     #6$,@#4         ;SET UP FOR TIMEOUT
002140  012737  177570  001202          MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
002146  012737  177570  001200          MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
002154  022777  177777  177020          CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
002162  001402                          BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
002164  000407                          BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
002166  022626                  6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002170  012737  000176  001202          MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002176  012737  000174  001200          MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002204  012637  000004                  MOV     (SP)+,@#4       ;RESTORE VECTORS
002210  012637  000006                  MOV     (SP)+,@#6       ;
002214  105737  001324                  TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002220  001012                          BNE     20$             ;BR IF YES
002222  022737  003532  000042          CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002230  001406                          BEQ     20$
002232  104402  001000                  TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002236  104402  027257                  TYPE    ,ROM1           ;TYPE VERSION MESSAGE
002242  104402  026462                  TYPE    ,MESWCH         ;TYPE SWITCH 7 MESSAGE
002246  004737  007652          20$:    JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
002252  017737  176724  001236          MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002260  005737  000042                  TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002264  001402                          BEQ     .+6             ;BR IF NO
002266  005037  001236                  CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002272  032737  000001  001236          BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002300  001012                          BNE     17$             ;BR IF SW00=1
002302  105737  001236                  TSTB    STRTSW          ;BIT7=1??
002306  100007                          BPL     17$             ;BR IF SW07=0
002310  005737  001306                  TST     DMACTV          ;ARE ANY DEVICES SELECTED?
```

```
519  002314  001006                        BNE     16$             ;BR IF YES
520  002316  104402  007201                TYPE,   NOACT           ;NO DEVICES SELECTED.
521  002322  000000                        HALT                    ;STOP THE SHOW
522  002324  000776                        BR      .-2             ;DISQUALIFY CONTINUE SWITCH
523  002326  004737  010556        17$:    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
524  002332  105737  001324        16$:    TSTB    INIFLG          ;FIRST TIME?
525  002336  001410                        BEQ     21$             ;BR IF YES
526  002340  105737  001236                TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
527  002344  100431                        BMI     1$
528  002346  032737  000006  001236        BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
529  002354  001403                        BEQ     24$             ;IF NO THEN TYPE STATUS
530  002356  000424                        BR      1$              ;IF YES DO NOT TYPE STATUS
531  002360  005137  001324        21$:    COM     INIFLG          ;SET FLAG
532  002364  104402  006240        24$:    TYPE    ,XHEAD          ;TYPE HEADER
533  002370  012704  001500                MOV     #DM.MAP,R4      ;SET POINTER
534  002374  010437  001246        5$:     MOV     R4,TEMP1        ;SET ADDRESS
535  002400  012437  001250                MOV     (R4)+,TEMP2     ;SET CSR
536  002404  001411                        BEQ     1$              ;ALL DONE IF ZERO
537  002406  012437  001252                MOV     (R4)+,TEMP3     ;SET STAT1
538  002412  012437  001254                MOV     (R4)+,TEMP4     ;SET STAT2
539  002416  012437  001256                MOV     (R4)+,TEMP5     ;SET STAT3
540  002422  104410                        CONVRT                  ;TYPE OUT STATUS MAP
541  002424  007520                        XSTATQ                  ;
542  002426  000762                        BR      5$
543  002430  012700  001500        1$:     MOV     #DM.MAP,R0      ;R0 POINTS TO STATUS TABLE
544
545                                 ;;**************************************************************************
546                                 ;;*AUTO SIZE TEST
547                                 ;;*THIS TEST VERIFYS THAT THE M8200-YCS AND/OR KMC11S ARE AT THE CORRECT FLOATING
548                                 ;;*ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
549                                 ;;*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC).
550                                 ;;*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE M8200-YC, THE FIRST
551                                 ;;*M8200-YC ADDRESS IS 760070, KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
552                                 ;;*ADDRESS 760000. THIS TEST MAY REQUIRE 2 OR MORE ATTEMPTS TO GET THE
553                                 ;;*RIGHT ADDRESSES. AFTER YOU HAVE CHANGED THE ADDRESS TO WHAT IT TOLD
554                                 ;;*YOU THE FIRST TIME, IT MAY COME BACK AND TELL YOU A DIFFERENT ADDRESS
555                                 ;;*THE NEXT TIME YOU RUN IT. PLEASE HAVE PATIENCE, THE FINAL ADDRESS
556                                 ;;*WILL BE CORRECT (AS LONG AS ALL DEVICES IN FRONT OF THE DMC'S ARE
557                                 ;;*CORRECT).
558                                 ;;**************************************************************************
559
560  002434  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
561  002440  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
562  002444  005037  000006                CLR     @#6             ;CLEAR VEC+2
563  002450  005037  001252                CLR     TEMP3           ;CLEAR FLAG
564  002454  005005                        CLR     R5              ;R5=0=DMC, R5=-1=KMC
565  002456  011037  001404        AUSTRT: MOV     (R0),DMCSR      ;GET NEXT DMC CSR
566  002462  001564                        BEQ     AUDONE          ;BR IF DONE
567  002464  005705                        TST     R5              ;DMC OR KMC?
568  002466  001005                        BNE     1$              ;BR IF KMC
569  002470  032760  100000  000002        BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
570  002476  001061                        BNE     SKIP            ;SKIP IF NOT DMC
571  002500  000404                        BR      2$              ;ITS A DMC SO CONTINUE
572  002502  032760  100000  000002  1$:    BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
573  002510  001454                        BEQ     SKIP            ;SKIP IF NOT KMC
574  002512  012737  002704  000004  2$:    MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
```

```
                                        TST     R5              ;DMC OR KMC?
                                        BNE     3$              ;BR IF KMC
                    000006              MOV     #6,R3           ;R3 IS COUNT OF DEVICES BEFORE DMC
                                        BR      4$              ;GO ON
                    000010      3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
                    003020      4$:     MOV     #DEVTAB,R2      ;R2 IS DEVICE TABLE PONTER
                    160010              MOV     #160010,R1      ;START WITH ADDRESS 160010
                                FLOAT:  TST     (R1)            ;CHECK ADDRESS IN R1
                                        MOVB    (R2),R4         ;IF NO TIMEOUT, GET NEXT ADDRESS
                                        ADD     R4,R1           ;IN R1
                                        INC     R1              ;
                                        BIC     R4,R1           ;
                                        TST     R3              ;ANY MORE DEVICES TO CHECK FOR?
                                        BNE     FLOAT           ;BR IF YES
                    002710  000004      MOV     #ERR,@#4        ;OK ONLY DMC'S ARE LEFT, SET UP FOR TIMEOUT
                    003032              MOV     R1,XLOC         ;SAVE FIRST DMC/KMC ADDRESS
                                FY:     TST     R5              ;DMC OR KMC?
                                        BNE     1$              ;BR IF KMC
                    100000  000002      BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
                                        BNE     SKIP            ;SKIP IF NOT DMC
                                        BR      2$              ;ITS A DMC SO CONTINUE
                    100000  000002  1$: BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
                                        BEQ     SKIP            ;SKIP IF NOT KMC
                                2$:     TST     (R1)            ;CHECK DMC ADDRESS
                    001404              CMP     R1,DMCSR        ;DOES IT MATCH
                                        BEQ     OK              ;BR IF YES
                    000010              ADD     #10,R1          ;GET NEXT DMC ADDRESS
                                        BR      FY              ;DO IT AGAIN
                    000010      SKIP:   ADD     #10,R0          ;SKIP TO NEXT CSR IN TABLE
                    001404              MOV     (R0),DMCSR      ;GET NEXT CSR
                                        BEQ     AUDONE          ;BR IF DONE
                                        BR      FY              ;ELSE CONTINUE
                    000010      OK:     ADD     #10,R0          ;SKIP TO NEXT DMC CSR
                    000010  003032      ADD     #10,XLOC        ;UPDATE EXPECTED DMC/KMC ADDRESS
                    001404              MOV     (R0),DMCSR      ;GET NEXT DMC/KMC CSR
                                        BEQ     AUDONE          ;BR IF DONE
                    003032              MOV     XLOC,R1         ;GET EXPECTED DMC/KMC ADDRESS
                                        BR      FY              ;CONTINUE
                                NODEV:  CMPB    (R2)+,-(R3)     ;ON TIMEOUT, INC R2, DEC R3
                                        RTI                     ;RETURN
                    001252      ERR:    TST     TEMP3           ;CHECK FLAG IF = 0 TYPE HEADER
                                        BNE     1$              ;SKIP HEADER
                                        TYPE                    ;TYPEOUT HEADER MESSAGE
                                        CONERR                  ;CONFIGURATION ERROR!!!!
                    002710  001276      MOV     #ERR,SAVPC      ;SAVE PC FOR TYPEOUT
                                        CNVRT                   ;TYPE OUT ERROR PC
                                        ERRPC                   ;
                                        TYPE                    ;TYPE REST OF HEADER
                                        CNERR                   ;
                    177777  001252      MOV     #-1,TEMP3       ;SET FLAG SO IT ONLY GETS TYPED ONCE
                    001262      1$:     MOV     R1,SAVR1        ;SAVE R1 FOR TYPEOUT
                                        CONVRT
                                        CONTAB                  ;TYPE CSR VALUES
                                        TST     R5              ;DMC OR KMC ?
                                        BNE     3$              ;BR IF KMC
                                        TYPE
```

```
                                        DMCM
                                        BR      4$              ;CONTINUE
                                3$:     TYPE
                                        KMCM
                                4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
                                        BR      OK              ;BR TO GET OUT
                                ERRPC:  1
                                        .BYTE   6,2
                                        SAVPC
                                CONTAB: 2
                                        .BYTE   6,4
                                        XLOC
                                        .BYTE   6,2
                                        DMCSR
                                DEVTAB: .BYTE   7               ;DJ
                                        .BYTE   17              ;DH
                                        .BYTE   7               ;DQ
                                        .BYTE   7               ;DU
                                        .BYTE   7               ;DUP
                                        .BYTE   7               ;LK
                                        .BYTE   7               ;DMC
                                        .BYTE   7               ;DZ
                                        .BYTE   7               ;KMC
                                .EVEN
                                XLOC:   0
                                AUDONE: TST     R5              ;DMC?
                                        BNE     1$              ;BR IF KMC AND ALL DONE
                                        MOV     #-1,R5          ;SET R5 TO -1 (KMC)
                                        MOV     #DM.MAP,R0      ;RESET R0 TO START OF TABLE
                                        BR      AUSTRT          ;GO DO KMC'S
                                1$:     MOV     (SP)+,@#6       ;RESTORE LOC 6
                                        MOV     (SP)+,@#4       ;RESTORE LOC 4
                                        BIT     #SW03,STRTSW    ;SELECT SPECIFIC DEVICES??
                                        BEQ     3$              ;BR IF NO.
                                        TYPE    ,MNEW           ;TYPE THE MESSAGE.
                                        CLR     R0              ;ZERO DATA LIGHTS
                                        HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
                                        CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
                                        BLOS    2$              ;BR IF NUMBER IS OK.
                                        TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
                                        HALT                    ;STOP EVERY THING.
                                        BR      .-2             ;RESTART THE PROGRAM AGAIN.
                                2$:     MOV     @SWR,DMACTV     ;GET NEW DEVICE PATTERN
                                        MOV     DMACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
                                        HALT                    ;CONTINUE DYNAMIC SWITCHES.
                                3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
                                        MOV     #302,R1         ;VECTOR AREA. 300-776
                                4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
                                        CLR     (R1)+           ;IN VECTOR AREA.
                                        CMP     (R0)+,(R1)+     ;POP POINTERS
                                        CMP     #1000,R0        ;ALL DONE??
                                        BNE     4$              ;BR IF NO.

                                ;TEST START AND RESTART
                                ;----------------------
```

```
   687  003162  012706  001200           .BEGIN: MOV     #STACK,SP       ;SET UP STACK
   688  003166  013746  000006                   MOV     @#6,-(SP)       ;SAVE LOC 6
   689  003172  013746  000004                   MOV     @#4,-(SP)       ;SAVE LOC 4
   690  003176  005000                            CLR     R0              ;START AT 0
   691  003200  012737  003244  000004           MOV     #2$,@#4         ;SET UP FOR TIME OUT
   692  003206  005037  000006                   CLR     @#6             ;TO AUTOSIZE MEMORY
   693  003212  005720                    6$:     TST     (R0)+           ;CHECK ADDRESS IN R0
   694  003214  022700  157776                   CMP     #157776,R0      ;IS IT AT LEAST 28K
   695  003220  001374                            BNE     6$              ;BR IF NO
   696  003222  162700  007776                   SUB     #7776,R0        ;SAVE 2K FOR MONITORS
   697  003226  010037  001304            7$:    MOV     R0,MEMLIM       ;STORE MEMORY LIMIT
   698  003232  012637  000004                   MOV     (SP)+,@#4       ;RESTORE LOC 4
   699  003236  012637  000006                   MOV     (SP)+,@#6       ;RESTORE LOC 6
   700  003242  000413                            BR      10$             ;CONTINUE
   701  003244  022626                    2$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
   702  003246  162700  000004                   SUB     #4,R0           ;GET LAST GOOD ADDRESS
   703  003252  162700  007776                   SUB     #7776,R0        ;SAVE 2K FOR MONITORS
   704  003256  022700  030000                   CMP     #30000,R0       ;IS IT 8K?
   705  003262  001361                            BNE     7$              ;BR IF NO
   706  003264  012700  037400                   MOV     #37400,R0       ;IF 8K DON'T SAVE 2K
   707  003270  000756                            BR      7$              ;
   708  003272  012737  000340  177776   10$:    MOV     #340,PS         ;LOCK OUT INTERRUPTS
   709  003300  032737  000004  001236           BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
   710  003306  001411                            BEQ     1$              ;BR IF NO LOCK DESIRED.
   711  003310  104402  006054                    TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
   712  003314  012737  000240  003622           MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
   713  003322  012737  000240  003624           MOV     #NOP,TTST+2     ;SET UP TO LOCK
   714  003330  000406                            BR      3$              ;CONTINUE ALONG.
   715  003332  013737  003740  003622   1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
   716  003340  013737  003742  003624           MOV     BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
   717  003346  012737  010124  001214   3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
   718  003354  032737  000002  001236   4$:     BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED?
   719  003362  001002                            BNE     5$              ;BR IF YES
   720  003364  104402  005766                    TYPE    ,MR             ;TYPE R
   721  003370  000177  175620            5$:     JMP     @RETURN         ;START TESTING
```

```
                                        ;END OF PASS
                                        ;TYPE NAME OF TEST
                                        ;UPDATE PASS COUNT
                                        ;CHECK FOR EXIT TO ACT-11
                                        ;RESTART TEST

003374  000005                  .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
003376  005037  001234                  CLR     LSTERR          ;CLEAR LAST ERROR PC
003402  105037  001325                  CLRB    ERRFLG          ;CLEAR ERROR FLAG
003406  005237  001230                  INC     PASCNT          ;UPDATE PASS COUNT
003412  013777  001230  175560          MOV     PASCNT,@DISPLAY ;DISPLAY PASS COUNT
003420  104402  005743                  TYPE    ,MEPASS         ;TYPE END PASS
003424  104402  006103                  TYPE    ,MCSRX          ;TYPE CSR
003430  104411  003556                  CNVRT   ,XCSR           ;SHOW IT
003434  104402  006111                  TYPE    ,MVECX          ;TYPE VECTOR
003440  104411  003564                  CNVRT   ,XVEC           ;SHOW IT
003444  104402  006117                  TYPE    ,MPASSX         ;TYPE PASSES
003450  104411  003572                  CNVRT   ,XPASS          ;SHOW IT
003454  104402  006130                  TYPE    ,MERRX          ;TYPE ERRORS
003460  104411  003600                  CNVRT   ,XERR           ;SHOW IT
003464  013700  001322                  MOV     MILK,R0         ;GET POINTER TO PASS COUNT
003470  013720  001230                  MOV     PASCNT,(R0)+    ;STORE PASS COUNT FOR THIS M8200-YC
003474  013720  001232                  MOV     ERRCNT,(R0)+    ;STORE ERROR COUNT FOR THIS M8200-YC
003500  005337  001314                  DEC     SAVNUM          ;ARE ALL DEVICES TESTED?
003504  001017                          BNE     RESTRT          ;BR IF NO.
003506  112737  000377  001327          MOVB    #377,QV.FLG     ;SET THE QUICK VERIFY FLAG.
003514  013737  001310  001314          MOV     DMNUM,SAVNUM    ;RESTORE THE COUNT
003522  013701  000042                  MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
003526  001406                          BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
003530  000005                          RESET                   ;STOP THE SHOW--CLEAR THE WORLD
003532                          $ENDAD:
003532  004711                          JSR     PC,(R1)
003534  000240                          NOP
003536  000240                          NOP
003540  000240                          NOP
003542  000240                          NOP
003544  012737  010124  001214  RESTRT: MOV     #CYCLE,RETURN
003552  000137  010124                  JMP     CYCLE
003556  000001                  XCSR:   1
003560     006     002                  .BYTE   6,2
003562  001404                          DMCSR
003564  000001                  XVEC:   1
003566     004     002                  .BYTE   4,2
003570  001374                          DMRVEC
003572  000001                  XPASS:  1
003574     006     002                  .BYTE   6,2
003576  001230                          PASCNT
003600  000001                  XERR:   1
003602     006     002                  .BYTE   6,2
003604  001232                          ERRCNT

                                        ;SCOPE LOOP AND INTERATION HANDLER
                                        ;--------------------------------

003606  004737  007652          .SCOPE: JSR     PC,CKSWR        ;CKECK FOR SOFT SWR
003612  010016                          MOV     R0,(SP)         ;SAVE R0 ON THE STACK
```

```
   778  003614  032777  040000  175360          BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
   779  003622  001407                  TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
   780  003624  000437                          BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
   781  003626  005737  003744                  TST     DONE            ;WAS TKCSR DONE SET?
   782  003632  001434                          BEQ     3$              ;BR IF NO (LOCKED ON TEST)
   783  003634  005037  003744                  CLR     DONE            ;YES, CLEAR FLAG
   784  003640  000415                          BR      2$              ;GO TO NEXT TEST
   785  003642  032777  004000  175332  1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
   786  003650  001011                          BNE     2$              ;BR IF YES
   787  003652  105737  001327                  TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
   788  003656  001406                          BEQ     2$              ;BR IF QUICK PASS.
   789  003660  005237  001224                  INC     LPCNT           ;UPDATE ITERATION COUNTER
   790  003664  023737  001224  001222          CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
   791  003672  101414                          BLOS    3$              ;BR IF NOT YET
   792  003674  105037  001325          2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
   793  003700  005037  001224                  CLR     LPCNT           ;START ICOUNTER AT 0
   794  003704  005037  001220                  CLR     LOCK
   795  003710  012737  000020  001222          MOV     #20,ICOUNT      ;RESET ITERATIONS
   796  003716  013737  001216  001214          MOV     NEXT,RETURN     ;GET NEXT TEST
   797  003724  011600                  3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
   798  003726  022626                          POP2SP                  ;FAKE AN "RTI"
   799  003730  013701  001404                  MOV     DMCSR,R1        ;R1 CONTAINS BASE M8200-YC ADDRESS
   800  003734  000177  175254                  JMP     @RETURN         ;GO DO THE TEST
   801  003740  001407                  BRW:    1407
   802  003742  000437                  BRX:    437
   803  003744  000000                  DONE:   0
   804
   805                                          ;CHECK FOR FREEZE ON CURRENT DATA
   806                                          ;--------------------------------
   807
   808  003746  004737  007652          .SCOP1: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
   809  003752  032777  001000  175222          BIT     #SW09,@SWR      ;IS SW09=1(SET)?
   810  003760  001405                          BEQ     1$              ;BR IF NOT SET.
   811  003762  005737  001220                  TST     LOCK
   812  003766  001402                          BEQ     1$
   813  003770  013716  001220                  MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
   814  003774  000002                  1$:     RTI                     ;GO BACK.
   815
   816                                          ;TELETYPE OUTPUT ROUTINE
   817                                          ;-----------------------
   818
   819  003776  010546                  .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
   820  004000  017605  000002                  MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
   821  004004  062766  000002  000002          ADD     #2,2(SP)        ;POP OVER ADDRESS.
   822  004012  005737  010062          4$:     TST     SWFLG           ;SOFT SWR MESSAGE?
   823  004016  001004                          BNE     1$              ;IF YES TYPE IT OUT REGARDLESS OF SW12
   824  004020  032777  010000  175154          BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
   825  004026  001012                          BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
   826  004030  105715                  1$:     TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
   827  004032  100002                          BPL     2$              ;BR IF NUMBER IS PLUS
   828  004034  104402  005702                  TYPE    ,MCRLF          ;TYPE A CR/LF!
   829  004040  105777  175144          2$:     TSTB    @TPCSR          ;TTY READY?
   830  004044  100375                          BPL     2$              ;BR IF NO.
   831  004046  112577  175140                  MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
   832  004052  001357                          BNE     4$              ;IF NOT ZERO KEEP PRINTING!
   833  004054  012605                  3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
                                        RTI                     ;GO HOME
                                        ;-----------------------
                                .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
                                        MOV     R4,-(SP)        ;SAVE R4 ON STACK
004064  010346
004062  010446
004064  017637  000004  004102          MOV     @4(SP),.MSG
004072  062766  000002  000004          ADD     #2,4(SP)
004100  104402                 .INST1: TYPE
004102  000000                 .MSG:   0
004104  012704  007546                  MOV     #INBUF,R4
004110  012703  000007                  MOV     #7,R3
004114  105777  175064          1$:     TSTB    @TKCSR
004120  100375                          BPL     1$
004122  117714  175060                  MOVB    @TKDBR,(R4)
004126  142714  000200                  BICB    #200,(R4)
004132  122427  000015                  CMPB    (R4)+,#15
004136  001417                          BEQ     INSTR2
004140  105777  175044          2$:     TSTB    @TPCSR
004144  100375                          BPL     2$
004146  017777  175034  175036          MOV     @TKDBR,@TPDBR
004154  005303                          DEC     R3
004156  001356                          BNE     1$
004160  012604                          MOV     (SP)+,R4
004162  012603                          MOV     (SP)+,R3
004164  104402  005676          .INSTE: TYPE    ,MQM
004170  010346                          MOV     R3,-(SP)
004172  010446                          MOV     R4,-(SP)
004174  000741                          BR      .INST1
004176  012604                  INSTR2: MOV     (SP)+,R4        ;RESTORE R4
004200  012603                          MOV     (SP)+,R3        ;RESTORE R3
004202  000002                          RTI

                                        ;CONVERT ASCII STRING TO OCTAL
                                        ;-----------------------------

004204  010546                  .PARAM: MOV     R5,-(SP)
004206  010446                          MOV     R4,-(SP)
004210  016605  000004                  MOV     4(SP),R5
004214  012537  004374                  MOV     (R5)+,LOLIM
004220  012537  004376                  MOV     (R5)+,HILIM
004224  012537  004400                  MOV     (R5)+,DEVADR
004230  112537  004402                  MOVB    (R5)+,LOBITS
004234  112537  004403                  MOVB    (R5)+,ADRCNT
004240  010566  000004                  MOV     R5,4(SP)
004244  005005                  PARAM1: CLR     R5
004246  012704  007546                  MOV     #INBUF,R4
004252  122714  000015                  CMPB    #15,(R4)
004256  001420                          BEQ     PARERR
004260  121427  000060          1$:     CMPB    (R4),#60
004264  002415                          BLT     PARERR
004266  121427  000067                  CMPB    (R4),#67
004272  003012                          BGT     PARERR
004274  142714  000060                  BICB    #60,(R4)
004300  152405                          BISB    (R4)+,R5
004302  122714  000015                  CMPB    #15,(R4)
004306  001406                          BEQ     LIMITS
```

```
                        ASL     R5
                        ASL     R5
                        ASL     R5
                        BR      1$
                PARERR: INSTER
                        BR      PARAM1

                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
                        ;--------------------------------------

                LIMITS: CMP     R5,HILIM
                        BHI     PARERR
                        CMP     R5,LOLIM
                        BLO     PARERR
                        BITB    LOBITS,R5
                        BNE     PARERR

                        ;STORE NUMBER AT SPECIFIED ADDRESS

                        MOV     DEVADR,R4
                1$:     MOV     R5,(R4)+
                        ADD     #2,R5
                        DECB    ADRCNT
                        BNE     1$
                        MOV     (SP)+,R4
                        MOV     (SP)+,R5
                        RTI
                LOLIM:  0
                HILIM:  0
                DEVADR: 0
                LOBITS: 0
                ADRCNT=LOBITS+1

                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
                        ;-------------------------------------

                .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                        ;SAVE R0-R5

                SV05:   MOV     R5,SAVR5        ;SAVE R5
                        MOV     R4,SAVR4        ;SAVE R4
                        MOV     R3,SAVR3        ;SAVE R3
                        MOV     R2,SAVR2        ;SAVE R2
                        MOV     R1,SAVR1        ;SAVE R1
                        MOV     R0,SAVR0        ;SAVE R0
                        RTI                     ;LEAVE.

