DMC11

BASIC W/R AND MICROPROC

MD-11-DZDMC-B

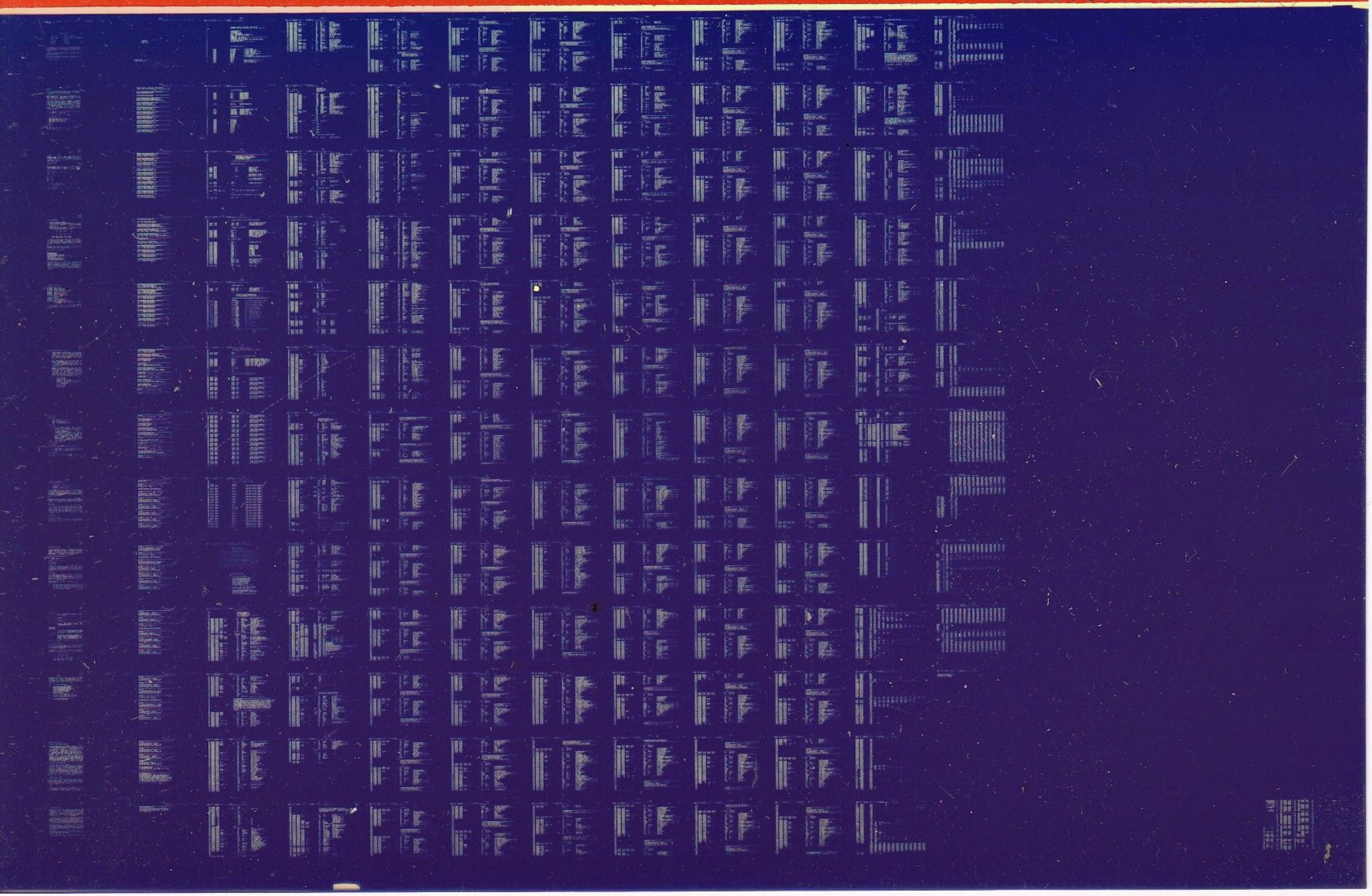
EP-DZDMC-B-DL-A

FICHE 1 OF 1

AUG 1977

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDMC-B-D
PRODUCT NAME:	BASIC W/R AND MICRO-PROCESSOR TESTS
DATE:	MAY 1977
MAINTAINER:	DIAGNOSTICS
AUTHOR:	FAY BASHAW

1. ABSTRACT

The function of the DMC11 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 DMC11 are correct in its environment.

Parameters must be set up to alert the diagnostics to the DMC11 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.