                        ;RESTORE R0-R5

                .RES05: MOV     SAVR0,R0        ;RESTORE R0
                        MOV     SAVR1,R1        ;RESTORE R1
                        MOV     SAVR2,R2        ;RESTORE R2
                        MOV     SAVR3,R3        ;RESTORE R3
                        MOV     SAVR4,R4        ;RESTORE R4
                        MOV     SAVR5,R5        ;RESTORE R5
```

```
                        RTI                     ;LEAVE

                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                        ;-------------------------------------------------------

004476  104402  005702  .CONVR: TYPE    ,MCRLF
004502  010046          .CNVRT: MOV     R0,-(SP)
004504  010146                  MOV     R1,-(SP)
004506  010346                  MOV     R3,-(SP)
004510  010446                  MOV     R4,-(SP)
004512  010546                  MOV     R5,-(SP)
004514  017601  000012          MOV     @12(SP),R1
004520  062766  000002  000012  ADD     #2,12(SP)
004526  012137  004720          MOV     (R1)+,WRDCNT
004532  112137  004722  1$:     MOVB    (R1)+,CHRCNT
004536  112137  004723          MOVB    (R1)+,SPACNT
004542  013137  004724          MOV     @(R1)+,BINWRD
004546  122737  000003  004722  CMPB    #3,CHRCNT
004554  001003                  BNE     2$
004556  042737  177400  004724  BIC     #177400,BINWRD
004564  013704  004724  2$:     MOV     BINWRD,R4
004570  113705  004722          MOVB    CHRCNT,R5
004574  012700  001416          MOV     #TEMP,R0
004600  010403          3$:     MOV     R4,R3
004602  042703  177770          BIC     #177770,R3
004606  062703  000060          ADD     #060,R3
004612  110320                  MOVB    R3,(R0)+
004614  000241                  CLC
004616  006004                  ROR     R4
004620  000241                  CLC
004622  006004                  ROR     R4
004624  000241                  CLC
004626  006004                  ROR     R4
004630  005305                  DEC     R5
004632  001362                  BNE     3$
004634  012703  007610          MOV     #MDATA,R3
004640  114023          4$:     MOVB    -(R0),(R3)+
004642  105337  004722          DECB    CHRCNT
004646  001374                  BNE     4$
004650  105737  004723          TSTB    SPACNT
004654  001405                  BEQ     6$
004656  112723  000040  5$:     MOVB    #040,(R3)+
004662  105337  004723          DECB    SPACNT
004666  001373                  BNE     5$
004670  105013          6$:     CLRB    (R3)
004672  104402  007610          TYPE    ,MDATA
004676  005337  004720          DEC     WRDCNT
004702  001313                  BNE     1$
004704  012605                  MOV     (SP)+,R5
004706  012604                  MOV     (SP)+,R4
004710  012603                  MOV     (SP)+,R3
004712  012601                  MOV     (SP)+,R1
004714  012600                  MOV     (SP)+,R0
004716  000002                  RTI
004720  000000  WRDCNT: 0
004722  000000  CHRCNT: 0
```

```
1002            004723                  SPACNT=CHRCNT+1
1003  004724    000000                  BINWRD: 0
1004
1005
1006                                    ;TRAP DISPATCH SERVICE
1007                                    ;ARGUMENT OF TRAP IS EXTRACTED
1008                                    ;AND USED AS OFFSET TO OBTAIN POINTER
1009                                    ;TO SELECTED SUBROUTINE
1010
1011  004726    011646                  .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
1012  004730    162716    000002                SUB     #2,(SP)         ;=PC OF TRAP
1013  004734    017616    000000                MOV     @(SP),(SP)      ;GET TRP
1014  004740    006316                  TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
1015  004742    042716    177001                BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
1016  004746    062716    001330                ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
1017  004752    017616    000000                MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
1018  004756    000136                        JMP     @(SP)+          ;GO TO SUBROUTINE
1019
1020                                    ;ERROR HANDLER
1021                                    ;-------------
1022
1023  004760    004737    007652        .HLT:   JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
1024  004764    032777    010000  174210        BIT     #SW12,@SWR      ;BELL ON ERROR?
1025  004772    001406                        BEQ     XBX             ;BR IF NO BELL
1026  004774    105777    174210              TSTB    @TPCSR          ;TTY READY.
1027  005000    100003                        BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
1028  005002    112777    000207  174202        MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
1029  005010    032777    020000  174164  XBX:  BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1030  005016    001105                        BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
1031  005020    021637    001234              CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
1032  005024    001404                        BEQ     1$              ;BR IF YES
1033  005026    011637    001234              MOV     (SP),LSTERR     ;RECORD BEING HERE
1034  005032    105037    001325              CLRB    ERRFLG          ;PREPARE HEADER
1035  005036    104406                  1$:     SAV05                   ;SAVE ALL PROC REGISTERS
1036  005040    011605                        MOV     (SP),R5         ;GET THE PC OF ERROR
1037  005042    162705    000002              SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1038  005046    011504                        MOV     (R5),R4         ;GET HLT INSTRUCTION
1039  005050    006304                        ASL     R4              ;MULT BY TWO
1040  005052    061504                        ADD     (R5),R4         ;DOUBLE IT
1041  005054    006304                        ASL     R4              ;MULT AGAIN
1042  005056    042704    177001              BIC     #177001,R4      ;CLEAR JUNK
1043  005062    062704    027644              ADD     #.ERRTAB,R4     ;GET POINTER
1044  005066    012437    005202              MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1045  005072    012437    005214              MOV     (R4)+,DATAHD    ;GET DATA HEADRER
1046  005076    011437    005226              MOV     (R4),DATABP     ;GET DATA TABLE
1047  005102    105737    001325              TSTB    ERRFLG          ;TYPE HEADREER
1048  005106    001403                        BEQ     TYPMSG          ;BR IF YES
1049  005110    005737    005226              TST     DATABP          ;DOES DATA TABLE EXIST?
1050  005114    001040                        BNE     TYPDAT          ;BR IF YES.
1051  005116    104402    005702  TYPMSG:     TYPE    ,MCRLF
1052  005122    104402    005702              TYPE    ,MCRLF
1053  005126    005737    001220              TST     LOCK
1054  005132    001402                        BEQ     1$
1055  005134    104402    006153              TYPE    ,MASTEK
1056  005140    104402    006141  1$:         TYPE    ,MTSTN
1057  005144    104411    005340              CNVRT   ,XTSTN          ;SHOW IT
```

```
1058  005150  104402  006233          TYPE    ,MERRPC         ;TYPE PC.
1059  005154  104411  005332          CNVRT   ,ERTABO         ;SHOW IT
1060  005160  104402  005702          TYPE    ,MCRLF          ;GIVE A CR/LF
1061  005164  112737  177777  001325  MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1062  005172  005737  005202          TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1063  005176  001402                  BEQ     WRKO.FM         ;BR IF NO.
1064  005200  104402                  TYPE                    ;TYPE
1065  005202  000000          ERRMSG: 0                       ;    ERROR MESSAGE
1066  005204                  WRKO.FM:                        ;
1067  005204  005737  005214          TST     DATAHD          ;DATA HEADER?
1068  005210  001402                  BEQ     TYPDAT          ;BR IF NO
1069  005212  104402                  TYPE                    ;TYPE
1070  005214  000000          DATAHD: 0                       ;    DATA HEADER
1071  005216  005737  005226  TYPDAT: TST     DATABP          ;DATA TABLE?
1072  005222  001402                  BEQ     RESREG          ;BR IF NO.
1073  005224  104410                  CONVRT                  ;SHOW
1074  005226  000000          DATABP: 0                       ;    DATA TABLE
1075  005230  104407          RESREG: RES05                   ;RESTORE PROC REGISTERS
1076  005232  022737  003532  000042  HALTS:  CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, HALT!!
1077  005240  001403                  BEQ     1$
1078  005242  005777  173734          TST     @SWR            ;HALT ON ERROR?
1079  005246  100005                  BPL     EXITER          ;BR IF NO HALT ON ERROR
1080  005250  010046          1$:     PUSHR0                  ;SAVE R0
1081  005252  016600  000002          MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
1082  005256  000000                  HALT                    ;HALT
1083  005260  012600                  POPR0                   ;GET R0
1084  005262  005237  001232  EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
1085  005266  032777  000400  173706  BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1086  005274  001007                  BNE     1$              ;BR IF YES
1087  005276  032777  002000  173676  BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1088  005304  001411                  BEQ     2$              ;BR IF NO
1089  005306  013737  001216  001214  MOV     NEXT,RETURN     ;SET FOR NEXT TEST
1090  005314  012706  001200  1$:     MOV     #STACK,SP       ;RESET SP
1091  005320  013701  001404          MOV     DMCSR,R1        ;SET UP R1
1092  005324  000177  173664          JMP     @RETURN         ;GOTO SPECIFIED TEST
1093  005330  000002          2$:     RTI                     ;RETURN
1094  005332  000001          ERTABO: 1
1095  005334     006     002          .BYTE   6,2
1096  005336  001276                  SAVPC
1097  005340  000001          XTSTN:  1
1098  005342     003     002          .BYTE   3,2
1099  005344  001226                  TSTNO
1100                                  ;ENTER HERE ON POWER FAILURE
1101                                  ;--------------------------
1102
1103
1104  005346                  .PFAIL:
1105  005346  012737  005360  000024          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
1106  005354  000000                          HALT                            ;HALT ON POWER DOWN NORMAL
1107  005356  000777                          BR      .
1108
1109                                  ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1110
1111  005360                  RESTAR:
1112  005360  012737  005346  000024          MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
1113  005366  012706  001200                  MOV     #STACK,SP       ;RESET THE STACK POINTER
```

```
1114  005372  013701  001404              MOV     DMCSR,R1        ;RESTORE R1
1115  005376  005037  001416              CLR     TEMP            ;READY FOR TIMMER
1116  005402  005237  001416              INC     TEMP            ;PLUS ONE TO THE TIMER!
1117  005406  001375                      BNE     .-4             ;BR IF MORE TO GO
1118  005410  104402  005705              TYPE    ,MPFAIL         ;TYPE THE MESSAGE
1119  005414  104411  005440              CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
1120  005420  105037  001325              CLRB    ERRFLG          ;START CLEAN
1121  005424  005037  001234              CLR     LSTERR          ;'''''''''''''''
1122  005430  005011                      CLR     (R1)            ;CLEAR MAINT BITS
1123  005432  104412                      MSTCLR                  ;START CLEAN UP OF DEVICE
1124  005434  000177  173554              JMP     @RETURN         ;START DOING THAT TEST AGAIN.
1125  005440  000001              PFTAB:  1
1126  005442     003     002      .BYTE   3,2
1127  005444  001226                      TSTNO
1128
1129  005446                      .DELAY:
1130  005446  012777  000020  173736      MOV     #20,@DMP04
1131  005454  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1132  005456  121111                      121111                  ;POKE CLOCK DELAY BIT
1133  005460                      1$:
1134  005460  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1135  005462  121224                      121224                  ;PORT4_IBUS*11
1136  005464  032777  000020  173720      BIT     #BIT4,@DMP04    ;IS CLOCK BIT SET?
1137  005472  001772                      BEQ     1$              ;BR IF NO
1138  005474  000002                      RTI
1139
1140  005476                      .MSTCLR:
1141  005476  152777  000100  173702      BISB    #BIT6,@DMCSRH   ;SET MASTER CLEAR
1142  005504  142777  000300  173674      BICB    #BIT6!BIT7,@DMCSRH  ;CLEAR MASTER CLEAR AND RUN
1143  005512  000002                      RTI                     ;RETURN
1144
1145  005514                      .ROMCLK:
1146  005514  152777  000002  173664      BISB    #BIT1,@DMCSRH   ;SET ROMI
1147  005522  013677  173666              MOV     @(SP)+,@DMP06   ;LOAD INSTRUCTION IN SEL6
1148  005526  062746  000002              ADD     #2,-(SP)        ;ADJUST STACK
1149  005532  032777  000100  173442      BIT     #SW06,@SWR      ;HALT IF SW06 =1
1150  005540  001401                      BEQ     1$              ;BR IF SW06 =0
1151  005542  000000                      HALT                    ;HALT BEFORE CLOCKING INSTRUCTION
1152  005544  152777  000003  173634  1$: BISB    #BIT1!BIT0,@DMCSRH ;CLOCK INSTRUCTION
1153  005552  142777  000007  173626      BICB    #BIT2!BIT1!BIT0,@DMCSRH  ;CLEAR ROMO, ROMI, STEP
1154  005560  000002                      RTI
1155
1156  005562                      .DATACLK:
1157  005562  013637  001416              MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
1158  005566  062746  000002              ADD     #2,-(SP)        ;ADJUST STACK
1159  005572  152777  000020  173606  1$: BISB    #BIT4,@DMCSRH   ;SET STEP LU
1160  005600  027777  173600  173576      CMP     @DMCSR,@DMCSR   ;WASTE TIME
1161  005606  142777  000020  173572      BICB    #BIT4,@DMCSRH   ;CLEAR STEP LU
1162  005614  005337  001416              DEC     TEMP            ;DEC TICK COUNT
1163  005620  001364                      BNE     1$              ;BR IF NOT DONE
1164  005622  000002                      RTI                     ;RETURN
1165  005624  000001              3$:     .BLKW 1
1166
1167  005626                      .TIMER:
1168  005626  013637  001416              MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
1169  005632  062746  000002              ADD     #2,-(SP)        ;ADJUST STACK
```

```
1170  005636                          1$:
1171  005636  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1172  005640  021364                          021364                  ;PORT4 IBUS* REG11
1173  005642  032777  000002  173542          BIT     #2,@DMPO4       ;IS PGM CLOCK BIT CLEAR?
1174  005650  001772                          BEQ     1$              ;BR IF YES
1175  005652                          2$:
1176  005652  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1177  005654  021364                          021364                  ;PORT4 IBUS* REG11
1178  005656  032777  000002  173526          BIT     #2,@DMPO4       ;IS PGM CLOCK BIT SET?
1179  005664  001372                          BNE     2$              ;BR IF YES
1180  005666  005337  001416                  DEC     TEMP            ;DEC COUNT
1181  005672  001361                          BNE     1$              ;BR IF NOT DONE
1182  005674  000002                          RTI                     ;RETURN
1183
1184  005676  020040  000077          MQM:    .ASCIZ  /  ?/
 (2)  005702  005015     000          MCRLF:  .ASCIZ   <15><12>
 (2)  005705     377  053520  020122  MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
 (2)  005743     377  047105  020104  MEPASS: .ASCIZ  <377>/END PASS CRLPMB  /
 (2)  005766  051377     000          MR:     .ASCIZ  <377>/R/
 (2)  005771     377  047516  042040  MERR2:  .ASCIZ  <377>/NO DEVICES PRESENT./
 (2)  006016  044777  051516  043125  MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
 (2)  006042  052377  051505  020124  MTSTPC: .ASCIZ  <377>/TEST PC-/
 (2)  006054  046377  041517  020113  MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
 (2)  006103     103  051123  020072  MCSRX:  .ASCIZ   /CSR: /
 (2)  006111     126  041505  020072  MVECX:  .ASCIZ  /VEC: /
 (2)  006117     120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
 (2)  006130  051105  047522  051522  MERRX:  .ASCIZ  /ERRORS: /
 (2)  006141     124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
 (2)  006153     052     000          MASTEK: .ASCIZ  /*/
 (2)  006155     377  042523  020124  MNEW:   .ASCIZ  <377>/SET SWITCH REG TO M8200-YC'S DESIRED ACTIVE./
 (2)  006233     120  035103  000040  MERRPC: .ASCIZ  /PC: /
 (2)  006240  020212  020040  020040  XHEAD:  .ASCII  <212>/           MAP OF M8200-YC STATUS/
 (2)  006302  020377  020040  020040          .ASCII  <377>/           ----------------------/
 (2)  006344  020212  050040  020103          .ASCII  <212>/  PC      CSR     STAT1    STAT2    STAT3/
 (2)  006416  026777  026455  026455          .ASCIZ  <377>/------   ------   ------   ------   ------/
 (2)  006472  044377  053517  046440  NUM:    .ASCIZ  <377>/HOW MANY M8200-YC'S TO BE TESTED?/
 (2)  006535     377  051503  020122  CSR:    .ASCIZ  <377>/CSR ADDRESS?/
 (2)  006553     377  042526  052103  VEC:    .ASCIZ  <377>/VECTOR ADDRESS?/
 (2)  006574  041377  020122  051120  PRIO:   .ASCIZ  <377>/BR PRIORITY LEVEL? (4,5,6,7)?/
 (2)  006633     377  043111  042040  CRAM:   .ASCIZ  <377>/IF DMC HAS CRAM (M8204) TYPE "Y", IF CROM (M8200) TYPE "N" ?/
 (2)  006731     377  044127  041511  MODU:   .ASCIZ  <377>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M8202 TYP
 (2)  007043     377  053523  052111  LINE:   .ASCIZ  <377>/SWITCH PAC#1 (DDCMP LINE #)?/
 (2)  007101     377  053523  052111  BM:     .ASCIZ  <377>/SWITCH PAC#2 (BM873 BOOT ADD)?/
 (2)  007141     377  051511  052040  CONN:   .ASCIZ  <377>/IS THE LOOP BACK CONNECTOR ON?/
 (2)  007201     377  047516  042040  NOACT:  .ASCIZ  <377>/NO DEVICES ARE SELECTED/
 (2)  007232  005377  053523  036522  SWMES:  .ASCIZ  <377><12>/SWR= /
 (2)  007242  042516  037527  000040  SWMES1: .ASCIZ  /NEW? /
 (2)  007250  177777  034115  030062  CONERR: .ASCIZ  <377><377>/M8200-YC FOUND AT NON-STANDARD ADDRESS  PC: /
 (2)  007327     377  054105  042520  CNERR:  .ASCIZ  <377>/EXPECTED  FOUND/
 (2)  007350  024040  034115  030062  DMCM:   .ASCIZ  / (M8200-YC)  /
 (2)  007366  024040  046513  024503  KMCM:   .ASCIZ  / (KMC) /
 (2)  007376  046777  031070  030060  SPEED:  .ASCIZ  <377>/M8200-YC-AR(REMOTE,LOW SPEED) OR M8200-YC-AL(LOCAL,HIGH SPEED) TYP
 (2)                                  .EVEN
 (2)  007520  000005                  XSTATQ: 5
1185  007522     006     003                  .BYTE   6,3
1186  007524  001246                          TEMP1
```

```
1187  007526  000006  003                   .BYTE   6,3
1188  007530  001250                        TEMP2
1189  007532  000006  003                   .BYTE   6,3
1190  007534  001252                        TEMP3
1191  007536  000006  003                   .BYTE   6,3
1192  007540  001254                        TEMP4
1193  007542  000006  002                   .BYTE   6,2
1194  007544  001256                        TEMP5
1195                            .EVEN
1196
1197                                        ;BUFFERS FOR INPUT-OUTPUT
1198
1199  007546  000000            INBUF:    0
1200          007610                      .=.+40
1201  007610  000000            MDATA:    0
1202          007652                      .=.+40
1203
1204
1205                                        ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1206                                        ;REGISTER USING THE CONSOLE TERMINAL
1207                                        ;-------------------------------------
1208
1209  007652  022737  000176  001202  CKSWR:   CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
1210  007660  001077                        BNE     CKSWR5          ;BR IF NO
1211  007662  105777  171316                TSTB    @TKCSR          ;IS DONE SET?
1212  007666  100003                        BPL     2$              ;GO ON IF NOT SET
1213  007670  012737  177777  003744        MOV     #-1,DONE        ;IF DONE SET, SET FLAG
1214  007676  022777  000007  171302  2$:     CMP     #7,@TKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
1215  007704  001404                        BEQ     1$              ;BR IF YES
1216  007706  022777  000207  171272        CMP     #207,@TKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII)
1217  007714  001061                        BNE     CKSWR5          ;BR IF NO
1218  007716  010246                1$:     MOV     R2,-(SP)        ;STORE R2
1219  007720  010346                        MOV     R3,-(SP)        ;STORE R3
1220  007722  010446                        MOV     R4,-(SP)        ;STORE R4
1221  007724  012737  177777  010062        MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
1222  007732  005002            CKSWR1:   CLR     R2              ;CLEAR NEW SWR CONTENTS
1223  007734  012704  177777                MOV     #-1,R4          ;SET FLAG TO ALL ONES
1224  007740  104402  007232                TYPE    ,SWMES          ;TYPE "SWR= "
1225  007744  104411            CKSWR2:   CNVRT                   ;TYPE OUT PRESENT CONTENTS
1226  007746  010116                        SOFTSW                  ;OF SOFT SWITCH REGISTER
1227  007750  104402  007242    CKSWR3:   TYPE    ,SWMES1         ;TYPE "NEW? "
1228  007754  004737  010064    CKSWR4:   JSR     PC,INCHAR       ;GET RESPONSE
1229  007760  022703  000015                CMP     #15,R3          ;WAS IT A CR?
1230  007764  001424                        BEQ     5$              ;BR IF YES
1231  007766  022703  000012                CMP     #12,R3          ;WAS IT A LF?
1232  007772  001416                        BEQ     4$              ;BR IF YES
1233  007774  022703  000025                CMP     #25,R3          ;WAS IT CTRL U?
1234  010000  001754                        BEQ     CKSWR1          ;BR IF YES(START OVER)
1235  010002  022703  000007                CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
1236  010006  001762                        BEQ     CKSWR4
1237  010010  005004                        CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
1238  010012  042703  177770                BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1239  010016  006302                        ASL     R2              ;SHIFT R2 3 TIMES
1240  010020  006302                        ASL     R2
1241  010022  006302                        ASL     R2
1242  010024  050302                        BIS     R3,R2           ;ADD LAST DIGIT
```

```
1243  010026  000752                        BR      CKSWR4          ;GET NEXT CHARACTER
1244  010030  012766  002002  000006  4$:   MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
1245  010036  005704                  5$:   TST     R4              ;IS FLAG CLEAR?
1246  010040  001002                        BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
1247  010042  010277  171134                MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
1248  010046  005037  010062          6$:   CLR     SWFLG           ;CLEAR TYPEOUT FLAG
1249  010052  012604                        MOV     (SP)+,R4        ;RESTORE R4
1250  010054  012603                        MOV     (SP)+,R3        ;RESTORE R3
1251  010056  012602                        MOV     (SP)+,R2        ;RESTORE R2
1252  010060  000207                  CKSWR5: RTS     PC              ;RETURN
1253
1254  010062  000000                  SWFLG:  0
1255
1256  010064  105777  171114          INCHAR: TSTB    @TKCSR
1257  010070  100375                        BPL     .-4
1258  010072  017703  171110                MOV     @TKDBR,R3
1259  010076  105777  171106                TSTB    @TPCSR
1260  010102  100375                        BPL     .-4
1261  010104  010377  171102                MOV     R3,@TPDBR
1262  010110  042703  000200                BIC     #BIT7,R3
1263  010114  000207                        RTS     PC
1264
1265  010116  000001                  SOFTSW: 1
1266  010120     006     002                 .BYTE   6,2
1267  010122  000176                        SWREG
```

```
1268
1269                                        ;
1270                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO 16 M8200-YC'S
1271                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1272                                        ;AND RUNS THE SPECIFIED M8200-YC'S.   THIS ROUTINE *MUST*
1273                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1274                                        ;SETUP NECESSARY.
1275                                        ;
1276
1277  010124  005737  001306        CYCLE:  TST     DMACTV          ;ARE ANY M8200-YC'S TO BE TESTED?
1278  010130  001004                        BNE     1$              ;BR IF OK.
1279  010132  104402  007201                TYPE    ,NOACT          ;NO M8200-YC'S SELECTED!!
1280  010136  000000                        HALT                    ;STOP THE SHOW.
1281  010140  000776                        BR      .-2             ;DISQUALIFY CONT. SW.
1282  010142  000241                1$:     CLC                     ;CLEAR PROC. CARRY BIT.
1283  010144  006137  001316                ROL     RUN             ;UPDATE POINTER
1284  010150  005537  001316                ADC     RUN             ;CATCH CARRY FROM RUN
1285  010154  062737  000004  001322        ADD     #4,MILK         ;UPDATE POINTER
1286  010162  062737  000010  001320        ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
1287  010170  022737  001700  001320        CMP     #DM.MAP+200,CREAM
1288  010176  001006                        BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
1289  010200  012737  001500  001320        MOV     #DM.MAP,CREAM   ;RESET ADDRESS POINTER.
1290  010206  012737  001702  001322        MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
1291  010214  033737  001316  001306  2$:   BIT     RUN,DMACTV      ;IS THIS ONE ACTIVE?
1292  010222  001747                        BEQ     1$              ;BR IF NO
1293  010224  013700  001320                MOV     CREAM,R0        ;GET ADDRESS POINTER
1294  010230  013702  001322                MOV     MILK,R2         ;GET PASS COUNT POINTER
1295  010234  012037  001404                MOV     (R0)+,DMCSR     ;LOAD SYSTEM CTRL. REG
1296  010240  011037  001374                MOV     (R0),DMRVEC     ;LOAD VECTOR
1297  010244  042737  177000  001374        BIC     #177000,DMRVEC  ;CLEAR UNWANTED BITS
1298  010252  012037  001366                MOV     (R0)+,STAT1     ;LOAD STAT1
1299  010256  012037  001370                MOV     (R0)+,STAT2     ;LOAD STAT2
1300  010262  012037  001372                MOV     (R0)+,STAT3     ;LOAD STAT3
1301  010266  012237  001230                MOV     (R2)+,PASCNT    ;LOAD PASS COUNT
1302  010272  012237  001232                MOV     (R2)+,ERRCNT    ;LOAD ERROR COUNT
1303  010276  012700  000002                MOV     #2,R0           ;SAVE CORE THIS WAY!
1304  010302  013737  001404  001406        MOV     DMCSR,DMCSRH
1305  010310  005237  001406                INC     DMCSRH
1306  010314  013737  001406  001410        MOV     DMCSRH,DMCTL
1307  010322  005237  001410                INC     DMCTL
1308  010326  013737  001410  001412        MOV     DMCTL,DMP04
1309  010334  060037  001412                ADD     R0,DMP04
1310  010340  013737  001412  001414        MOV     DMP04,DMP06
1311  010346  060037  001414                ADD     R0,DMP06
1312
1313  010352  013737  001374  001376        MOV     DMRVEC,DMRLVL   ;PTY LVL
1314  010360  060037  001376                ADD     R0,DMRLVL       ;
1315  010364  013737  001376  001400        MOV     DMRLVL,DMTVEC   ;TX VEC
1316  010372  060037  001400                ADD     R0,DMTVEC       ;
1317  010376  013737  001400  001402        MOV     DMTVEC,DMTLVL   ;TX LVL
1318  010404  060037  001402                ADD     R0,DMTLVL
1319
1320  010410  032737  000002  001236        BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
1321  010416  001450                        BEQ     7$              ;BR IF NO
1322  010420                        4$:
1323  010420  005737  000042                TST     @#42            ;RUNNING IN AUTO MODE?
```

```
1324  010424  001045                        BNE     7$              ;BR IF YES
1325  010426  104402  005702                TYPE    ,MCRLF
1326  010432  104403                        INSTR                   ;GET TEST NO.
1327  010434  006141                        MTSTN
1328  010436  104405                        PARAM
1329  010440  000001                        1
1330  010442  001000                        1000
1331  010444  001226                        TSTNO
1332  010446     000                .BYTE   0
1333  010447     001                .BYTE   1
1334  010450  012700  022404                MOV     #TST1,R0
1335  010454  022710                5$:     CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
1336  010456  012737                        MOV     (PC)+,@(PC)+
1337  010460  001020                        BNE     6$              ;BR IF NOT SAME
1338  010462  023760  001226  000002        CMP     TSTNO,2(R0)     ;DOES TSTNO MATCH?
1339  010470  001014                        BNE     6$              ;BR IF NO
1340  010472  022760  001226  000004        CMP     #TSTNO,4(R0)    ;IS LAST WORD OK?
1341  010500  001010                        BNE     6$              ;BR IF NO
1342  010502  010037  001214                MOV     R0,RETURN       ;IT IS A LEGAL TEST SO DO IT
1343  010506  104402  005766                TYPE    ,MR
1344  010512  042737  000002  001236        BIC     #SW01,STRTSW
1345  010520  000412                        BR      8$
1346  010522  005720                6$:     TST     (R0)+           ;POP R0
1347  010524  020027  026114                CMP     R0,#TLAST+10    ;AT END YET?
1348  010530  001351                        BNE     5$              ;BR IF NO
1349  010532  104402  005676                TYPE    ,MQM            ;YES ILLEGAL TEST NO.
1350  010536  000730                        BR      4$              ;TRY AGAIN
1351
1352  010540  012737  022404  001214  7$:   MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
1353  010546  013701  001404        8$:     MOV     DMCSR,R1        ;R1 = BASE M8200-YC ADDRESS
1354  010552  000177  170436                JMP     @RETURN         ;GO START TESTING.
1355
1356
1357                                        ;ROUTINE USED TO "AUTO SIZE" THE M8200-YC
1358                                        ;CSR AND VECTOR.
1359                                        ;NOTE:  THE CSR MAY BE ANY WHERE IN THE
1360                                        ;       ADDRESS RANGE (170440:170510)
1361                                        ;       AND THE VECTOR MAY BE ANY WHERE IN THE
1362                                        ;       FLOATING VECTOR RANGE (300:770)
1363                                        ;
1364
1365  010556                        AUTO.SIZE:
1366  010556  000005                        RESET                   ;INSURE A BUS INIT.
1367  010560  012702  001500        CSRMAP: MOV     #DM.MAP,R2      ;LOAD MAP POINTER.
1368  010564  005022                1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1369  010566  022702  001700                CMP     #DM.END,R2      ;ALL DONE?
1370  010572  001374                        BNE     1$              ;BR IF NO
1371  010574  005037  001310                CLR     DMNUM           ;SET OCTAL NUMBER OF M8200-YC'S TO 0
1372  010600  012702  001500                MOV     #DM.MAP,R2      ;R2 POINTS TO M8200-YC MAP
1373  010604  005037  001306                CLR     DMACTV          ;CLEAR ACTIVE
1374  010610  032737  000001  001236        BIT     #SW00,STRTSW    ;QUESTIONS?
1375  010616  001002                        BNE     .+6             ;BR IF YES
1376  010620  000137  011326                JMP     7$              ;IF NO SKIP QUESTIONS
1377  010624  012737  000001  001256        MOV     #1,TEMP5        ;START WITH 1
1378  010632  104403                        INSTR
1379  010634  006472                        NUM
```

```
1380  010636  104405                          PARAM
1381  010640  000001                          1
1382  010642  000020                          16.
1383  010644  001252                          TEMP3
1384  010646     000                          .BYTE   0
1385  010647     001                          .BYTE   1
1386  010650  013737  001252  001310          MOV     TEMP3,DMNUM     ;DMNUM = HOW MANY
1387  010656  104402  005702          12$:    TYPE    ,MCRLF
1388  010662  104410                          CONVRT                  ;TYPE WHICH DMC IS BEING DONE
1389  010664  012060                          WHICH                   ;TEMP5 IS WHICH DMC
1390  010666  005237  001256                  INC     TEMP5
1391  010672  104403                          INSTR
1392  010674  006535                          CSR
1393  010676  104405                          PARAM
1394  010700  170440                          170440
1395  010702  170510                          170510
1396  010704  001254                          TEMP4
1397  010706     000                          .BYTE   0
1398  010707     001                          .BYTE   1
1399  010710  013722  001254                  MOV     TEMP4,(R2)+     ;STORE CSR IN MAP
1400  010714  104403                          INSTR
1401  010716  006553                          VEC
1402  010720  104405                          PARAM
1403  010722  000000                          0
1404  010724  000776                          776
1405  010726  001254                          TEMP4
1406  010730     000                          .BYTE   0
1407  010731     001                          .BYTE   1
1408  010732  013712  001254                  MOV     TEMP4,(R2)      ;STORE VECTOR IN MAP
1409  010736  104402                  10$:    TYPE
1410  010740  006574                          PRIO                    ;ASK WHAT BR LEVEL
1411  010742  004737  012344                  JSR     PC,INTTY        ;GET RESPONSE
1412  010746  022703  000024                  CMP     #24,R3          ;
1413  010752  101014                          BHI     50$             ;BR IF LESS THAN 4
1414  010754  022703  000027                  CMP     #27,R3          ;
1415  010760  103411                          BLO     50$             ;BR IF GREATER THAN 7
1416  010762  012704  000011                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
1417  010766  006303                          ASL     R3              ;SHIFT R3 LEFT
1418  010770  005304                          DEC     R4              ;DEC SHIFT COUNT
1419  010772  001375                          BNE     .-4             ;BR IF NOT DONE
1420  010774  042703  170777                  BIC     #170777,R3      ;BIC UNWANTED BITS
1421  011000  050312                          BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
1422  011002  000403                          BR      8$              ;CONTINUE
1423  011004  104402                  50$:    TYPE
1424  011006  005676                          MQM                     ;RESPONSE IS OUT OF LIMITS
1425  011010  000752                          BR      10$             ;TRY AGAIN
1426  011012                          8$:
1427  011012  000137  011304                  JMP     33$
1428  011016  104402                          TYPE
1429  011020  006633                          CRAM                    ;DOES DMC HAVE CRAM?
1430  011022  004737  012344                  JSR     PC,INTTY        ;GET REPLY
1431  011026  022703  000131                  CMP     #131,R3
1432  011032  001427                          BEQ     9$              ;YES
1433  011034  022703  000116                  CMP     #116,R3         ;NO
1434  011040  001403                          BEQ     40$             ;NOT A Y OR N
1435  011042  104402                          TYPE
```

```
 1436  011044  005676                          MQM                     ;TYPE ''?''
 1437  011046  000761                          BR      8$              ;ASK AGAIN
 1438  011050  104402                  40$:    TYPE
 1439  011052  007376                          SPEED                   ;M8200-YC-AR OR M8200-YC-AL?
 1440  011054  004737  012344                  JSR     PC,INTTY        ;GET RESPONSE
 1441  011060  022703  000122                  CMP     #122,R3         ;IS IT R
 1442  011064  001414                          BEQ     16$             ;BR IF REMOTE
 1443  011066  022703  000114                  CMP     #114,R3         ;IS IT L
 1444  011072  001403                          BEQ     41$             ;BR IF LOCAL
 1445  011074  104402                          TYPE
 1446  011076  005676                          MQM
 1447  011100  000763                          BR      40$             ;TRY AGAIN
 1448  011102  052762  000002  000004  41$:    BIS     #BIT1,4(R2)     ;SET BIT1 IN STAT3
 1449  011110  000402                          BR      16$             ;CONTINUE
 1450  011112  052712  100000          9$:     BIS     #BIT15,(R2)     ;SET BIT 15 IF CRAM
 1451  011116  104402                  16$:    TYPE
 1452  011120  006731                          MODU                    ;ASK WHICH LINE UNIT
 1453  011122  004737  012344                  JSR     PC,INTTY        ;GET REPLY
 1454  011126  022703  000021                  CMP     #21,R3          ;''1''
 1455  011132  001417                          BEQ     30$
 1456  011134  022703  000022                  CMP     #22,R3          ;''2''
 1457  011140  001412                          BEQ     31$
 1458  011142  022703  000116                  CMP     #116,R3         ;''N''
 1459  011146  001403                          BEQ     32$
 1460  011150  104402                          TYPE
 1461  011152  005676                          MQM                     ;IF NOT A 1,2 OR N TYPE ''?''
 1462  011154  000760                          BR      16$             ;TRY AGIAN
 1463  011156  052722  010000          32$:    BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
 1464  011162  022222                          CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
 1465  011164  000447                          BR      33$
 1466  011166  052712  020000          31$:    BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF M8202
 1467  011172  104402                  30$:    TYPE
 1468  011174  007141                          CONN                    ;ASK IF LOOP-BACK IS ON
 1469  011176  004737  012344                  JSR     PC,INTTY        ;GET REPLY
 1470  011202  022703  000131                  CMP     #131,R3         ;Y
 1471  011206  001406                          BEQ     17$
 1472  011210  022703  000116                  CMP     #116,R3         ;N
 1473  011214  001406                          BEQ     18$
 1474  011216  104402                          TYPE
 1475  011220  005676                          MQM                     ;IF NOT Y OR N TYPE ''?''
 1476  011222  000763                          BR      30$             ;TRY AGAIN
 1477  011224  052722  040000          17$:    BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
 1478  011230  000402                          BR      19$
 1479  011232  042722  040000          18$:    BIC     #BIT14,(R2)+    ;NO TURNAROUND
 1480  011236                          19$:
 1481  011236  104403                          INSTR
 1482  011240  007043                          LINE
 1483  011242  104405                          PARAM
 1484  011244  000000                          0
 1485  011246  000377                          377
 1486  011250  001254                          TEMP4
 1487  011252     000                          .BYTE   0
 1488  011253     001                          .BYTE   1
 1489  011254  113722  001254                  MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
 1490  011260  104403                          INSTR
 1491  011262  007101                          BM
```

```
 1492  011264  104405                        PARAM
 1493  011266  000000                        0
 1494  011270  000377                        377
 1495  011272  001254                        TEMP4
 1496  011274     000                        .BYTE   0
 1497  011275     001                        .BYTE   1
 1498  011276  113722  001254                MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
 1499  011302  005722                        TST     (R2)+           ;POP OVER STAT3
 1500  011304                        33$:
 1501  011304  062702  000006                ADD     #6,R2
 1502  011310  005337  001252                DEC     TEMP3           ;DEC DMC COUNT
 1503  011314  001402                        BEQ     34$             ;BR IF DONE
 1504  011316  000137  010656                JMP     12$             ;JUMP IF NOT
 1505  011322  000137  011760        34$:    JMP     13$             ;CONTINUE
 1506  011326  012701  170440        7$:     MOV     #170440,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
 1507  011332  012737  012052  000004        MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
 1508  011340  005011                2$:     CLR     (R1)            ;CLEAR SEL0
 1509  011342  005711                        TST     (R1)            ;IF M8200-YC DMCSR S/B 0
 1510  011344  001173                        BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO M8200-YC
 1511  011346  005061  000006                CLR     6(R1)           ;CLEAR SEL6
 1512  011352  000424                        BR      21$
 1513  011354  005761  000006                TST     6(R1)           ;IF M8200-YC THEN DMRIC S/B =0
 1514  011360  001165                        BNE     3$              ;BR IF NOT M8200-YC
 1515  011362  012711  002000                MOV     #BIT10,(R1)     ;SET ROM0
 1516  011366  005061  000004                CLR     4(R1)           ;CLEAR SEL4
 1517  011372  012761  125252  000006        MOV     #125252,6(R1)   ;WRITE THIS TO SEL6
 1518  011400  052711  020000                BIS     #BIT13,(R1)     ;WRITE IT!
 1519  011404  022761  125252  000004        CMP     #125252,4(R1)   ;WAS IT WRITTEN?
 1520  011412  001004                        BNE     21$             ;IF NO IT IS NOT CRAM
 1521  011414  052762  100000  000002        BIS     #BIT15,2(R2)    ;SET BIT15 IF CRAM
 1522  011422  000431                        BR      22$
 1523  011424  012711  001000        21$:    MOV     #BIT9,(R1)      ;SET ROMI
 1524  011430  012761  100400  000006        MOV     #100400,6(R1)   ;PUT INSTRUCTION IN SEL6
 1525  011436  012711  001400                MOV     #BIT9!BIT8,(R1) ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
 1526  011442  012711  002000                MOV     #BIT10,(R1)     ;SET ROM0
 1527  011446  022761  000456  000006        CMP     #456,6(R1)      ;IS IT LOCAL CROM
 1528  011454  001411                        BEQ     23$             ;BR IF YES
 1529  011456  022761  016520  000006        CMP     #16520,6(R1)    ;IS IT REMOTE CROM?
 1530  011464  001410                        BEQ     22$             ;BR IF YES
 1531  011466  022761  177777  000006        CMP     #-1,6(R1)       ;NO CROM?
 1532  011474  001404                        BEQ     22$             ;BR IF YES
 1533  011476  000516                        BR      3$              ;NOT A DMC
 1534  011500  052762  000002  000006 23$:    BIS     #BIT1,6(R2)     ;SET BIT 1 IN STAT3
 1535                                ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A M8200-YC CSR ADDRESS.
 1536  011506  010122                22$:    MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
 1537  011510  012711  001000        15$:    MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
 1538  011514  005061  000004                CLR     4(R1)           ;CLEAR PORT4
 1539  011520  012761  122113  000006        MOV     #122113,6(R1)   ;LOAD INSTRUCTION (CLR DTR)
 1540  011526  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
 1541  011532  012761  021264  000006        MOV     #021264,6(R1)   ;LOAD INSTRUCTION
 1542  011540  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
 1543  011544  122761  000377  000004        CMPB    #377,4(R1)      ;IS IT ALL ONES?
 1544  011552  001003                        BNE     .+10            ;BR IF NO
 1545  011554  052712  010000                BIS     #BIT12,(R2)     ;IF YES, NO LINE UNIT, SET STATUS BIT
 1546  011560  000436                        BR      20$
 1547  011562  032761  000002  000004        BIT     #BIT1,4(R1)     ;IS SWITCH A ONE?
```

```
1548  011570  001403                        BEQ     .+10            ;BR IF M8201
1549  011572  052712  060000                BIS     #BIT13!BIT14,(R2) ;M8202 ASSUME CONNECTOR
1550  011576  000427                        BR      20$             ;CONNECTOR ON)
1551  011600  032761  000010  000004        BIT     #BIT3,4(R1)     ;IS MRDY SET
1552  011606  001023                        BNE     20$             ;BR IF M8201 NO CONNECTOR (ON LINE)
1553  011610  012761  000100  000004        MOV     #BIT6,4(R1)     ;LOAD PORT4
1554  011616  012761  122113  000006        MOV     #122113,6(R1)   ;LOAD INSTRUCTION
1555  011624  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(SET DTR)
1556  011630  012761  021264  000006        MOV     #021264,6(R1)   ;LOAD INSTRUCTION
1557  011636  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(READ MODEM REG)
1558  011642  032761  000010  000004        BIT     #BIT3,4(R1)     ;IS MRDY SET NOW?
1559  011650  001402                        BEQ     20$             ;BR IF NO CONNECTOR
1560  011652  052712  040000                BIS     #BIT14,(R2)     ;SET STATUS BIT FOR CONNECTOR
1561  011656  005722                20$:    TST     (R2)+           ;POP POINTER
1562  011660  012761  021324  000006        MOV     #021324,6(R1)   ;PUT INSTRUCTION IN PORT6
1563  011666  012711  001400                MOV     #BIT9!BIT8,(R1) ;PORT4_LU 15
1564  011672  156122  000004                BISB    4(R1),(R2)+     ;STORE DDCMP LINE # IN TABLE
1565  011676  012761  021344  000006        MOV     #021344,6(R1)   ;PORT6_INSTRUCTION
1566  011704  012711  001400                MOV     #BIT8!BIT9,(R1) ;CLOCK_INSTR.
1567  011710  156122  000004                BISB    4(R1),(R2)+     ;STORE BM873 ADD IN TABLE
1568  011714  005722                        TST     (R2)+           ;POP OVER STAT3
1569  011716  005011                        CLR     (R1)            ;CLEAR ROMI
1570  011720  005237  001310                INC     DMNUM           ;UPDATE DEVICE COUNTER
1571  011724  022737  000020  001310        CMP     #20,DMNUM       ;ARE MAX. NO. OF DEV FOUND?
1572  011732  001412                        BEQ     13$             ;YES DON'T LOOK FOR ANY MORE.
1573  011734  005011                3$:     CLR     (R1)            ;CLEAR BIT 10
1574  011736  005061  000006                CLR     6(R1)           ;CLEAR SEL 6
1575  011742  062701  000010          14$:  ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
1576  011746  022701  170510                CMP     #170510,R1
1577  011752  001402                        BEQ     13$             ;BR IF DONE
1578  011754  000137  011340                JMP     2$              ;JUMP IF NOT
1579  011760  005037  001306          13$:  CLR     DMACTV
1580  011764  005737  001310                TST     DMNUM           ;WERE ANY M8200-YC'S FOUND AT ALL?
1581  011770  001423                        BEQ     5$              ;ERROR AUTO SIZER FOUND NO M8200-YC'S IN THIS SYS.
1582  011772  013701  001310                MOV     DMNUM,R1
1583  011776  010137  001314                MOV     R1,SAVNUM       ;SAVE NUMBER OF DEVICES
1584  012002  000241                4$:     CLC
1585  012004  006137  001306                ROL     DMACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
1586  012010  005237  001306                INC     DMACTV          ;SET THE BIT
1587  012014  005301                        DEC     R1
1588  012016  001371                        BNE     4$              ;BR IF MORE TO GENERATE
1589  012020  012737  000006  000004        MOV     #6,@#4          ;RESTORE TRAP VECTOR
1590  012026  013737  001306  001312        MOV     DMACTV,SAVACT   ;SAVE ACTIVE REGISTER
1591  012034  000137  012066                JMP     VECMAP          ;GO FIND THE VECTOR NOW.
1592  012040  104402  005771          5$:   TYPE    ,MERR2          ;NOTIFY OPR THAT NO M8200-YC'S FOUND.
1593  012044  005000                        CLR     R0              ;MAKE DATA LIGHTS ZERO
1594  012046  000000                        HALT                    ;STOP THE SHOW
1595  012050  000776                        BR      .-2             ;DISABLE CONT. SW.
1596  012052  012716  011742          6$:   MOV     #14$,(SP)       ;ENTERED BY NON-EXISTANT TIME-OUT.
1597  012056  000002                        RTI                     ;RETURN TO MAINSTREAM
1598
1599  012060  000001                WHICH:  1
1600  012062     002     002                .BYTE   2,2
1601  012064  001256                        TEMP5
1602
1603  012066  032737  000001  001236  VECMAP: BIT     #SW00,STRTSW
```

```
1604  012074  001114                        BNE     5$
1605  012076  012737  000340  000022        MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
1606  012104  012737  012260  000020        MOV     #4$,@#20        ;SET IOT TRAP VECTOR
1607  012112  012702  001500                MOV     #DM.MAP,R2      ;SET SOFTWARE POINTER
1608  012116  012700  000300                MOV     #300,R0         ;FLOATING VECTORS START HERE.
1609  012122  012701  000302                MOV     #302,R1         ;PC OF IOT INSTR.
1610  012126  010120            1$:         MOV     R1,(R0)+        ;START FILLING VECTOR AREA
1611  012130  012721  000004                MOV     #4,(R1)+        ;WITH .+2; IOT
1612  012134  022021                        CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
1613  012136  020127  001000                CMP     R1,#1000
1614  012142  101771                        BLOS    1$              ;BR IF MORE TO FILL
1615  012144  013737  001306  001246        MOV     DMACTV,TEMP1    ;STORE TEMPORALLY
1616  012152  006037  001246            2$: ROR     TEMP1           ;BRING OUT A BIT
1617  012156  103063                        BCC     5$              ;BR IF ALL DONE
1618  012160  012704  000012                MOV     #12,R4          ;R4 IS INDEX REGISTER
1619  012164  016437  012330  177776        MOV     BRLVL(R4),PS    ;SET PS TO 7
1620  012172  011201                        MOV     (R2),R1
1621  012174  012761  000200  000004        MOV     #200,4(R1)
1622  012202  012711  001000                MOV     #BIT9,(R1)      ;SET ROMI
1623  012206  012761  121111  000006        MOV     #121111,6(R1)   ;PUT INSTRUCTION IN PORT6
1624  012214  012711  001400                MOV     #BIT9!BIT8,(R1) ;FORCE AN INTERRUPT
1625  012220  105200            7$:         INCB    R0              ;STALL
1626  012222  001376                        BNE     .-2             ;FOR TIME TO INTERUPT
1627  012224  162704  000002                SUB     #2,R4           ;GET NEXT LOWEST PS LEVEL
1628  012230  001404                        BEQ     6$              ;BR IF R4 = 0
1629  012232  016437  012330  177776        MOV     BRLVL(R4),PS    ;MOVE NEXT LOWER LEVEL IN PS
1630  012240  000767                        BR      7$              ;BR TO DELAY
1631  012242  052762  005300  000002  6$:   BIS     #5300,2(R2)     ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX M8200-YC LATE
1632  012250  005011            3$:         CLR     (R1)            ;CLEAR ROMI
1633  012252  062702  000010                ADD     #10,R2          ;POP SOFTWARE POINTER
1634  012256  000735                        BR      2$              ;KEEP GOING
1635  012260  051662  000002            4$: BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
1636  012264  042762  000007  000002        BIC     #7,2(R2)        ;CLEAR JUNK
1637  012272  016405  012332                MOV     BRLVL+2(R4),R5  ;GET BR LEVEL OF M8200-YC
1638  012276  006305                        ASL     R5              ;SHIFT LEVEL 4 PLACES
1639  012300  006305                        ASL     R5              ;TO THE LEFT FOR THE
1640  012302  006305                        ASL     R5              ;STATUS TABLE
1641  012304  006305                        ASL     R5
1642  012306  042705  170777                BIC     #170777,R5      ;CLEAR UNWANTED BITS
1643  012312  050562  000002                BIS     R5,2(R2)        ;PUT BR LEVEL IN STATUS TABLE
1644  012316  022626                        CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
1645  012320  012716  012250                MOV     #3$,(SP)        ;SET FOR RETURN
1646  012324  000002                        RTI
1647  012326  000207            5$:         RTS     PC              ;ALL DONE WITH "AUTO SIZING"
1648
1649  012330  000000            BRLVL:      0       ;LEVEL 0
1650  012332  000000                        0       ;LEVEL 0
1651  012334  000200                        200     ;LEVEL 4
1652  012336  000240                        240     ;LEVEL 5
1653  012340  000300                        300     ;LEVEL 6
1654  012342  000340                        340     ;LEVEL 7
1655
1656
1657  012344  105777  166634        INTTY:  TSTB    @TKCSR          ;WAIT FOR DONE
1658  012350  100375                        BPL     .-4
1659  012352  017703  166630                MOV     @TKDBR,R3       ;PUT CHAR IN R3
```

```
1660  012356  105777  166626          TSTB    @TPCSR          ;WAIT UNTIL PRINTER IS READY
1661  012362  100375                  BPL     .-4
1662  012364  010377  166622          MOV     R3,@TPDBR       ;ECHO CHAR
1663  012370  042703  000240          BIC     #BIT7!BIT5,R3   ;MASK OFF LOWER CASE
1664  012374  000207                  RTS     PC              ;RETURN
1665
1666
1667  012376  000000          ROMMAP: 0                       ;POINTER TO V5 OR V4 MICRO-CODE
1668
1669  012400                  V4MAP:                          ;VERSION 4 MICRO-CODE
1670
1671                          ;THE MICRO-CODE IMAGE RESIDES HERE - ONLY OCTAL NUMBERS
1672
```

```
1673                                    .TITLE  SLAVE.MAC1
1674                                    .IDENT  /4.01/
1675                                    ;
1676                                    ; LPA11-K MICRO CODE
1677                                    ;
1678                                    ; CHARLES A. SAMUELSON
1679                                    ; NOVEMBER, 1977
1680                                    ;
1681  016402                            V5MAP:
1682  016402  000456  023240  060360            .WORD     456, 23240, 60360,101407, 22351, 22250,100406,   405
1683  016410  101407  022351  022250
1684  016416  100406  000405
1685  016422  062231  033343  022740            .WORD    62231, 33343, 22740, 73163,101012, 23365, 23366,   414
1686  016430  073163  101012  023365
1687  016436  023366  000414
1688  016442  061231  023360  023654            .WORD    61231, 23360, 23654,102441, 23210,   417, 60670, 73224
1689  016450  102441  023210  000417
1690  016456  060670  073224
1691  016462  053221  060610  103655            .WORD    53221, 60610,103655,103151, 61620,103253, 60530,113723
1692  016470  103151  061620  103253
1693  016476  060530  113723
1694  016502  100522  061620  103053            .WORD   100522, 61620,103053, 23230,   417, 60670, 73224, 53221
1695  016510  023230  000417  060670
1696  016516  073224  053221
1697  016522  060610  103742  100614            .WORD    60610,103742,100614,102456, 20660, 63305, 60534,107663
1698  016530  102456  020660  063305
1699  016536  060534  107663
1700  016542  103063  020700  063306            .WORD   103063, 20700, 63306, 60605,107024,103731,106204,106601
1701  016550  060605  107024  103731
1702  016556  106204  106601
1703  016562  061620  117121  116674            .WORD    61620,117121,116674, 60566,101422, 10374, 60606,  2776
1704  016570  060566  101422  010374
1705  016576  060606  002776
1706  016602  116113  002775  116515            .WORD   116113,  2775,116515,  2757,117267,  2577,117514, 61620
1707  016610  002757  117267  002577
1708  016616  117514  061620
1709  016622  002773  116671  002737            .WORD    2773,116671,  2737,117266, 60526,  2767,117116,  2677
1710  016630  117266  060526  002767
1711  016636  117116  002677
1712  016642  117670  100422  023340            .WORD   117670,100422, 23340, 63124,   410, 70404, 20640,102526
1713  016650  063124  000410  070404
1714  016656  020640  102526
1715  016662  022204  036500  020640            .WORD    22204, 36500, 20640,102532, 22205, 20500,102145,   415
1716  016670  102532  022205  020500
1717  016676  102145  000415
1718  016702  061230  120600  102141            .WORD    61230,120600,102141,120620,102146, 23360, 22520, 62210
1719  016710  120620  102146  023360
1720  016716  022520  062210
1721  016722  100417  010012  102155            .WORD   100417, 10012,102155, 10010, 23347,   421, 63220, 42226
1722  016730  010010  023347  000421
1723  016736  063220  042226
1724  016742  057230  042227  000500            .WORD    57230, 42227,   500, 62222, 22363, 61210, 20640,102566
1725  016750  062222  022363  061210
1726  016756  020640  102566
1727  016762  022202  020640  102571            .WORD    22202, 20640,102571, 22203, 63070, 60470, 62226, 61210
1728  016770  022203  063070  060470
```

```
1729  016776  062226  061210
1730  017002  120600  102200  060570          .WORD   120600,102200, 60570, 62226, 20640,102604, 22202, 20640
1731  017010  062226  020640  102604
1732  017016  022202  020640
1733  017022  102607  022203  061210          .WORD   102607, 22203, 61210,100421,   500, 60704, 62230, 70201
1734  017030  100421  000500  060704
1735  017036  062230  070201
1736  017042  040620  061620  102646          .WORD    40620, 61620,102646,100643, 20620, 23357, 20640,102632
1737  017050  100643  020620  023357
1738  017056  020640  102632
1739  017062  020600  100625  063177          .WORD    20600,100625, 63177,101226, 20640, 61620,103234, 20620
1740  017070  101226  020640  061620
1741  017076  103234  020620
1742  017102  020640  103240  022351          .WORD    20640,103240, 22351, 70204, 22740,100421,   423, 70401
1743  017110  070204  022740  100421
1744  017116  000423  070401
1745  017122  050220  022740  100643          .WORD    50220, 22740,100643, 23374,100657,   401, 63234,   423
1746  017130  023374  100657  000401
1747  017136  063234  000423
1748  017142  070401  050220  057230          .WORD    70401, 50220, 57230,101702, 57231, 43232, 20640,102666
1749  017150  101702  057231  043232
1750  017156  020640  102666
1751  017162  062472  040371  101700          .WORD    62472, 40371,101700, 70212, 22600, 63174,101257,100421
1752  017170  070212  022600  063174
1753  017176  101257  100421
1754  017202  062610  100673  020640          .WORD    62610,100673, 20640,102702, 20600, 63174,101302,100421
1755  017210  102702  020600  063174
1756  017216  101302  100421
1757  017222  063237  000423  070401          .WORD    63237,   423, 70401, 50220, 57232, 57233, 54620, 43234
1758  017230  050220  057232  057233
1759  017236  054620  043234
1760  017242  060374  165617  062474          .WORD    60374,165617, 62474, 40373,101727, 70214,164477, 62612
1761  017250  040373  101727  070214
1762  017256  164477  062612
1763  017262  100725  123150  123160          .WORD   100725,123150,123160, 60470, 63100, 61226, 61207,   577
1764  017270  060470  063100  061226
1765  017276  061207  000577
1766  017302  063265  100421  063530          .WORD    63265,100421, 63530,103763, 23357, 20640,103345, 61620
1767  017310  103763  023357  020640
1768  017316  103345  061620
1769  017322  103345  022231  063177          .WORD   103345, 22231, 63177,101345, 20640,102354,102754, 22210
1770  017330  101345  020640  102354
1771  017336  102754  022210
1772  017342  063177  101354  100421          .WORD    63177,101354,100421, 63220,   436, 63270,   410, 70410
1773  017350  063220  000436  063270
1774  017356  000410  070410
1775  017362  060520  107407  056224          .WORD    60520,107407, 56224, 42225,   415, 61230,120600,102376
1776  017370  042225  000415  061230
1777  017376  120600  102376
1778  017402  020640  107000  022011          .WORD    20640,107000, 22011, 20640,107003, 22031,100421, 56226
1779  017410  020640  107003  022031
1780  017416  100421  056226
1781  017422  042227  020640  061620          .WORD    42227, 20640, 61620,107011, 22222, 20640, 61620,107015
1782  017430  107011  022222  020640
1783  017436  061620  107015
1784  017442  022223  000421  061230          .WORD    22223,   421, 61230,100421,   757, 63265, 60603,105433
```

```
1785  017450  100421  000757  063265
1786  017456  060603  105433
1787  017462  063467  105661  100421          .WORD    63467,105661,100421,   407, 73223, 63224, 53221,105646
1788  017470  000407  073223  063224
1789  017476  053221  105646
1790  017502  057630  106246  043231          .WORD    57630,106246, 43231, 76571,105246, 43231, 76571,105246
1791  017510  076571  105246  043231
1792  017516  076571  105246
1793  017522  043631  061620  116747          .WORD    43631, 61620,116747, 37373, 57220, 43232, 76572,105103
1794  017530  037373  057220  043232
1795  017536  076572  105103
1796  017542  043232  076572  105103          .WORD    43232, 76572,105103, 55220, 77173, 55222, 55223, 55224
1797  017550  055220  077173  055222
1798  017556  055223  055224
1799  017562  041225  060611  107254          .WORD    41225, 60611,107254, 70461,136500,136520, 74620,136400
1800  017570  070461  136500  136520
1801  017576  074620  136400
1802  017602  136440  122460  104506          .WORD   136440,122460,104506, 70461, 36760, 22760, 60611,106126
1803  017610  070461  036760  022760
1804  017616  060611  106126
1805  017622  106516  000513  100710          .WORD   106516,   513,100710,104665, 43220,104526, 63073,105126
1806  017630  104665  043220  104526
1807  017636  063073  105126
1808  017642  000410  070401  043233          .WORD      410, 70401, 43233,   404, 70401, 42413, 60610,107550
1809  017650  000404  070401  042413
1810  017656  060610  107550
1811  017662  117367  010016  000477          .WORD   117367, 10016,   477, 60360,105136, 10014, 54620,106274
1812  017670  060360  105136  010014
1813  017676  054620  106274
1814  017702  062226  042227  062203          .WORD    62226, 42227, 62203,   501, 62222,   421, 61230,100421
1815  017710  000501  062222  000421
1816  017716  061230  100421
1817  017722  000420  060704  062230          .WORD      420, 60704, 62230,   555,100710,104672, 42222,   561
1818  017730  000555  100710  104672
1819  017736  042222  000561
1820  017742  100710  104657  042223          .WORD   100710,104657, 42223, 60610,117367,   403, 60360,105274
1821  017750  060610  117367  000403
1822  017756  060360  105274
1823  017762  010020  063120  054400          .WORD    10020, 63120, 54400,106274, 62226, 42227,   421, 61230
1824  017770  106274  062226  042227
1825  017776  000421  061230
1826  020002  104646  010016  000775          .WORD   104646, 10016,   775,104606, 10014,   776, 63265, 57220
1827  020010  104606  010014  000776
1828  020016  063265  057220
1829  020022  063060  063060  062204          .WORD    63060, 63060, 62204, 42225,   415, 61230, 70603, 63224
1830  020030  042225  000415  061230
1831  020036  070603  063224
1832  020042  053221  105646  057230          .WORD    53221,105646, 57230, 74620, 74620, 43231, 20640,107670
1833  020050  074620  074620  043231
1834  020056  020640  107670
1835  020062  060604  062230  020640          .WORD    60604, 62230, 20640,106232, 22010, 60610,106676, 60620
1836  020070  106232  022010  060610
1837  020076  106676  060620
1838  020102  022030  060611  107707          .WORD    22030, 60611,107707, 60611, 61620,116542, 73563,105035
1839  020110  060611  061620  116542
1840  020116  073563  105035
```

```
1841  020122  060607  101421  063167          .WORD    60607,101421, 63167,104433, 70201, 76470,104474,   602
1842  020130  104433  070201  076470
1843  020136  104474  000602
1844  020142  110706  000601  110706          .WORD   110706,   601,110706,   600,110706,   440, 21365,110706
1845  020150  000600  110706  000440
1846  020156  021365  110706
1847  020162  000420  104666  000460          .WORD     420,104666,   460,104666,   500,104666, 70201, 43233
1848  020170  104666  000500  104666
1849  020176  070201  043233
1850  020202  000775  062673  023033          .WORD     775, 62673, 23033,   600, 60713, 62230,104641,   440
1851  020210  000600  060713  062230
1852  020216  104641  000440
1853  020222  060704  062230  020640          .WORD   60704, 62230, 20640,106312,122150, 20640,106315,122170
1854  020230  106312  122150  020640
1855  020236  106315  122170
1856  020242  104643  043231  000724          .WORD   104643, 43231,   724,100710,114743, 42222,   730,100710
1857  020250  100710  114743  042222
1858  020256  000730  100710
1859  020262  104657  114731  063237          .WORD   104657,114731, 63237,   404, 60360,115261,   422, 60400
1860  020270  000404  060360  115261
1861  020276  000422  060400
1862  020302  070400  057633  116261          .WORD    70400, 57633,116261, 43232,   700, 60712, 62227, 62225
1863  020310  043232  000700  060712
1864  020316  062227  062225
1865  020322  060530  113411  111000          .WORD    60530,113411,111000,   402, 60413, 62224,   415, 61230
1866  020330  000402  060413  062224
1867  020336  000415  061230
1868  020342  060612  107766  060611          .WORD    60612,107766, 60611, 61620,106766,174477,120600,106366
1869  020350  061620  106766  174477
1870  020356  120600  106366
1871  020362  022002  022023  022106          .WORD    22002, 22023, 22106,   421, 61230, 60612,113410,110417
1872  020370  000421  061230  060612
1873  020376  113410  110417
1874  020402  000404  060413  062226          .WORD     404, 60413, 62226,   421, 61230, 60611, 61620,112424
1875  020410  000421  061230  060611
1876  020416  061620  112424
1877  020422  174477  060611  113017          .WORD   174477, 60611,113017,111021, 60612,113417,174477,   577
1878  020430  111021  060612  113417
1879  020436  174477  000577
1880  020442  110426  060532  113424          .WORD   110426, 60532,113424,174477,   677, 63073, 62672,   500
1881  020450  174477  000677  063073
1882  020456  062672  000500
1883  020462  062222  062223  060613          .WORD    62222, 62223, 60613, 62226,   621, 61230,174477, 10015
1884  020470  062226  000621  061230
1885  020476  174477  010015
1886  020502  042225  056227  037372          .WORD    42225, 56227, 37372, 14421, 61223, 43235, 10100,   407
1887  020510  014421  061223  043235
1888  020516  010100  000407
1889  020522  063236  023357  063177          .WORD    63236, 23357, 63177,111052, 20660, 20640, 61620,112455
1890  020530  111052  020660  020640
1891  020536  061620  112455
1892  020542  020660  060610  113674          .WORD    20660, 60610,113674, 60531,113150,110534, 57000, 63172
1893  020550  060531  113150  110534
1894  020556  057000  063172
1895  020562  111073  123012  123040          .WORD   111073,123012,123040, 22030, 56226, 56222, 62203, 55230
1896  020570  022030  056226  056222
```

```
1897  020576  062203  055230
1898  020602  056224  023655  061620          .WORD    56224, 23655, 61620,113122, 10014, 22362, 56226,135070
1899  020610  113122  010014  022362
1900  020616  056226  135070
1901  020622  040620  102224  120600          .WORD    40620,102224,120600,112112, 42226,121070, 20640, 61620
1902  020630  112112  042226  121070
1903  020636  020640  061620
1904  020642  113116  100624  112501          .WORD   113116,100624,112501, 55230, 60535,111145,107663, 20660
1905  020650  055230  060535  111145
1906  020656  107663  020660
1907  020662  056222  055230  022010          .WORD    56222, 55230, 22010, 50220, 60611,112073,112466, 20640
1908  020670  050220  060611  112073
1909  020676  112466  020640
1910  020702  112543  023200  110473          .WORD   112543, 23200,110473,   440,110546,   420, 62231,110504
1911  020710  000440  110546  000420
1912  020716  062231  110504
1913  020722  123012  056222  056226          .WORD   123012, 56222, 56226, 60611,112161,112561, 20640,112556
1914  020730  060611  112161  112561
1915  020736  020640  112556
1916  020742  023200  062203  055230          .WORD    23200, 62203, 55230,120600,112163, 56226, 56222, 55230
1917  020750  120600  112163  056226
1918  020756  056222  055230
1919  020762  110627  062203  056224          .WORD   110627, 62203, 56224, 23655, 61620,113177,110504,112573
1920  020770  023655  061620  113177
1921  020776  110504  112573
1922  021002  055230  060535  111145          .WORD    55230, 60535,111145,107663, 60620, 60620, 22010, 20420
1923  021010  107663  060620  060620
1924  021016  022010  020420
1925  021022  056224  055230  062230          .WORD    56224, 55230, 62230, 20660, 60620, 60620, 56226, 22010
1926  021030  020660  060620  060620
1927  021036  056226  022010
1928  021042  055230  120600  112221          .WORD    55230,120600,112221, 22030, 56226, 55230, 50220, 60611
1929  021050  022030  056226  055230
1930  021056  050220  060611
1931  021062  112172  112636  020640          .WORD   112172,112636, 20640,112543, 22203,110572, 57000, 63172
1932  021070  112543  022203  110572
1933  021076  057000  063172
1934  021102  111171  123012  123040          .WORD   111171,123012,123040,110571, 43000, 63172,111251,123012
1935  021110  110571  043000  063172
1936  021116  111251  123012
1937  021122  123040  060520  060376          .WORD   123040, 60520, 60376,111256,   500,110546, 60415, 62226
1938  021130  111256  000500  110546
1939  021136  060415  062226
1940  021142  020640  112703  061620          .WORD    20640,112703, 61620,113265,100624,112660, 22202, 20640
1941  021150  113265  100624  112660
1942  021156  022202  020640
1943  021162  112667  022203  121070          .WORD   112667, 22203,121070, 20660, 60611,112251,112644, 20640
1944  021170  020660  060611  112251
1945  021176  112644  020640
1946  021202  112543  023200  110651          .WORD   112543, 23200,110651,   460, 62231,110516,113710, 60704
1947  021210  000460  062231  110516
1948  021216  113710  060704
1949  021222  061220  020640  113311          .WORD    61220, 20640,113311,122011,120400,103421,120520,102021
1950  021230  122011  120400  103421
1951  021236  120520  102021
1952  021242  116707  117142  104646          .WORD   116707,117142,104646, 21345, 20640,112724, 23211, 20640
```

```
1953  021250  021345  020640  112724
1954  021256  023211  020640
1955  021262  112727  023210  060530          .WORD   112727, 23210, 60530,113345, 10100, 20640,112735, 23212
1956  021270  113345  010100  020640
1957  021276  112735  023212
1958  021302  020640  112740  036600          .WORD    20640,112740, 36600, 63172,111340, 70604, 63233,   434
1959  021310  063172  111340  070604
1960  021316  063233  000434
1961  021322  063221  000424  063173          .WORD    63221,   424, 63173,111756, 63001,110752, 72601, 14415
1962  021330  111756  063001  110752
1963  021336  072601  014415
1964  021342  063232  000401  063233          .WORD    63232,   401, 63233, 20640,112763, 36600, 63173,111373
1965  021350  020640  112763  036600
1966  021356  063173  111373
1967  021362  076611  000407  070401          .WORD    76611,   407, 70401, 63172,111363,   407, 70401, 57220
1968  021370  063172  111363  000407
1969  021376  070401  057220
1970  021402  075202  055220  043233          .WORD    75202, 55220, 43233,   404, 70401, 76600,136400, 62613
1971  021410  000404  070401  076600
1972  021416  136400  062613
1973  021422  060530  117013  110437          .WORD    60530,117013,110437,115015,114427,   402, 63232, 20640
1974  021430  115015  114427  000402
1975  021436  063232  020640
1976  021442  117511  000420  060704          .WORD   117511,   420, 60704, 62230, 63172,115021,114432, 60611
1977  021450  062230  063172  115021
1978  021456  114432  060611
1979  021462  116062  116462  000404          .WORD   116062,116462,   404, 63310,   674, 73232,   403, 63234
1980  021470  063310  000674  073232
1981  021476  000403  063234
1982  021502  040620  115450  000420          .WORD    40620,115450,   420, 73012, 63174,115040,   400,110706
1983  021510  073012  063174  115040
1984  021516  000400  110706
1985  021522  000423  070401  072612          .WORD      423, 70401, 72612,   404, 62412, 57234,   420, 76412
1986  021530  000404  062412  057234
1987  021536  000420  076412
1988  021542  076614  062614  060611          .WORD    76614, 62614, 60611,117072, 60531,117504, 70201,   635
1989  021550  117072  060531  117504
1990  021556  070201  000635
1991  021562  062670  100421  063070          .WORD    62670,100421, 63070,   415, 70401, 43220,   500, 63310
1992  021570  000415  070401  043220
1993  021576  000500  063310
1994  021602  000502  104732  114507          .WORD      502,104732,114507,114464, 23371,   416,114474,   500
1995  021610  114464  023371  000416
1996  021616  114474  000500
1997  021622  110706  000420  110706          .WORD   110706,   420,110706, 63060, 63060, 63060, 63060, 43266
1998  021630  063060  063060  063060
1999  021636  063060  043266
2000  021642  114523  000737  063265          .WORD   114523,   737, 63265, 23376,   407, 73224, 53221,115542
2001  021650  023376  000407  073224
2002  021656  053221  115542
2003  021662  057230  074620  074620          .WORD    57230, 74620, 74620, 57631,117155, 60531,117572, 60611
2004  021670  057631  117155  060531
2005  021676  117572  060611
2006  021702  061620  116611  073164          .WORD    61620,116611, 73164,115126, 63176,101021, 63120,   423
2007  021710  115126  063176  101021
2008  021716  063120  000423
```

```
2009  021722  070400  000600  043234          .WORD    70400,   600, 43234, 62714,100421,   415, 70401, 54360
2010  021730  062714  100421  000415
2011  021736  070401  054360
2012  021742  115562  114535  014564          .WORD   115562,114535, 14564,114626,114535, 70201,   776, 43234
2013  021750  114626  114535  070201
2014  021756  000776  043234
2015  021762  062674  114535  000416          .WORD    62674,114535,   416, 70401, 40360,115577,114537,   421
2016  021770  070401  040360  115577
2017  021776  114537  000421
2018  022002  070401  000603  114626          .WORD    70401,   603,114626,114537, 70201, 43230,   402, 62710
2019  022010  114537  070201  043230
2020  022016  000402  062710
2021  022022  114537  060530  117142          .WORD   114537, 60530,117142,   404, 70401, 40360,115620,114542
2022  022030  000404  070401  040360
2023  022036  115620  114542
2024  022042  000622  114626  060620          .WORD      622,114626, 60620, 20640,117662,104630, 61223, 57234
2025  022050  020640  117662  104630
2026  022056  061223  057234
2027  022062  043235  060576  115246          .WORD    43235, 60576,115246, 63076, 60610, 61220, 60611, 61222
2028  022070  063076  060610  061220
2029  022076  060611  061222
2030  022102  000424  063230  063231          .WORD      424, 63230, 63231,   645,104732, 60620,123077,123010
2031  022110  000645  104732  060620
2032  022116  123077  123010
2033  022122  123051  020400  063274          .WORD   123051, 20400, 63274, 63174,111010, 20420, 63275, 63175
2034  022130  063174  111010  020420
2035  022136  063275  063175
2036  022142  111010  174617  000404          .WORD   111010,174617,   404, 61225,   420,110706, 63060, 63060
2037  022150  061225  000420  110706
2038  022156  063060  063060
2039  022162  063060  063060  043266          .WORD    63060, 63060, 43266,114676,   773, 63265, 23376,   407
2040  022170  114676  000773  063265
2041  022176  023376  000407
2042  022202  073224  053221  115707          .WORD    73224, 53221,115707, 57230, 74530,117307,115320, 73164
2043  022210  057230  074530  117307
2044  022216  115320  073164
2045  022222  115301  063176  101021          .WORD   115301, 63176,101021, 63120,   423, 70400,   500,114552
2046  022230  063120  000423  070400
2047  022236  000500  114552
2048  022242  074610  117323  114707          .WORD    74610,117323,114707, 60576,115337,   420, 60704, 62230
2049  022250  060576  115337  000420
2050  022256  060704  062230
2051  022262  104721  042223  000734          .WORD   104721, 42223,   734,104732, 60620, 63076,114707,   401
2052  022270  104732  060620  063076
2053  022276  114707  000401
2054  022302  061225  000560  110706          .WORD    61225,   560,110706,   402, 61225,   460,110706, 60530
2055  022310  000402  061225  000460
2056  022316  110706  060530
2057  022322  117352  104646  000767          .WORD   117352,104646,   767, 77670, 43220, 70201, 62620,117361
2058  022330  077670  043220  070201
2059  022336  062620  117361
2060  022342  104646  000500  063310          .WORD   104646,   500, 63310,   765,104732,104674,104646,   771
2061  022350  000765  104732  104674
2062  022356  104646  000771
2063  022362  104732  104674  060610          .WORD   104732,104674, 60610,107646,104626,     0,     0,     0
2064  022370  107646  104626  000000
```

```
 2065  022376  000000  000000
 2066
 2067  022402  177777                          .WORD   -1
 2068
```

```
                                        ;*************************** TEST 1 ***************************
                                        ;*THIS IS A SPECIAL TEST WHICH WILL RUN ON A KMC (DMC WITH
                                        ;*WRITTABLE CONTROL STORE) TO LOAD THE CRAM WITH THE DDCMP
                                        ;*MICRO-CODE. FIRST BE SURE BIT1 OF STAT3 IS SET UP AS FOLLOWS
                                        ;*1=LOCAL HIGH SPEED CODE,  0=REMOTE LOW SPEED CODE  THE STATUS
                                        ;*OF STAT3 BIT1 DETERMINES WHICH MICRO-CODE WILL
                                        ;*BE LOADED IN THE KMC. LOOP ON THIS TEST FOR A FEW SECONDS
                                        ;*TO LOAD THE KMC.
                                        ;:*************************************************************

                                        ;  TEST 1
                                        ;--------------
022404  012737  000001  001226  TST1:   MOV     #1,TSTNO
022412  012737  022476  001216          MOV     #TST2,NEXT
                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
022420  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
022424  032737  100000  001366          BIT     #BIT15,STAT1    ;BE SURE DMC HAS CRAM
022432  001420                          BEQ     2$              ;SKIP IF NO CRAM
022434  005000                          CLR     R0              ;R0=CRAM ADDRESS
022436  013702  012376                  MOV     ROMMAP,R2       ;R2 POINTS TO ROMMAP
022442  012711  002000          1$:     MOV     #BIT10,(R1)     ;SET ROM0
022446  010061  000004                  MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
022452  012261  000006                  MOV     (R2)+,6(R1)     ;LOAD WORD TO BE WRITTEN
022456  052711  020000                  BIS     #BIT13,(R1)     ;WRITE IT!
022462  005200                          INC     R0              ;NEXT ADDRESS
022464  022700  002000                  CMP     #2000,R0        ;DONE YET?
022470  001364                          BNE     1$              ;BR IF NO
022472  005011                          CLR     (R1)            ;CLEAR SEL0
022474  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*************************** TEST 2 ***************************
                                        ;*TEST OF BR RIGHT SHIFT
                                        ;*VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
                                        ;*SHIFTS THE RESULTING BR DATA RIGHT ONCE.
                                        ;:*************************************************************

                                        ;  TEST 2
                                        ;--------------
022476  012737  000002  001226  TST2:   MOV     #2,TSTNO
022504  012737  022610  001216          MOV     #TST3,NEXT
                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
022512  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
022514  013701  001404                  MOV     DMCSR,R1        ;R1 = DMC BASE ADDRESS
022520  005011                          CLR     (R1)            ;CLEAR SEL0
022522  012705  052525                  MOV     #52525,R5       ;START WITH 125
022526  010561  000004                  MOV     R5,4(R1)        ;PORT4 125
022532  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022534  120500                          120500                  ;BR   PORT4
022536  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022540  061620                          061620                  ;BR RSH BR, SHIFT BR RIGHT
022542  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022544  061225                          061225                  ;PORT5_BR
```

```
2125  022546  006005                          ROR     R5              ;R5 = "EXPECTED"
2126  022550  116104  000005                  MOVB    5(R1),R4        ;R4 = "FOUND"
2127  022554  120504                          CMPB    R5,R4           ;DID BR SHIFT RIGHT ONCE?
2128  022556  001401                          BEQ     1$              ;BR IF YES
2129  022560  104012                          HLT     12              ;BR RIGHT SHIFT ERROR
2130  022562                          1$:
2131  022562  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2132  022564  061620                          061620                  ;BR RSH BR, SHFT BR RIGHT AGAIN
2133  022566  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2134  022570  061225                          061225                  ;PORT5 BR
2135  022572  006005                          ROR     R5              ;R5 = "EXPECTED"
2136  022574  116104  000005                  MOVB    5(R1),R4        ;R4 = "FOUND"
2137  022600  120504                          CMPB    R5,R4           ;DID BR SHIFT RIGHT?
2138  022602  001401                          BEQ     2$              ;BR IF YES
2139  022604  104012                          HLT     12              ;BR RIGHT SHIFT ERROR
2140  022606  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
2141
2142
2143                                          ;***************************** TEST 3 ****************************
2144                                          ;*CROM READ TEST
2145                                          ;*THIS TEST READS EACH ROM LOCATION AND COMPARES
2146                                          ;*IT TO A SOFTWARE DUPLICATE OF THE CROM. THIS TEST
2147                                          ;*ALSO TESTS THE JUMP(I) MICRO-PROCESSOR INSTRUCTION.
2148                                          ;*IF THIS TEST FAILS CHECK YOUR CROM PART NUMBERS.
2149                                          ;*CRLPM-B SUPPORTS THE FOLLOWING PART NUMBERS:
2150                                          ;*
2151                                          ;*M8200-YC-AR  (M8200-YA)
2152                                          ;:*************************************************************************
2153
2154                                          ;  TEST 3
2155                                          ;--------------
2156  022610  012737  000003  001226  TST3:   MOV     #3,TSTNO
2157  022616  012737  023004  001216          MOV     #TST4,NEXT
2158  022624  012737  022662  001220          MOV     #1$,LOCK
2159                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2160  022632  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2161  022634  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT RAM OR ROM
2162  022642  001057                          BNE     4$              ;SKIP TEST IF CRAM
2163  022644  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2164  022650  005011                          CLR     (R1)            ;CLEAR RUN
2165  022652  013700  012376                  MOV     ROMMAP,R0       ;R0 POINTS TO SOFTWARE ROM MAP
2166  022656  005002                          CLR     R2              ;R2 CONTAINS ROM ADDRESS BITS 0-7
2167  022660  005003                          CLR     R3              ;R3 CONTAINS ROM ADDRESS BITS 8&9 IN BITS 11&12
2168  022662  042737  014377  022702  1$:     BIC     #14377,2$       ;CLEAR ADDRESS FIELDS OF INSTRUCTION
2169  022670  050237  022702                  BIS     R2,2$           ;ADD BITS 0-7 TO INSTRUCTION
2170  022674  050337  022702                  BIS     R3,2$           ;ADD BITS 11&12 TO INSTRUCTION
2171  022700  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2172  022702  100400                  2$:     100400                  ;JUMP(I) TO ROM ADDRESS IN R2 & R3
2173  022704  012711  002000                  MOV     #BIT10,(R1)     ;SET ROM0
2174  022710  011005                          MOV     (R0),R5         ;PUT "EXPECTED" IN R5
2175  022712  016104  000006                  MOV     6(R1),R4        ;PUT "FOUND" IN R4
2176  022716  020504                          CMP     R5,R4           ;COMPARE ROM CONTENTS TO SOFT DUP
2177  022720  001414                          BEQ     3$              ;BR IF OK
2178  022722  010337  001252                  MOV     R3,TEMP3        ;PUT ROM ADDRESS IN TEMP3
2179  022726  000241                          CLC                     ;FOR ERROR TYPEOUT
2180  022730  006037  001252                  ROR     TEMP3
```

```
2181  022734  006037  001252                ROR     TEMP3
2182  022740  006037  001252                ROR     TEMP3
2183  022744  050237  001252                BIS     R2,TEMP3        ;TEMP3 NOW CONTAINS CORRECT ADDRESS
2184  022750  104004                        HLT     4               ;ROM READ ERROR
2185  022752  104401              3$:       SCOP1                   ;LOOP TO 1$ IF SW09=1
2186  022754  005720                        TST     (R0)+           ;BUMP SOFT POINTER
2187  022756  005202                        INC     R2              ;BUMP ROM ADDRESS
2188  022760  022702  000400                CMP     #400,R2         ;IS R2 TO MAX YET?
2189  022764  001336                        BNE     1$              ;BR IF NO
2190  022766  005002                        CLR     R2              ;YES, RESET R2 TO 0
2191  022770  062703  004000                ADD     #4000,R3        ;INC  TO NEXT PAGE OF ROM
2192  022774  022703  020000                CMP     #20000,R3       ;DONE YET?
2193  023000  001330                        BNE     1$              ;BR IF NO
2194  023002  104400              4$:       SCOPE                   ;SCOPE THIS TEST
2195
2196
2197                                        ;*************************** TEST 4 ***************************
2198                                        ;*CROM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
2199                                        ;*PERFORM THE JUMP INSTRUCTION
2200                                        ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2201                                        ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2202                                        ;;*************************************************************
2203
2204                                        ;  TEST 4
2205                                        ;--------------
2206  023004  012737  000004  001226  TST4: MOV     #4,TSTNO
2207  023012  012737  023200  001216        MOV     #TST5,NEXT
2208  023020  012737  023044  001220        MOV     #1$,LOCK
2209                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
2210  023026  104412                        MSTCLR                  ;MASTER CLEAR M8200-YC
2211  023030  032737  100000  001366        BIT     #BIT15,STAT1    ;IS IT CRAM?
2212  023036  001057                        BNE     6$+2            ;SKIP TEST IF YES
2213  023040  004737  026434                JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2214  023044                        1$:
2215  023044  004737  026300                JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2216  023050  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2217  023052  100400                        100400                  ;START AT ROM PC=0
2218  023054  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2219  023056  114377                        114377!<400*0>          ;JUMP TO ROM PC OF 1777
2220  023060  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2221  023064  000002                        2                       ;INDEX
2222  023066  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2223  023070  001401                        BEQ     2$              ;BR IF YES
2224  023072  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2225  023074  104401              2$:       SCOP1                   ;LOOP TO 1$ IF SW09=1
2226  023076  012737  023104  001220        MOV     #3$,LOCK        ;NEW SCOP1
2227  023104                        3$:
2228  023104  004737  026300                JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2229  023110  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2230  023112  100403                        100403                  ;START AT ROM PC=3
2231  023114  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2232  023116  100000                        100000!<400*0>  ;JUMP TO ROM PC OF 0
2233  023120  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2234  023124  000010                        10                      ;INDEX
2235  023126  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2236  023130  001401                        BEQ     4$              ;BR IF YES
```

```
2237  023132  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2238  023134  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2239  023136  012737  023144  001220          MOV     #5$,LOCK        ;NEW SCOP1
2240  023144                          5$:
2241  023144  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2242  023150  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2243  023152  100406                          100406                  ;START AT ROM PC=6
2244  023154  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2245  023156  104125                          104125!<400*0>          ;JUMP TO ROM PC OF 525
2246  023160  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2247  023164  000016                          16                      ;INDEX
2248  023166  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2249  023170  001401                          BEQ     6$              ;BR IF YES
2250  023172  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2251  023174  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2252  023176  104400                          SCOPE                   ;SCOPE THIS TEST
2253
2254
2255                                          ;***************************** TEST 5 ****************************
2256                                          ;*CROM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
2257                                          ;*PERFORM THE JUMP INSTRUCTION
2258                                          ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2259                                          ;;*********************************************************************
2260
2261                                          ;  TEST 5
2262                                          ;--------------
2263  023200  012737  000005  001226  TST5:   MOV     #5,TSTNO
2264  023206  012737  023360  001216          MOV     #TST6,NEXT
2265  023214  012737  023240  001220          MOV     #1$,LOCK
2266                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2267  023222  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2268  023224  032737  100000  001366          BIT     #BIT15,STA11    ;IS IT CRAM?
2269  023232  001051                          BNE     6$+2            ;SKIP TEST IF YES
2270  023234  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2271  023240                          1$:
2272  023240  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2273  023242  100400                          100400                  ;START AT ROM PC=0
2274  023244  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2275  023246  114777                          114377!<400*1>          ;JUMP TO ROM PC OF 1777
2276  023250  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2277  023254  003776                          3776                    ;INDEX
2278  023256  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2279  023260  001401                          BEQ     2$              ;BR IF YES
2280  023262  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2281  023264  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2282  023266  012737  023274  001220          MOV     #3$,LOCK        ;NEW SCOP1
2283  023274                          3$:
2284  023274  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2285  023276  100403                          100403                  ;START AT ROM PC=3
2286  023300  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2287  023302  100400                          100000!<400*1>  ;JUMP TO ROM PC OF 0
2288  023304  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2289  023310  000000                          0                       ;INDEX
2290  023312  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2291  023314  001401                          BEQ     4$              ;BR IF YES
2292  023316  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
```

```
2293  023320  104401                    4$:   SCOP1                   ;LOOP TO 3$ IF SW09=1
2294  023322  012737  023330  001220          MOV     #5$,LOCK        ;NEW SCOP1
2295  023330                            5$:
2296  023330  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2297  023332  100406                          100406                  ;START AT ROM PC=6
2298  023334  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2299  023336  104525                          104125!<400*1>          ;JUMP TO ROM PC OF 525
2300  023340  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2301  023344  001252                          1252                    ;INDEX
2302  023346  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2303  023350  001401                          BEQ     6$              ;BR IF YES
2304  023352  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2305  023354  104401                    6$:   SCOP1                   ;LOOP TO 5$ IF SW59=1
2306  023356  104400                          SCOPE                   ;SCOPE THIS TEST
2307
2308
2309                                          ;***************************** TEST 6 *****************************
2310                                          ;*CROM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
2311                                          ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION,
2312                                          ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2313                                          ;:*************************************************************************
2314
2315                                          ;  TEST 6
2316                                          ;--------------
2317  023360  012737  000006  001226  TST6:   MOV     #6,TSTNO
2318  023366  012737  023554  001216          MOV     #TST7,NEXT
2319  023374  012737  023420  001220          MOV     #1$,LOCK
2320                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2321  023402  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2322  023404  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2323  023412  001057                          BNE     6$+2            ;SKIP TEST IF YES
2324  023414  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2325  023420                            1$:
2326  023420  004737  026346                  JSR     PC,SETC ;SET THE C BIT'
2327  023424  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2328  023426  100400                          100400                  ;START AT ROM PC=0
2329  023430  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2330  023432  115377                          114377!<400*2>          ;JUMP TO ROM PC OF 1777
2331  023434  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2332  023440  003776                          3776                    ;INDEX
2333  023442  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2334  023444  001401                          BEQ     2$              ;BR IF YES
2335  023446  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2336  023450  104401                    2$:   SCOP1                   ;LOOP TO 1$ IF SW09=1
2337  023452  012737  023460  001220          MOV     #3$,LOCK        ;NEW SCOP1
2338  023460                            3$:
2339  023460  004737  026346                  JSR     PC,SETC ;SET THE C BIT'
2340  023464  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2341  023466  100403                          100403                  ;START AT ROM PC=3
2342  023470  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2343  023472  101000                          100000!<400*2>  ;JUMP TO ROM PC OF 0
2344  023474  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2345  023500  000000                          0                       ;INDEX
2346  023502  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2347  023504  001401                          BEQ     4$              ;BR IF YES
2348  023506  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
```

```
2349  023510  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2350  023512  012737  023520  001220          MOV     #5$,LOCK        ;NEW SCOP1
2351  023520                          5$:
2352  023520  004737  026346                  JSR     PC,SETC ;SET THE C BIT'
2353  023524  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2354  023526  100406                          100406                  ;START AT ROM PC=6
2355  023530  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2356  023532  105125                          104125!<400*2>          ;JUMP TO ROM PC OF 525
2357  023534  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2358  023540  001252                          1252                    ;INDEX
2359  023542  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2360  023544  001401                          BEQ     6$              ;BR IF YES
2361  023546  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2362  023550  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2363  023552  104400                          SCOPE                   ;SCOPE THIS TEST
2364
2365
2366                                          ;***************************** TEST 7 *****************************
2367                                          ;*CROM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
2368                                          ;*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION,
2369                                          ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2370                                          ;:*************************************************************************
2371
2372                                          ;  TEST 7
2373                                          ;--------------
2374  023554  012737  000007  001226  TST7:   MOV     #7,TSTNO
2375  023562  012737  023750  001216          MOV     #TST10,NEXT
2376  023570  012737  023614  001220          MOV     #1$,LOCK
2377                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2378  023576  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2379  023600  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2380  023606  001057                          BNE     6$+2            ;SKIP TEST IF YES
2381  023610  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2382  023614                          1$:
2383  023614  004737  026364                  JSR     PC,SETZ ;SET THE Z BIT'
2384  023620  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2385  023622  100400                          100400                  ;START AT ROM PC=0
2386  023624  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2387  023626  115777                          114377!<400*3>          ;JUMP TO ROM PC OF 1777
2388  023630  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2389  023634  003776                          3776                    ;INDEX
2390  023636  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2391  023640  001401                          BEQ     2$              ;BR IF YES
2392  023642  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2393  023644  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2394  023646  012737  023654  001220          MOV     #3$,LOCK        ;NEW SCOP1
2395  023654                          3$:
2396  023654  004737  026364                  JSR     PC,SETZ ;SET THE Z BIT'
2397  023660  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2398  023662  100403                          100403                  ;START AT ROM PC=3
2399  023664  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2400  023666  101400                          100000!<400*3>  ;JUMP TO ROM PC OF 0
2401  023670  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2402  023674  000000                          0                       ;INDEX
2403  023676  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2404  023700  001401                          BEQ     4$              ;BR IF YES
```

```
2405  023702  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2406  023704  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2407  023706  012737  023714  001220          MOV     #5$,LOCK        ;NEW SCOP1
2408  023714                          5$:
2409  023714  004737  026364                  JSR     PC,SETZ ;SET THE Z BIT'
2410  023720  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2411  023722  100406                          100406                  ;START AT ROM PC=6
2412  023724  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2413  023726  105525                          104125!<400*3>          ;JUMP TO ROM PC OF 525
2414  023730  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2415  023734  001252                          1252                    ;INDEX
2416  023736  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2417  023740  001401                          BEQ     6$              ;BR IF YES
2418  023742  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2419  023744  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2420  023746  104400                          SCOPE                   ;SCOPE THIS TEST
2421
2422
2423                                          ;***************************** TEST 10 *****************************
2424                                          ;*CROM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.
2425                                          ;*SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION,
2426                                          ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2427                                          ;:*************************************************************************
2428
2429                                          ;  TEST 10
2430                                          ;--------------
2431  023750  012737  000010  001226  TST10:  MOV     #10,TSTNO
2432  023756  012737  024144  001216          MOV     #TST11,NEXT
2433  023764  012737  024010  001220          MOV     #1$,LOCK
2434                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2435  023772  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2436  023774  032737  100000  001366          BIT     #BIT15,STA11    ;IS IT CRAM?
2437  024002  001057                          BNE     6$+2            ;SKIP TEST IF YES
2438  024004  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2439  024010                          1$:
2440  024010  004737  026316                  JSR     PC,SETBRO       ;SET THE BRO BIT'
2441  024014  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2442  024016  100400                          100400                  ;START AT ROM PC=0
2443  024020  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2444  024022  116377                          114377!<400*4>          ;JUMP TO ROM PC OF 1777
2445  024024  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2446  024030  003776                          3776                    ;INDEX
2447  024032  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2448  024034  001401                          BEQ     2$              ;BR IF YES
2449  024036  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2450  024040  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2451  024042  012737  024050  001220          MOV     #3$,LOCK        ;NEW SCOP1
2452  024050                          3$:
2453  024050  004737  026316                  JSR     PC,SETBRO       ;SET THE BRO BIT'
2454  024054  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2455  024056  100403                          100403                  ;START AT ROM PC=3
2456  024060  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2457  024062  102000                          100000!<400*4>  ;JUMP TO ROM PC OF 0
2458  024064  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2459  024070  000000                          0                       ;INDEX
2460  024072  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
```

```
2461  024074  001401                        BEQ     4$              ;BR IF YES
2462  024076  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2463  024100  104401                4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2464  024102  012737  024110  001220        MOV     #5$,LOCK        ;NEW SCOP1
2465  024110                         5$:
2466  024110  004737  026316                JSR     PC,SETBR0       ;SET THE BR0 BIT'
2467  024114  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2468  024116  100406                        100406                  ;START AT ROM PC=6
2469  024120  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2470  024122  106125                        104125!<400*4>          ;JUMP TO ROM PC OF 525
2471  024124  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2472  024130  001252                        1252                    ;INDEX
2473  024132  020504                        CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2474  024134  001401                        BEQ     6$              ;BR IF YES
2475  024136  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2476  024140  104401                6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2477  024142  104400                        SCOPE                   ;SCOPE THIS TEST
2478
2479
2480                                        ;***************************** TEST 11 ****************************
2481                                        ;*CROM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
2482                                        ;*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION,
2483                                        ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2484                                        ;:**************************************************************
2485
2486                                        ;  TEST 11
2487                                        ;--------------
2488  024144  012737  000011  001226  TST11: MOV     #11,TSTNO
2489  024152  012737  024340  001216        MOV     #TST12,NEXT
2490  024160  012737  024204  001220        MOV     #1$,LOCK
2491                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
2492  024166  104412                        MSTCLR                  ;MASTER CLEAR M8200-YC
2493  024170  032737  100000  001366        BIT     #BIT15,STAT1    ;IS IT CRAM?
2494  024176  001057                        BNE     6$+2            ;SKIP TEST IF YES
2495  024200  004737  026434                JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2496  024204                         1$:
2497  024204  004737  026324                JSR     PC,SETBR1       ;SET THE BR1 BIT'
2498  024210  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2499  024212  100400                        100400                  ;START AT ROM PC=0
2500  024214  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2501  024216  116777                        114377!<400*5>          ;JUMP TO ROM PC OF 1777
2502  024220  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2503  024224  003776                        3776                    ;INDEX
2504  024226  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2505  024230  001401                        BEQ     2$              ;BR IF YES
2506  024232  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2507  024234  104401                2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2508  024236  012737  024244  001220        MOV     #3$,LOCK        ;NEW SCOP1
2509  024244                         3$:
2510  024244  004737  026324                JSR     PC,SETBR1       ;SET THE BR1 BIT'
2511  024250  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2512  024252  100403                        100403                  ;START AT ROM PC=3
2513  024254  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2514  024256  102400                        100000!<400*5>  ;JUMP TO ROM PC OF 0
2515  024260  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2516  024264  000000                        0                       ;INDEX
```

```
2517  024266  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2518  024270  001401                        BEQ     4$              ;BR IF YES
2519  024272  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2520  024274  104401                4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2521  024276  012737  024304  001220        MOV     #5$,LOCK        ;NEW SCOP1
2522  024304                        5$:
2523  024304  004737  026324                JSR     PC,SETBR1       ;SET THE BR1 BIT'
2524  024310  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2525  024312  100406                        100406                  ;START AT ROM PC=6
2526  024314  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2527  024316  106525                        104125!<400*5>          ;JUMP TO ROM PC OF 525
2528  024320  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2529  024324  001252                        1252                    ;INDEX
2530  024326  020504                        CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2531  024330  001401                        BEQ     6$              ;BR IF YES
2532  024332  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2533  024334  104401                6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2534  024336  104400                        SCOPE                   ;SCOPE THIS TEST
2535
2536
2537                                        ;*************************** TEST 12 ***************************
2538                                        ;*CROM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
2539                                        ;*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION,
2540                                        ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2541                                        ;:***********************************************************************
2542
2543                                        ;  TEST 12
2544                                        ;--------------
2545  024340  012737  000012  001226  TST12: MOV     #12,TSTNO
2546  024346  012737  024534  001216        MOV     #TST13,NEXT
2547  024354  012737  024400  001220        MOV     #1$,LOCK
2548                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
2549  024362  104412                        MSTCLR                  ;MASTER CLEAR M8200-YC
2550  024364  032737  100000  001366        BIT     #BIT15,STAT1    ;IS IT CRAM?
2551  024372  001057                        BNE     6$+2            ;SKIP TEST IF YES
2552  024374  004737  026434                JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2553  024400                        1$:
2554  024400  004737  026332                JSR     PC,SETBR4       ;SET THE BR4 BIT'
2555  024404  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2556  024406  100400                        100400                  ;START AT ROM PC=0
2557  024410  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2558  024412  117377                        114377!<400*6>          ;JUMP TO ROM PC OF 1777
2559  024414  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2560  024420  003776                        3776                    ;INDEX
2561  024422  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2562  024424  001401                        BEQ     2$              ;BR IF YES
2563  024426  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2564  024430  104401                2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2565  024432  012737  024440  001220        MOV     #3$,LOCK        ;NEW SCOP1
2566  024440                        3$:
2567  024440  004737  026332                JSR     PC,SETBR4       ;SET THE BR4 BIT'
2568  024444  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2569  024446  100403                        100403                  ;START AT ROM PC=3
2570  024450  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2571  024452  103000                        100000!<400*6>  ;JUMP TO ROM PC OF 0
2572  024454  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
```

```
2573  024460  000000                          0                       ;INDEX
2574  024462  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2575  024464  001401                          BEQ     4$              ;BR IF YES
2576  024466  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2577  024470  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2578  024472  012737  024500  001220          MOV     #5$,LOCK        ;NEW SCOP1
2579  024500                          5$:
2580  024500  004737  026332                  JSR     PC,SETBR4       ;SET THE BR4 BIT'
2581  024504  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2582  024506  100406                          100406                  ;START AT ROM PC=6
2583  024510  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2584  024512  107125                          104125!<400*6>          ;JUMP TO ROM PC OF 525
2585  024514  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2586  024520  001252                          1252                    ;INDEX
2587  024522  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2588  024524  001401                          BEQ     6$              ;BR IF YES
2589  024526  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2590  024530  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2591  024532  104400                          SCOPE                   ;SCOPE THIS TEST
2592
2593
2594                                          ;*************************** TEST 13 ***************************
2595                                          ;*CROM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
2596                                          ;*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION,
2597                                          ;*VERIFY THE JUMP BY READING THE CONTENTS OF THE NEW ROM PC
2598                                          ;;***********************************************************************
2599
2600                                          ;  TEST 13
2601                                          ;--------------
2602  024534  012737  000013  001226  TST13:  MOV     #13,TSTNO
2603  024542  012737  024730  001216          MOV     #TST14,NEXT
2604  024550  012737  024574  001220          MOV     #1$,LOCK
2605                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2606  024556  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2607  024560  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2608  024566  001057                          BNE     6$+2            ;SKIP TEST IF YES
2609  024570  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2610  024574                          1$:
2611  024574  004737  026340                  JSR     PC,SETBR7       ;SET THE BR7 BIT'
2612  024600  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2613  024602  100400                          100400                  ;START AT ROM PC=0
2614  024604  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2615  024606  117777                          114377!<400*7>          ;JUMP TO ROM PC OF 1777
2616  024610  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2617  024614  003776                          3776                    ;INDEX
2618  024616  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2619  024620  001401                          BEQ     2$              ;BR IF YES
2620  024622  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2621  024624  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2622  024626  012737  024634  001220          MOV     #3$,LOCK        ;NEW SCOP1
2623  024634                          3$:
2624  024634  004737  026340                  JSR     PC,SETBR7       ;SET THE BR7 BIT'
2625  024640  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2626  024642  100403                          100403                  ;START AT ROM PC=3
2627  024644  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2628  024646  103400                          100000!<400*7>  ;JUMP TO ROM PC OF 0
```

```
2629  024650  004737  026372                    JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2630  024654  000000                            0                       ;INDEX
2631  024656  020504                            CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2632  024660  001401                            BEQ     4$              ;BR IF YES
2633  024662  104006                            HLT     6               ;ERROR, CROM PC IS WRONG
2634  024664  104401                    4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2635  024666  012737  024674  001220            MOV     #5$,LOCK        ;NEW SCOP1
2636  024674                            5$:
2637  024674  004737  026340                    JSR     PC,SETBR7       ;SET THE BR7 BIT'
2638  024700  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2639  024702  100406                            100406                  ;START AT ROM PC=6
2640  024704  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2641  024706  107525                            104125!<400*7>          ;JUMP TO ROM PC OF 525
2642  024710  004737  026372                    JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2643  024714  001252                            1252                    ;INDEX
2644  024716  020504                            CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2645  024720  001401                            BEQ     6$              ;BR IF YES
2646  024722  104006                            HLT     6               ;ERROR, CROM PC IS WRONG
2647  024724  104401                    6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2648  024726  104400                            SCOPE                   ;SCOPE THIS TEST
2649
2650
2651                                            ;************************** TEST 14 **************************
2652                                            ;*CROM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
2653                                            ;*CLEAR THE C BIT, PERFORM THE JUMP INSTRUCTION,
2654                                            ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2655                                            ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2656                                            ;:**********************************************************************
2657
2658                                            ;  TEST 14
2659                                            ;--------------
2660  024730  012737  000014  001226    TST14:  MOV     #14,TSTNO
2661  024736  012737  025124  001216            MOV     #TST15,NEXT
2662  024744  012737  024770  001220            MOV     #1$,LOCK
2663                                                                    ;R1 CONTAINS BASE M8200-YC ADDRESS
2664  024752  104412                            MSTCLR                  ;MASTER CLEAR M8200-YC
2665  024754  032737  100000  001366            BIT     #BIT15,STAT1    ;IS IT CRAM?
2666  024762  001057                            BNE     6$+2            ;SKIP TEST IF YES
2667  024764  004737  026434                    JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2668  024770                            1$:
2669  024770  004737  026300                    JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2670  024774  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2671  024776  100400                            100400                  ;START AT ROM PC=0
2672  025000  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2673  025002  115377                            114377!<400*2>          ;JUMP TO ROM PC OF 1777
2674  025004  004737  026372                    JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2675  025010  000002                            2                       ;INDEX
2676  025012  020504                            CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2677  025014  001401                            BEQ     2$              ;BR IF YES
2678  025016  104006                            HLT     6               ;ERROR, CROM PC IS WRONG
2679  025020  104401                    2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2680  025022  012737  025030  001220            MOV     #3$,LOCK        ;NEW SCOP1
2681  025030                            3$:
2682  025030  004737  026300                    JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2683  025034  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2684  025036  100403                            100403                  ;START AT ROM PC=3
```

```
2685  025040  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2686  025042  101000                          100000!<400*2>  ;JUMP TO ROM PC OF 0
2687  025044  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2688  025050  000010                          10                      ;INDEX
2689  025052  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2690  025054  001401                          BEQ     4$              ;BR IF YES
2691  025056  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2692  025060  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2693  025062  012737  025070  001220          MOV     #5$,LOCK        ;NEW SCOP1
2694  025070                          5$:
2695  025070  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2696  025074  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2697  025076  100406                          100406                  ;START AT ROM PC=6
2698  025100  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2699  025102  105125                          104125!<400*2>          ;JUMP TO ROM PC OF 525
2700  025104  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2701  025110  000016                          16                      ;INDEX
2702  025112  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2703  025114  001401                          BEQ     6$              ;BR IF YES
2704  025116  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2705  025120  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2706  025122  104400                          SCOPE                   ;SCOPE THIS TEST
2707
2708
2709                                          ;************************** TEST 15 **************************
2710                                          ;*CROM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
2711                                          ;*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION,
2712                                          ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2713                                          ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2714                                          ;;**********************************************************************
2715
2716                                          ;  TEST 15
2717                                          ;--------------
2718  025124  012737  000015  001226  TST15:  MOV     #15,TSTNO
2719  025132  012737  025320  001216          MOV     #TST16,NEXT
2720  025140  012737  025164  001220          MOV     #1$,LOCK
2721                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2722  025146  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2723  025150  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2724  025156  001057                          BNE     6$+2            ;SKIP TEST IF YES
2725  025160  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2726  025164                          1$:
2727  025164  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2728  025170  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2729  025172  100400                          100400                  ;START AT ROM PC=0
2730  025174  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2731  025176  115777                          114377!<400*3>          ;JUMP TO ROM PC OF 1777
2732  025200  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2733  025204  000002                          2                       ;INDEX
2734  025206  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2735  025210  001401                          BEQ     2$              ;BR IF YES
2736  025212  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2737  025214  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2738  025216  012737  025224  001220          MOV     #3$,LOCK        ;NEW SCOP1
2739  025224                          3$:
2740  025224  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
```

```
2741  025230  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2742  025232  100403                          100403                  ;START AT ROM PC=3
2743  025234  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2744  025236  101400                          100000!<400*3>  ;JUMP TO ROM PC OF 0
2745  025240  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2746  025244  000010                          10                      ;INDEX
2747  025246  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2748  025250  001401                          BEQ     4$              ;BR IF YES
2749  025252  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2750  025254  104401              4$:         SCOP1                   ;LOOP TO 3$ IF SW09=1
2751  025256  012737  025264  001220          MOV     #5$,LOCK        ;NEW SCOP1
2752  025264                      5$:
2753  025264  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2754  025270  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2755  025272  100406                          100406                  ;START AT ROM PC=6
2756  025274  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2757  025276  105525                          104125!<400*3>          ;JUMP TO ROM PC OF 525
2758  025300  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2759  025304  000016                          16                      ;INDEX
2760  025306  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2761  025310  001401                          BEQ     6$              ;BR IF YES
2762  025312  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2763  025314  104401              6$:         SCOP1                   ;LOOP TO 5$ IF SW59=1
2764  025316  104400                          SCOPE                   ;SCOPE THIS TEST
2765
2766
2767                                          ;***************************** TEST 16 *****************************
2768                                          ;*CROM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.
2769                                          ;*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION,
2770                                          ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2771                                          ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2772                                          ;:*************************************************************************
2773
2774                                          ;  TEST 16
2775                                          ;--------------
2776  025320  012737  000016  001226  TST16:  MOV     #16,TSTNO
2777  025326  012737  025514  001216          MOV     #TST17,NEXT
2778  025334  012737  025360  001220          MOV     #1$,LOCK
2779                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2780  025342  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2781  025344  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2782  025352  001057                          BNE     6$+2            ;SKIP TEST IF YES
2783  025354  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2784  025360                      1$:
2785  025360  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2786  025364  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2787  025366  100400                          100400                  ;START AT ROM PC=0
2788  025370  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2789  025372  116377                          114377!<400*4>          ;JUMP TO ROM PC OF 1777
2790  025374  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2791  025400  000002                          2                       ;INDEX
2792  025402  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2793  025404  001401                          BEQ     2$              ;BR IF YES
2794  025406  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2795  025410  104401              2$:         SCOP1                   ;LOOP TO 1$ IF SW09=1
2796  025412  012737  025420  001220          MOV     #3$,LOCK        ;NEW SCOP1
```

```
2797  025420                        3$:
2798  025420  004737  026300              JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2799  025424  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2800  025426  100403                      100403                  ;START AT ROM PC=3
2801  025430  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2802  025432  102000                      100000!<400*4>  ;JUMP TO ROM PC OF 0
2803  025434  004737  026372              JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2804  025440  000010                      10                      ;INDEX
2805  025442  020504                      CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2806  025444  001401                      BEQ     4$              ;BR IF YES
2807  025446  104006                      HLT     6               ;ERROR, CROM PC IS WRONG
2808  025450  104401              4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2809  025452  012737  025460  001220      MOV     #5$,LOCK        ;NEW SCOP1
2810  025460                        5$:
2811  025460  004737  026300              JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2812  025464  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2813  025466  100406                      100406                  ;START AT ROM PC=6
2814  025470  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2815  025472  106125                      104125!<400*4>          ;JUMP TO ROM PC OF 525
2816  025474  004737  026372              JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2817  025500  000016                      16                      ;INDEX
2818  025502  020504                      CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2819  025504  001401                      BEQ     6$              ;BR IF YES
2820  025506  104006                      HLT     6               ;ERROR, CROM PC IS WRONG
2821  025510  104401              6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2822  025512  104400                      SCOPE                   ;SCOPE THIS TEST
2823
2824
2825                                      ;*************************** TEST 17 ****************************
2826                                      ;*CROM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
2827                                      ;*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION,
2828                                      ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2829                                      ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2830                                      ;;*************************************************************************
2831
2832                                      ;  TEST 17
2833                                      ;--------------
2834  025514  012737  000017  001226  TST17:  MOV     #17,TSTNO
2835  025522  012737  025710  001216          MOV     #TST20,NEXT
2836  025530  012737  025554  001220          MOV     #1$,LOCK
2837                                                          ;R1 CONTAINS BASE M8200-YC ADDRESS
2838  025536  104412                      MSTCLR              ;MASTER CLEAR M8200-YC
2839  025540  032737  100000  001366      BIT     #BIT15,STAT1    ;IS IT CRAM?
2840  025546  001057                      BNE     6$+2            ;SKIP TEST IF YES
2841  025550  004737  026434              JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2842  025554                        1$:
2843  025554  004737  026300              JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2844  025560  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2845  025562  100400                      100400                  ;START AT ROM PC=0
2846  025564  104414                      ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2847  025566  116777                      114377!<400*5>          ;JUMP TO ROM PC OF 1777
2848  025570  004737  026372              JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2849  025574  000002                      2                       ;INDEX
2850  025576  020504                      CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2851  025600  001401                      BEQ     2$              ;BR IF YES
2852  025602  104006                      HLT     6               ;ERROR, CROM PC IS WRONG
```

```
2853  025604  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2854  025606  012737  025614  001220          MOV     #3$,LOCK        ;NEW SCOP1
2855  025614                          3$:
2856  025614  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2857  025620  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2858  025622  100403                          100403                  ;START AT ROM PC=3
2859  025624  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2860  025626  102400                          100000!<400*5>  ;JUMP TO ROM PC OF 0
2861  025630  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2862  025634  000010                          10                      ;INDEX
2863  025636  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2864  025640  001401                          BEQ     4$              ;BR IF YES
2865  025642  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2866  025644  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2867  025646  012737  025654  001220          MOV     #5$,LOCK        ;NEW SCOP1
2868  025654                          5$:
2869  025654  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2870  025660  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2871  025662  100406                          100406                  ;START AT ROM PC=6
2872  025664  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2873  025666  106525                          104125!<400*5>          ;JUMP TO ROM PC OF 525
2874  025670  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2875  025674  000016                          16                      ;INDEX
2876  025676  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2877  025700  001401                          BEQ     6$              ;BR IF YES
2878  025702  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2879  025704  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2880  025706  104400                          SCOPE                   ;SCOPE THIS TEST
2881
2882
2883                                          ;***************************** TEST 20 *****************************
2884                                          ;*CROM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
2885                                          ;*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION,
2886                                          ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2887                                          ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2888                                          ;:*************************************************************************
2889
2890                                          ;  TEST 20
2891                                          ;--------------
2892  025710  012737  000020  001226  TST20:  MOV     #20,TSTNO
2893  025716  012737  026104  001216          MOV     #TST21,NEXT
2894  025724  012737  025750  001220          MOV     #1$,LOCK
2895                                                                  ;R1 CONTAINS BASE M8200-YC ADDRESS
2896  025732  104412                          MSTCLR                  ;MASTER CLEAR M8200-YC
2897  025734  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT CRAM?
2898  025742  001057                          BNE     6$+2            ;SKIP TEST IF YES
2899  025744  004737  026434                  JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2900  025750                          1$:
2901  025750  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2902  025754  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2903  025756  100400                          100400                  ;START AT ROM PC=0
2904  025760  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2905  025762  117377                          114377!<400*6>          ;JUMP TO ROM PC OF 1777
2906  025764  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2907  025770  000002                          2                       ;INDEX
2908  025772  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
```

```
2909  025774  001401                        BEQ     2$              ;BR IF YES
2910  025776  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2911  026000  104401                2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2912  026002  012737  026010  001220        MOV     #3$,LOCK        ;NEW SCOP1
2913  026010                        3$:
2914  026010  004737  026300                JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2915  026014  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2916  026016  100403                        100403                  ;START AT ROM PC=3
2917  026020  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2918  026022  103000                        100000!<400*6>  ;JUMP TO ROM PC OF 0
2919  026024  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2920  026030  000010                        10                      ;INDEX
2921  026032  020504                        CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2922  026034  001401                        BEQ     4$              ;BR IF YES
2923  026036  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2924  026040  104401                4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2925  026042  012737  026050  001220        MOV     #5$,LOCK        ;NEW SCOP1
2926  026050                        5$:
2927  026050  004737  026300                JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2928  026054  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2929  026056  100406                        100406                  ;START AT ROM PC=6
2930  026060  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2931  026062  107125                        104125!<400*6>          ;JUMP TO ROM PC OF 525
2932  026064  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2933  026070  000016                        16                      ;INDEX
2934  026072  020504                        CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2935  026074  001401                        BEQ     6$              ;BR IF YES
2936  026076  104006                        HLT     6               ;ERROR, CROM PC IS WRONG
2937  026100  104401                6$:     SCOP1                   ;LOOP TO 5$ IF SW59=1
2938  026102  104400                        SCOPE                   ;SCOPE THIS TEST
2939
2940
2941                                        ;***************************** TEST 21 ****************************
2942                                        ;*CROM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
2943                                        ;*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION,
2944                                        ;*VERIFY THAT THE JUMP DID NOT OCCUR BY READING
2945                                        ;*THE CONTENTS OF THE NEW ROM PC(IT SHOULD INCREMENT BY ONE).
2946                                        ;;*************************************************************************
2947
2948                                        ;  TEST 21
2949                                        ;--------------
2950  026104  012737  000021  001226  TST21: MOV     #21,TSTNO
2951  026112  012737  003374  001216        MOV     #.EOP,NEXT
2952  026120  012737  026144  001220        MOV     #1$,LOCK
2953                                                                ;R1 CONTAINS BASE M8200-YC ADDRESS
2954  026126  104412                        MSTCLR                  ;MASTER CLEAR M8200-YC
2955  026130  032737  100000  001366        BIT     #BIT15,STAT1    ;IS IT CRAM?
2956  026136  001057                        BNE     6$+2            ;SKIP TEST IF YES
2957  026140  004737  026434                JSR     PC,MAPCK        ;CHECK FOR HI OR LO
2958  026144                        1$:
2959  026144  004737  026300                JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2960  026150  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2961  026152  100400                        100400                  ;START AT ROM PC=0
2962  026154  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2963  026156  117777                        114377!<400*7>          ;JUMP TO ROM PC OF 1777
2964  026160  004737  026372                JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
```

```
2965  026164  000002                          2                       ;INDEX
2966  026166  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2967  026170  001401                          BEQ     2$              ;BR IF YES
2968  026172  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2969  026174  104401              2$:         SCOP1                   ;LOOP TO 1$ IF SW09=1
2970  026176  012737  026204  001220          MOV     #3$,LOCK        ;NEW SCOP1
2971  026204                      3$:
2972  026204  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2973  026210  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2974  026212  100403                          100403                  ;START AT ROM PC=3
2975  026214  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2976  026216  103400                          100000!<400*7>  ;JUMP TO ROM PC OF 0
2977  026220  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2978  026224  000010                          10                      ;INDEX
2979  026226  020504                          CMP     R5,R4           ;ARE NEW PC CONTENTS CORRECT?
2980  026230  001401                          BEQ     4$              ;BR IF YES
2981  026232  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2982  026234  104401              4$:         SCOP1                   ;LOOP TO 3$ IF SW09=1
2983  026236  012737  026244  001220          MOV     #5$,LOCK        ;NEW SCOP1
2984  026244                      5$:
2985  026244  004737  026300                  JSR     PC,CLRALL       ;CLEAR ALL  CONDITIONS
2986  026250  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2987  026252  100406                          100406                  ;START AT ROM PC=6
2988  026254  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2989  026256  107525                          104125!<400*7>          ;JUMP TO ROM PC OF 525
2990  026260  004737  026372                  JSR     PC,ROMDAT       ;R5=EXPECTED ROM DATA,R4=ACTUAL ROM DATA
2991  026264  000016                          16                      ;INDEX
2992  026266  020504                          CMP     R5,R4           ;ARE NEW ROM PC CONTENTS CORRECT?
2993  026270  001401                          BEQ     6$              ;BR IF YES
2994  026272  104006                          HLT     6               ;ERROR, CROM PC IS WRONG
2995  026274  104401              6$:         SCOP1                   ;LOOP TO 5$ IF SW59=1
2996  026276  104400                          SCOPE                   ;SCOPE THIS TEST
2997
2998
2999                              ;SUBROUTINES
3000                              ;-----------
3001
3002  026300                      CLRALL:
3003                                      ;THIS SUBROUTINE CLEARS THE C&Z BITS AND THE BR
3004
3005  026300  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3006  026302  000400                          000400                  ;BR_0
3007  026304  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3008  026306  063220                          063220                  ;SP(0)_BR
3009  026310  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3010  026312  060400                          060400                  ;BR_SP(0)+BR
3011  026314  000207                          RTS     PC
3012
3013
3014  026316                      SETBR0:
3015                                      ;THIS SUBROUTINE SETS BR0 BIT
3016
3017  026316  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3018  026320  000401                          000401                  ;BR_001
3019  026322  000207                          RTS     PC
3020
```

```
3021
3022  026324                        SETBR1:
3023                                        ;THIS SUBROUTINE SETS BR1 BIT
3024
3025  026324  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3026  026326  000402                        000402                  ;BR_002
3027  026330  000207                        RTS     PC
3028
3029
3030  026332                        SETBR4:
3031                                        ;THIS SUBROUTINE SETS BR4 BIT
3032
3033  026332  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3034  026334  000420                        000420                  ;BR_020
3035  026336  000207                        RTS     PC
3036
3037
3038  026340                        SETBR7:
3039                                        ;THIS SUBROUTINE SETS BR7 BIT
3040
3041  026340  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3042  026342  000600                        000600                  ;BR_200
3043  026344  000207                        RTS     PC
3044
3045
3046  026346                        SETC:
3047                                        ;THIS SUBROUTINE SETS THE C BIT
3048
3049  026346  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3050  026350  000777                        000777                  ;BR_377
3051  026352  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3052  026354  063220                        063220                  ;SP(0)_BR
3053  026356  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3054  026360  060400                        060400                  ;BR_SP(0)+BR
3055  026362  000207                        RTS     PC
3056
3057
3058  026364                        SETZ:
3059                                        ;THIS SUBROUTINE SETS THE Z BIT
3060
3061  026364  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3062  026366  000777                        000777                  ;BR_377
3063  026370  000207                        RTS     PC
3064
3065
3066  026372                        ROMDAT:
3067                                        ;THIS SUBROUTINE LOADS R5 WITH EXPECTED ROM CONTENTS
3068                                        ;AND LOADS R4 WITH ACTUAL ROM CONTENTS
3069
3070  026372  017600  000000               MOV     @(SP),R0        ;INDEX FOR COMPARE
3071  026376  062716  000002               ADD     #2,(SP)         ;ADJUST STACK
3072  026402  012711  002000               MOV     #BIT10,(R1)     ;SET ROM0
3073  026406  016005  016402               MOV     V5MAP(R0),R5    ;PUT EXPECTED IN R5 (VERSION 5)
3074  026412  032777  000020  152562       BIT     #BIT4,@SWR      ;TEST IF V4 MICRO-CODE
3075  026420  001402                       BEQ     1$              ;BR IF V5 MICRO-CODE
3076  026422  016005  012400               MOV     V4MAP(R0),R5    ;PUT EXPECTED IN R5 (VERSION 4)
```

```
3077  026426  016104  000006          1$:     MOV     6(R1),R4        ;PUT "FOUND" IN R4
3078  026432  000207                          RTS     PC              ;RETURN
3079
3080  026434                          MAPCK:
3081                                          ;THIS SUBROUTINE CHECKS THE STATUS TABLE AND LOADS
3082                                          ;THE ROMMAP POINTER TO POINT TO EITHER THE HIGH OR
3083                                          ;LOW SPEED MICRO-CODE.
3084
3085  026434  012737  016402  012376          MOV     #V5MAP,ROMMAP   ;LOAD POINTER TO V5 MICRO-CODE
3086  026442  032777  000020  152532          BIT     #BIT4,@SWR      ;CHECK SWITCH REGISTER BIT 4
3087  026450  001403                          BEQ     1$              ;BR IF V5 MICRO-CODE
3088  026452  012737  012400  012376          MOV     #V4MAP,ROMMAP   ;LOAD POINTER TO V5 MICRO-CODE
3089  026460  000207                  1$:     RTS     PC              ;RETURN
3090
3091  026462  020200  020040  020040  MESWCH: .ASCII  <200>#     NOTE:#
      026475     200  047506  020122          .ASCII  <200>#FOR THIS PROGRAM TO RUN PROPERLY, SWITCH#
      026546  033600  020054  043117          .ASCII  <200>#7, OF THE VECTOR ADDRESS SWITCH PACK (E76),#
      026622  046600  051525  020124          .ASCIZ  <200>#MUST BE ON. (M8200-YC BOARD)#<200>

      026661     377  051103  046501  EM1:    .ASCIZ  <377>/CRAM DATA ERROR/
      026702  041777  040522  020115  EM2:    .ASCIZ  <377>/CRAM DUAL ADDRESSING ERROR/
      026736  041777  047522  020115  EM3:    .ASCIZ  <377>/CROM DATA ERROR/
      026757     377  052512  050115  EM4:    .ASCIZ  <377>/JUMP ERROR/
      026773     377  042117  020124  EM5:    .ASCIZ  <377>/ODT ERROR IN IBUS* REG10/
      027025     377  047511  020120  FM7:    .ASCIZ  <377>/IOP MAR TEST/
      027043     377  051102  051040  EM10:   .ASCIZ  <377>/BR RIGHT SHIFT TEST/
      027070  051377  041505  044505  EM11:   .ASCIZ  <377>/RECEIVE DATA ERROR/
      027114  043377  042522  020105  EM12:   .ASCIZ  <377>/FREE RUNNING ERROR/
      027140  041777  047117  051124  EM13:   .ASCIZ  <377>/CONTROL OUT ERROR/

      027163     377  054105  042520  DH1:    .ASCIZ  <377>/EXPECTED  FOUND  ADDRESS/
      027215     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND/
      027236  020377  042523  032114  DH3:    .ASCIZ  <377>/ SEL4      SEL6/
      027257     377  041412  046122  ROM1:   .ASCII  <377><12>/CRLPM-B SUPPORTS THE FOLLOWING CROM VERSIONS:/<200>
      027337     114  040520  020055          .ASCII  /LPA- M8200-YC  VERSION 5 MICRO CODE IF SWR = 0/<200>
      027416  050114  026501  046440          .ASCIZ  /LPA- M8200-YC  VERSION 4 MICRO CODE IF SWR = 20/<200>
      027500                          .EVEN

      027500  000003                  DT1:    3
      027502     006      004                 .BYTE   6,4
      027504  001264                          SAVR2
      027506     006      004                 .BYTE   6,4
      027510  001270                          SAVR4
      027512     004      002                 .BYTE   4,2
      027514  001260                          SAVR0
      027516  000003                  DT2:    3
      027520     006      004                 .BYTE   6,4
      027522  001272                          SAVR5
      027524     006      004                 .BYTE   6,4
      027526  001270                          SAVR4
      027530     004      002                 .BYTE   4,2
      027532  001264                          SAVR2
      027534  000003                  DT3:    3
      027536     006      004                 .BYTE   6,4
      027540  001272                          SAVR5
      027542     006      004                 .BYTE   6,4
```

```
027544  001270                          SAVR4
027546     004      002                 .BYTE   4,2
027550  001252                          TEMP3
027552  000002                  DT4:    2
027554     003      007                 .BYTE   3,7
027556  001272                          SAVR5
027560     003      002                 .BYTE   3,2
027562  001270                          SAVR4
027564  000002                  DT5:    2
027566     006      004                 .BYTE   6,4
027570  001272                          SAVR5
027572     006      002                 .BYTE   6,2
027574  001270                          SAVR4
027576  000003                  DT7:    3
027600     003      010                 .BYTE   3,10
027602  001272                          SAVR5
027604     003      004                 .BYTE   3,4
027606  001270                          SAVR4
027610     004      002                 .BYTE   4,2
027612  001264                          SAVR2
027614  000003                  DT10:   3
027616     003      007                 .BYTE   3,7
027620  001272                          SAVR5
027622     003      004                 .BYTE   3,4
027624  001270                          SAVR4
027626     006      002                 .BYTE   6,2
027630  001252                          TEMP3
027632  000002                  DT11:   2
027634     006      004                 .BYTE   6,4
027636  001252                          TEMP3
027640     006      002                 .BYTE   6,2
027642  001254                          TEMP4

027644                          .ERRTAB:
027644  000000                          0
027646  000000                          0
027650  000000                          0
027652  026661                  EM1
027654  027163                  DH1     ;HLT    1
027656  027500                  DT1
027660  026702                  EM2
027662  027163                  DH1     ;HLT    2
027664  027500                  DT1
027666  026661                  EM1
027670  027163                  DH1     ;HLT    3
027672  027516                  DT2
027674  026736                  EM3
027676  027163                  DH1     ;HLT    4
027700  027534                  DT3
027702  026757                  EM4
027704  027215                  DH2     ;HLT    5
027706  027552                  DT4
027710  026757                  EM4
027712  027215                  DH2     ;HLT    6
027714  027564                  DT5
027716  026773                  EM5
```

```
027720  027215                  DH2     ;HLT    7
027722  027552                  DT4
027724  000000                  0
027726  000000                  0
027730  000000                  0
027732  027025                  EM7
027734  027163                  DH1     ;HLT    11
027736  027576                  DT7
027740  027043                  EM10
027742  027215                  DH2     ;HLT    12
027744  027552                  DT4
027746  027070                  EM11
027750  027163                  DH1     ;HLT    13
027752  027614                  DT10
027754  027114                  EM12
027756  000000                  0       ;HLT    14
027760  000000                  0
027762  027114                  EM12
027764  027215                  DH2     ;HLT    15
027766  027564                  DT5
027770  027140                  EM13
027772  027236                  DH3     ;HLT    16
027774  027632                  DT11


027776                          CORMAX:
        000001                  .END
```

```
ADRCNT= 004403          876*   912*   921#
AUDONE  003034          566    605    610     656#
AUSTRT  002456          565#   660
AUTO.S  010556          523   1365#
BINWRD  004724          962*   965*   966     1003#
BIT0   = 000001           95#  1152   1153
BIT1   = 000002           94#   528   1146    1152   1153   1448   1534   1547
BIT10  = 002000           85#  1515   1526    2092   2173   3072
BIT11  = 004000           84#
BIT12  = 010000           83#  1463   1545
BIT13  = 020000           82#  1466   1518    1549   2095
BIT14  = 040000           81#   778   1477    1479   1549   1560
BIT15  = 100000           80#   480    569     572    593    596   1450   1521   2088   2161   2211   2268   2322
                        2379   2436   2493    2550   2607   2665   2723   2781   2839   2897   2955
BIT2   = 000004           93#   528    709    1153
BIT3   = 000010           92#  1551   1558
BIT4   = 000020           91#  1136   1159    1161   3074   3086
BIT5   = 000040           90#  1663
BIT6   = 000100           89#  1141   1142    1553
BIT7   = 000200           88#  1142   1262    1663
BIT8   = 000400           87#  1525   1540    1542   1555   1557   1563   1566   1624
BIT9   = 001000           86#  1523   1525    1537   1563   1566   1622   1624
BM      007101          1184#  1491
BRLVL   012330          1619   1629   1637    1649#
BRW     003740           715    801#
BRX     003742           716    802#
BUF1    012400 G        1673#
CHRCNT  004722           960*   963    967     983*   1001#  1002
CKSWR   007652           509    776    808     1023   1209#
CKSWR1  007732          1222#  1234
CKSWR2  007744          1225#
CKSWR3  007750          1227#
CKSWK4  007754          1228#  1236   1243
CKSWR5  010060          1210   1217   1252#
CLKX    001242           171#
CLRALL  026300          2215   2228   2241    2669   2682   2695   2727   2740   2753   2785   2798   2811   2843
                        2856   2869   2901    2914   2927   2959   2972   2985   3002#
CNERR   007327           623   1184#
CNT.MA  001702           196    364#   479     481    483   1290
CNVRT  = 104411           233#   620    735     737    739    741   1057   1059   1119   1225
CONERR  007250           618   1184#
CONN    007141          1184#  1468
CONTAB  003006           627    640#
CONVRT= 104410           231#   540    626     1073   1388
CORMAX  027776          3091#
CRAM    006633          1184#  1429
CREAM   001320           195#   478*  1286*   1287   1289*  1293
CSR     006535          1184#  1392
CSRMAP  010560          1367#
CYCLE   010124           717    758    759     1277#
DATABP  005226          1046*  1049   1071    1074#
DATACL= 104415           241#
DATAHD  005214          1045*  1067   1070#
DELAY  = 104413           237#
DEVADR  004400           874*   909    919#
DEVTAB  003020           580    645#
```

```
DH1     027163            3091#
DH2     027215            3091#
DH3     027236            3091#
DISPLA  001200             142#   493*   499*   732*
DISPRE  000174             128#   499
DMACTV  001306             189#   518    673*   674    1277   1291   1373*  1579*  1585*  1586*  1590   1615
DMCM    007350             631   1184#
DMCR00  001500             279#
DMCR01  001510             284#
DMCR02  001520             289#
DMCR03  001530             294#
DMCR04  001540             299#
DMCR05  001550             304#
DMCR06  001560             309#
DMCR07  001570             314#
DMCR10  001600             319#
DMCR11  001610             324#
DMCR12  001620             329#
DMCR13  001630             334#
DMCR14  001640             339#
DMCR15  001650             344#
DMCR16  001660             349#
DMCR17  001670             354#
DMCSR   001404             262#   565*   599    604*   609*   644    762    799    1091   1114   1160   1295*  1304
                          1353   2115
DMCSRH  001406             263#   1141*  1142*  1146*  1152*  1153*  1159*  1161*  1304*  1305*  1306
DMCTL   001410             264#   1306*  1307*  1308
DMNUM   001310             190#   474    748    1371*  1386*  1570*  1571   1580   1582
DMPO4   001412             265#   1130*  1136   1173   1178   1308*  1309*  1310
DMPO6   001414             266#   1147*  1310*  1311*
DMRLVL  001376             259#   1313*  1314*  1315
DMRVEC  001374             258#    765   1296*  1297*  1313
DMS100  001502             280#
DMS101  001512             285#
DMS102  001522             290#
DMS103  001532             295#
DMS104  001542             300#
DMS105  001552             305#
DMS106  001562             310#
DMS107  001572             315#
DMS110  001602             320#
DMS111  001612             325#
DMS112  001622             330#
DMS113  001632             335#
DMS114  001642             340#
DMS115  001652             345#
DMS116  001662             350#
DMS117  001672             355#
DMS200  001504             281#
DMS201  001514             286#
DMS202  001524             291#
DMS203  001534             296#
DMS204  001544             301#
DMS205  001554             306#
DMS206  001564             311#
DMS207  001574             316#
```

```
DMS210  001604            321#
DMS211  001614            326#
DMS212  001624            331#
DMS213  001634            336#
DMS214  001644            341#
DMS215  001654            346#
DMS216  001664            351#
DMS217  001674            356#
DMS300  001506            282#
DMS301  001516            287#
DMS302  001526            292#
DMS303  001536            297#
DMS304  001546            302#
DMS305  001556            307#
DMS306  001566            312#
DMS307  001576            317#
DMS310  001606            322#
DMS311  001616            327#
DMS312  001626            332#
DMS313  001636            337#
DMS314  001646            342#
DMS315  001656            347#
DMS316  001666            352#
DMS317  001676            357#
DMTLVL  001402            261#   1317*   1318*
DMTVEC  001400            260#   1315*   1316*   1317
DM.END  001700            359#   1369
DM.MAP  001500            195     278#    478     533     543     659    1287    1289    1367    1372    1607
DONE    003744            781     783*    803#   1213*
DT1     027500           3091#
DT10    027614           3091#
DT11    027632           3091#
DT2     027516           3091#
DT3     027534           3091#
DT4     027552           3091#
DT5     027564           3091#
DT7     027576           3091#
EM1     026661           3091#
EM10    027043           3091#
EM11    027070           3091#
EM12    027114           3091#
EM13    027140           3091#
EM2     026702           3091#
EM3     026736           3091#
EM4     026757           3091#
EM5     026773           3091#
EM7     027025           3091#
ERCT00  001704            366#
ERCT01  001710            369#
ERCT02  001714            372#
ERCT03  001720            375#
ERCT04  001724            378#
ERCT05  001730            381#
ERCT06  001734            384#
ERCT07  001740            387#
ERCT10  001744            390#
```

```
ERCT11  001750            393#
ERCT12  001754            396#
ERCT13  001760            399#
ERCT14  001764            402#
ERCT15  001770            405#
ERCT16  001774            408#
ERCT17  002000            411#
ERR     002710            589     615#    619
ERRCNT  001232            163#    744     771     1084*   1302*
ERRFLG  001325            202#    476*    730*    792*    1034*   1047    1061*   1120*
ERRMSG  005202           1044*   1062    1065#
ERRPC   003000            621     637#
ERTAB0  005332           1059    1094#
EXIT  = 000205             96#
EXITER  005262           1079    1084#
FLOAT   002546            582#    588
FY      002576            591#    602     606     612
HALTS   005232           1030    1076#
HILIM   004376            873*    900     918#
ICOUNT  001222            159#    790     795*
INBUF   007546            843     879    1199#
INCHAR  010064           1228    1256#
INIFLG  001324            201#    502     524     531*
INSTER= 104404            223#    894
INSTR = 104403            221#   1326    1378    1391    1400    1481    1490
INSTR2  004176            850     862#
INTTY   012344           1411    1430    1440    1453    1469    1657#
KMCM    007366            634    1184#
LIMITS  004324            889     900#
LINE    007043           1184#   1482
LOBITS  004402            875*    904     920#    921
LOCK    001220            158#    794*    811     813     1053    2158*   2208*   2226*   2239*   2265*   2282*   2294*   2319*
                         2337*   2350*   2376*   2394*   2407*   2433*   2451*   2464*   2490*   2508*   2521*   2547*   2565*
                         2578*   2604*   2622*   2635*   2662*   2680*   2693*   2720*   2738*   2751*   2778*   2796*   2809*
                         2836*   2854*   2867*   2894*   2912*   2925*   2952*   2970*   2983*
LOKFLG  001326            203#
LOLIM   004374            872*    902     917#
LPCNT   001224            160#    789*    790     793*
LSTERR  001234            164#    485*    729*   1031    1033*   1121*
MAPCK   026434           2087    2163    2213    2270    2324    2381    2438    2495    2552    2609    2667    2725    2783
                         2841    2899    2957    3080#
MASKX   001244            172#
MASTEK  006153           1055    1184#
MCRLF   005702            828     951    1051    1052    1060    1184#   1325    1387
MCSRX   006103            734    1184#
MDATA   007610            981     991    1201#
MEMLIM  001304            188#    697*
MEPASS  005743            733    1184#
MERRPC  006233           1058    1184#
MERRX   006130            740    1184#
MERR2   005771           1184#   1592
MERR3   006016            670    1184#
MESWCH  026462            508    3091#
MILK    001322            196#    479*    742    1285*   1290*   1294
MLOCK   006054            711    1184#
MNEW    006155            665    1184#
```

```
MODU    006731          1184#   1452
MPASSX  006117           738    1184#
MPFAIL  005705          1118    1184#
MQM     005676           858    1184#   1349    1424    1436    1446    1461    1475
MR      005766           720    1184#   1343
MRESET= 004000            96#
MSTCLR= 104412           235#   1123    2114    2160    2210    2267    2321    2378    2435    2492    2549    2606    2664
                        2722    2780    2838    2896    2954
MTITLE  001000           136#    506
MTSTN   006141          1056    1184#   1327
MTSTPC  006042          1184#
MVECX   006111           736    1184#
NEXT    001216           157#    796    1089    2085*   2112*   2157*   2207*   2264*   2318*   2375*   2432*   2489*   2546*
                        2603*   2661*   2719*   2777*   2835*   2893*   2951*
NOACT   007201           520    1184#   1279
NODEV   002704           574     613#
NUM     006472          1184#   1379
OK      002656           600     607#    636
ONE     001302           187#
PACT00  001702           365#
PACT01  001706           368#
PACT02  001712           371#
PACT03  001716           374#
PACT04  001722           377#
PACT05  001726           380#
PACT06  001732           383#
PACT07  001736           386#
PACT10  001742           389#
PACT11  001746           392#
PACT12  001752           395#
PACT13  001756           398#
PACT14  001762           401#
PACT15  001766           404#
PACT16  001772           407#
PACT17  001776           410#
PARAM = 104405           225#   1328    1380    1393    1402    1483    1492
PARAM1  004244           878#    895
PARBIT= 040000            96#
PARERR  004320           881     883     885     894#    901     903     905
PASCNT  001230           162#    731*    732     743     768    1301*
PERFOR= 004537            96#
PFTAB   005440          1119    1125#
POPR0 = 012600            72#   1083
POP1SP= 005726            70#
POP2SP= 022626            74#    798
PRIO    006574          1184#   1410
PS    = 177776            63#    471*    708*   1619*   1629*
PUSHR0= 010046            71#   1080
PUSH1S= 005746            69#
PUSH2S= 024646            73#
QV.FLG  001327           204#    477*    747*    787
RESREG  005230          1072    1075#
RESTAR  005360          1105    1111#
RESTRT  003544           746     750     758#
RESOS = 104407           229#   1075
RETURN  001214           156#    487*    717*    721     758*    796*    800    1089*   1092    1124    1342*   1352*   1354
```

```
ROMCLK= 104414   239# 1131 1134 1171 1176 2119 2121 2123 2131 2133 2171 2216 2218
                2229 2231 2242 2244 2272 2274 2284 2286 2296 2298 2327 2329 2340
                2342 2353 2355 2384 2386 2397 2399 2410 2412 2441 2443 2454 2456
                2467 2469 2498 2500 2511 2513 2524 2526 2555 2557 2568 2570 2581
                2583 2612 2614 2625 2627 2638 2640 2670 2672 2683 2685 2696 2698
                2728 2730 2741 2743 2754 2756 2786 2788 2799 2801 2812 2814 2844
                2846 2857 2859 2870 2872 2902 2904 2915 2917 2928 2930 2960 2962
                2973 2975 2986 2988 3005 3007 3009 3017 3025 3033 3041 3049 3051
                3053 3061
ROMDAT  026372  2220 2233 2246 2276 2288 2300 2331 2344 2357 2388 2401 2414 2445
                2458 2471 2502 2515 2528 2559 2572 2585 2616 2629 2642 2674 2687
                2700 2732 2745 2758 2790 2803 2816 2848 2861 2874 2906 2919 2932
                2964 2977 2990 3066#
ROMMAP  012376  1667# 2091 2165 3085* 3088*
ROM1    027257   507 3091#
RUN     001316   193#  480* 1283* 1284* 1291
SAVACT  001312   191#  668 1590*
SAVNUM  001314   192#  474*  745*  748* 1583*
SAVPC   001276   185#  619*  639  926* 1096
SAVR0   001260   178#  935*  940 3091
SAVR1   001262   179#  625*  934*  941
SAVR2   001264   180#  933*  942 3091
SAVR3   001266   181#  932*  943
SAVR4   001270   182#  931*  944 3091
SAVR5   001272   183#  930*  945 3091
SAVSP   001274   184#
SAV05 = 104406   227# 1035
SCOPE = 104400   215# 2100 2140 2194 2252 2306 2363 2420 2477 2534 2591 2648 2706
                2764 2822 2880 2938 2996
SCOP1 = 104401   217# 2185 2225 2238 2251 2281 2293 2305 2336 2349 2362 2393 2406
                2419 2450 2463 2476 2507 2520 2533 2564 2577 2590 2621 2634 2647
                2679 2692 2705 2737 2750 2763 2795 2808 2821 2853 2866 2879 2911
                2924 2937 2969 2982 2995
SETBR0  026316  2440 2453 2466 3014#
SETBR1  026324  2497 2510 2523 3022#
SETBR4  026332  2554 2567 2580 3030#
SETBR7  026340  2611 2624 2637 3038#
SETC    026346  2326 2339 2352 3046#
SETZ    026364  2383 2396 2409 3058#
SKIP    002642   570  573  594  597 603#
SOFTSW  010116  1226 1265#
SPACNT= 004723   961*  985  988* 1002#
SPEED   007376  1184# 1439
STACK = 001200    64#  472  687 1090 1113
STAT    001240   170#
STAT1   001366   251# 1298* 2088 2161 2211 2268 2322 2379 2436 2493 2550 2607 2665
                2723 2781 2839 2897 2955
STAT2   001370   252# 1299*
STAT3   001372   253# 1300*
STRTSW  001236   169#  510*  513*  514  516  526  528  663  709  718 1320 1344* 1374
                1603
SV05    004412   930#
SWFLG   010062   475*  822 1221* 1248* 1254#
SWMES   007232  1184# 1224
SWMES1  007242  1184# 1227
SWR     001202   143#  492*  494  498*  510  668  673  778  785  809  824 1024 1029
```

```
                    1078    1085    1087    1149    1209    1247*   3074    3086
SWREG   000176       129#    498    1209    1267
SW00  = 000001        45#    514    1374    1603
SW01  = 000002        44#    718    1320    1344
SW02  = 000004        43#
SW03  = 000010        42#    663
SW04  = 000020        41#
SW05  = 000040        40#
SW06  = 000100        39#   1149
SW07  = 000200        38#
SW08  = 000400        37#   1085
SW09  = 001000        36#    809
SW10  = 002000        35#   1087
SW11  = 004000        34#    785
SW12  = 010000        33#    824    1024
SW13  = 020000        32#   1029
SW14  = 040000        31#
SW15  = 100000        30#
TEMP    001416       271#    968    1115*   1116*   1157*   1162*   1168*   1180*
TEMP1   001246       173#    534*   1186    1615*   1616*
TEMP2   001250       174#    535*   1188
TEMP3   001252       175#    537*    563*    615     624*   1190    1383    1386    1502*   2178*   2180*   2181*   2182*
                    2183*   3091
TEMP4   001254       176#    538*   1192    1396    1399    1405    1408    1486    1489    1495    1498    3091
TEMP5   001256       177#    539*   1194    1377*   1390*   1601
TIMER = 104416       243#
TKCSR   001204       148#    845    1211    1256    1657
TKDBR   001206       149#    847     853    1214    1216    1258    1659
TLAST = 026104      1347    2997#
TPCSR   001210       150#    829     851    1026    1259    1660
TPDBR   001212       151#    831*    853*   1028*   1261*   1662*
TRPOK   004740      1014#
TSTNU   001226       161#    486*   1099    1127    1331    1338    1340    2084*   2111*   2156*   2206*   2263*   2317*
                    2374*   2431*   2488*   2545*   2602*   2660*   2718*   2776*   2834*   2892*   2950*
TST1    022404      1334    1352    2084#
TST10   023750      2375    2431#
TST11   024144      2432    2488#
TST12   024340      2489    2545#
TST13   024534      2546    2602#
TST14   024730      2603    2660#
TST15   025124      2661    2718#
TST16   025320      2719    2776#
TST17   025514      2777    2834#
TST2    022476      2085    2111#
TST20   025710      2835    2892#
TST21   026104      2893    2950#   2997
TST22 = ****** U    2951
TST3    022610      2112    2156#
TST4    023004      2157    2206#
TST5    023200      2207    2263#
TST6    023360      2264    2317#
TST7    023554      2318    2374#
TTST    003622       712*    713*    715*    716*    779#
TWOSYN= 010000        96#
TYPDAT  005216      1050    1068    1071#
TYPE  = 104402       219#    506     507     508     520     532     617     622     630     633     665     670     711
```

```
                        720     733     734     736     738     740     828     841     858     951     991    1051    1052
                       1055    1056    1058    1060    1064    1069    1118    1224    1227    1279    1325    1343    1349
                       1387    1409    1423    1428    1435    1438    1445    1451    1460    1467    1474    1592
TYPMSG  005116         1048    1051#
VEC     006553         1184#   1401
VECMAP  012066         1591    1603#
V4MAP   012400         1669#   3076    3088
V5MAP   016402         1681#   3073    3085
WHICH   012060         1389    1599#
WRDCNT  004720          959*    992*   1000#
WRKO.F  005204         1063    1066#
XBX     005010         1025    1027    1029#
XCSR    003556          735     760#
XERR    003600          741     769#
XHEAD   006240          532    1184#
XLOC    003032          590*    608*    611     642     655#
XPASS   003572          739     766#
XSTATQ  007520          541    1184#
XTSTN   005340         1057    1097#
XVEC    003564          737     763#
ZERO    001300          186#
$COD  = ****** U          1
$CRAP = 177777            1#    2070#   2073    2080#   2101#   2104    2107#   2141#   2144    2152#   2195#   2198    2202#
                       2253#   2256    2259#   2307#   2310    2313#   2364#   2367    2370#   2421#   2424    2427#   2478#
                       2481    2484#   2535#   2538    2541#   2592#   2595    2598#   2649#   2652    2656#   2707#   2710
                       2714#   2765#   2768    2772#   2823#   2826    2830#   2881#   2884    2888#   2939#   2942    2946#
$ENDAD  003532          123     504     752#   1076
$N    = 000021            1#    2070    2080    2082    2087#   2101    2107    2109    2114#   2141    2152    2154    2160
                       2161#   2195    2202    2204    2210    2211#   2253    2259    2261    2267    2268#   2307    2313
                       2315    2321    2322#   2364    2370    2372    2378    2379#   2421    2427    2429    2435    2436#
                       2478    2484    2486    2492    2493#   2535    2541    2543    2549    2550#   2592    2598    2600
                       2606    2607#   2649    2656    2658    2664    2665#   2707    2714    2716    2722    2723#   2765
                       2772    2774    2780    2781#   2823    2830    2832    2838    2839#   2881    2888    2890    2896
                       2897#   2939    2946    2948    2954    2955#   2997#
$S    = 000023            1#    2085    2087#   2112    2114#   2157    2161#   2207    2211#   2264    2268#   2318    2322#
                       2375    2379#   2432    2436#   2489    2493#   2546    2550#   2603    2607#   2661    2665#   2719
                       2723#   2777    2781#   2835    2839#   2893    2897#   2951    2955#
$Y    = 000017            1#     207#    215     217#    219#    221#    223#    225#    227#    229#    231#    233#    235#
                        237#    239#    241#    243#    245#
.     = 027776          108#     109     112#    119#    124#    127#    131#    135#    137#    189#    190#    191#    192#
                        272#    277#    279#    280#    281#    282#    284#    285#    286#    287#    289#    290#    291#
                        292#    294#    295#    296#    297#    299#    300#    301#    302#    304#    305#    306#    307#
                        309#    310#    311#    312#    314#    315#    316#    317#    319#    320#    321#    322#    324#
                        325#    326#    327#    329#    330#    331#    332#    334#    335#    336#    337#    339#    340#
                        341#    342#    344#    345#    346#    347#    349#    350#    351#    352#    354#    355#    356#
                        357#    512     522     654#    672    1107    1117    1165#   1200#   1202#   1257    1260    1281
                       1375    1419    1544    1548    1595    1626    1658    1661    3091#
.BEGIN  003162          687#
.CNVRT  004502          234     952#
.CONVR  004476          232     951#
.DATAC  005562          242    1156#
.DELAY  005446          238    1129#
.EOP    003374          728#    2951
.ERRTA  027644         1043    3091#
.HLT    004760          115    1023#
.INSTE  004164          224     858#
```

```
.INSTR  004060             222     837#
.INST1  004100             841#    861
.MSG    004102             839*    842#
.MSTCL  005476             236    1140#
.PARAM  004204             226     869#
.PFAIL  005346             113     473    1104#   1112
.RES05  004444             230     940#
.ROMCL  005514             240    1145#
.SAV05  004404             228     926#
.SCOPE  003606             216     776#
.SCOP1  003746             218     808#
.START  002002             132     471#    487     1244
.TIMER  005626             244    1167#
.TRPSR  004726             117    1011#
.TRPTA  001330             214#   1016
.TYPE   003776             220     819#
```

```
DMEND       1#    722
DMFRNT      1#
HLT        75#   2129   2139   2184   2224   2237   2250   2280   2292   2304   2335   2348   2361   2392   2405
         2418    2449   2462   2475   2506   2519   2532   2563   2576   2589   2620   2633   2646   2678   2691
         2704    2736   2749   2762   2794   2807   2820   2852   2865   2878   2910   2923   2936   2968   2981
         2994
$AUTO       1#    544
$BRRSH      1#   2101
$BUFFE      1#   1196
$COMP       1#
$CRAM       1#   2070
$CYCLE      1#   1268
$EOP        1#    722
$FINI       1#   2997
$GETPA      1#
$HEADE      1#
$JUMP       1#   2195   2253   2307   2364   2421   2478   2535   2592   2649   2707   2765   2823   2881   2939
$MARHI      1#
$MOCK       1#
$MSG        1#   1184
$PFAIL      1#   1100
$QUEST      1#   1378   1391   1400   1481   1490
$RAMCL      1#   1128
$RCLK       1#   1131   1134   1171   1176   2119   2121   2123   2130   2133   2171   2216   2218   2229   2231
         2242    2244   2272   2274   2284   2286   2296   2298   2327   2329   2340   2342   2353   2355   2384
         2386    2397   2399   2410   2412   2441   2443   2454   2456   2467   2469   2498   2500   2511   2513
         2524    2526   2555   2557   2568   2570   2581   2583   2612   2614   2625   2627   2638   2640   2670
         2672    2683   2685   2696   2698   2728   2730   2741   2743   2754   2756   2786   2788   2799   2801
         2812    2814   2844   2846   2857   2859   2870   2872   2902   2904   2915   2917   2928   2930   2960
         2962    2973   2975   2986   2988   3005   3007   3009   3017   3025   3033   3041   3049   3051   3053
         3061
$ROMNU      1#
$ROMKD      1#   2141
$SCOPE      1#    772
$SIMBC      1#
$SOFTC      1#   1204
$TRPDE      1#    215    217    219    221    223    225    227    229    231    233    235    237    239    241
          243
$TSTN       1#   2082   2109   2154   2204   2261   2315   2372   2429   2486   2543   2600   2658   2716   2774
         2832    2890   2948
$VARIA      1#    134
$XZ         1#   2070   2080   2101   2107   2141   2152   2195   2202   2253   2259   2307   2313   2364   2370
         2421    2427   2478   2484   2535   2541   2592   2598   2649   2656   2707   2714   2765   2772   2823
         2830    2881   2888   2939   2946

. ABS.  027776     000     OVR   RO    REL   GBL   D

 ERRORS DETECTED:  0

 CRLPMB,CRLPMB/SOL/CRF=CRLPMB.MAC,CRLPMB.P11
 RUN-TIME: 9 12 1 SECONDS
 RUN-TIME RATIO: 39/23=1.6
 CORE USED:  21K  (41 PAGES)
```