DZDMC tests the DMC11 micro-processor (M8200-YA or M8200-YB). It periorms write/read tests on the DMC unibus registers. checks the micro-processor operation, checks out Main Memory, scratch pad memory, the ALU functions as well as interrupts and NPR operation. DZDMC performs no tests on the line unit or any CROM dependent tests. It does not require a line unit to run. NOTE: This diagnostic will run on a KMC11 (M8204), however it is not advised that this diagnostic be used to check a KMC11, rather you should check a KMC11 with the KMC11 diagnostic package.

Currently there are five 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 five diagnostics are:

1. DZDMC [REV] Basic W/R and Micro-processor tests
2. DZDME [REV] DDCMP Line unit tests
3. DZDMF [REV] BITSTUFF Line unit tests
4. DZDMG [REV] Jump and CROM tests
5. DZDMH [REV] Free-running tests (Heat test tape)

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8k memory
ASR 33 (or equilivalent)
DMC11-AR (M8200-YA) or a DMC11-AL (M8200-YB)

2.2 STORAGE

Program will use all 8K 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 DMC11 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 DMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
001500	160010	145310	177777	000000
001510	160020	145320	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 two DMC11'S. 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 DMC11'S TO BE TESTED?1

0'
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
DOES MICRO-PROCESSOR HAVE CRAM?  (Y OR N)N
WHICH LINE UNIT?  IF NONE TYPE "N",  IF  M8201  TYPE  "1",  IF
M8202 TYPE "2"?1
IS THE LOOP BACK CONNECTOR ON?Y
SWITCH PAC#1 (DDCMP LINE#)?377
SWITCH PAC#2 (BM873 BOOT ADD)?377
```

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 arswered.

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/abell 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 Set: Reserved
SW 03 Set: Reselect DMC11'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 DMC11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to DMC11's active. this means if the system has four DMC11s; 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 DMC11s 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 DMC11s 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 DMC desired active.
EXAMPLE: If you have 4 DMC's but only want to run the first and the last set SWR bits 0 and 3 = 1. PRESS CONTINUE
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 'SCOPI') 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 enabled; 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 DMC11 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 DMC11'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 possibly 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 DMC11 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 DMC11 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 TTY to completly isolate problems.

7.2 OPERATING RESTRICTIONS

The first time a DMC11 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 DMC diagnostic because the STATUS TABLE is overlayed. The current parameters in the STATUS TABLE are used when SW07=1 on start up.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

DMC11(M8200)- 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 DMC11 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 DMC11'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 DZDMC CSR: 175000 VEC: 0300 PASSES: 000001
ERRORS: 000000

NOTE: The pass count and error counts are cummulitive for each DMC11 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 DMC11 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 DMC11 currently being tested. EXAMPLE: (RUN) 1302/C000000001000000 Means that DMC11 no.06 is the DMC11 now running.

DMCR00-DMCR17
DMST00-DMST17
(1500)-(1640)

These locations contain the information needed to test up to 16 (decimal) DMC11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each DMC11.

DMACTV (1306) Each bit set in this location indicates that the associated DMC11 will be tested in turn. EXAMPLE: (DMACTV) 1276/0000000000011111 means that DMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (DMACTV) 1276/0000000000010001 Means that DMC11 no. 00,04 will be tested.

DMCSR (1404) Contains the CSR of the current DMC11 under test.

8.4A 'STATUS TABLE' (1500-1640)

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 DMC11'S. the table can contain up to 16 DMC11'S. Following the map is a description of the bits for each map entry

MAP OF DMC11 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 DMC11. The PC shows where in core memory the first of the 4 words is. In the example above the first DMC'S status is in locations, 1500, 1502, 1504, and 1506. The second DMC status is located at 1510, 1512, 1514, and 1516. The information contained in each 4 word entry is defined as follows:

CSR: Contains DMC11 CSR address

STAT1: BITS 00-08 IS DMC11 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 DMC11 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 RUN FREE RUNNING TESTS ON KMC11
BIT1=0 DMC11-AR (LOW SPEED)
BIT1=1 DMC11-AL (HIGH SPEED)

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a DMC11 as follows: It starts at address 160000 and tests all address in increments of 10 up to and including address 167760. 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 626 or 16520 a DMC11 or KMC11 has been found, if not, the address is updated by 10 and the search continues. A -1 indicates a DMC11 with no CROM, a 125252 indicates a KMC11 with CRAM, a 626 indicates a DMC11-AL and a 16520 indicates a DMC11-AR. Further tests are performed at this point to determine which line unit, if any, is installed, if a loop-back connector is installed and various switch settings on the line unit. 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 DMC11's in the system will be found by the auto-sizer. If it does not find a DMC11 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 DMC11) 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 DMC11 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.

DOCUMENT
****
DZDMC .LST
****

```
6       MAINDEC-11-DZDMC-B   BASIC DMC11 CONTROLLER TEST
        COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
        ------------------------------------------------------------

1668    ************************** TEST 1 **************************
        VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
        DOES NOT CAUSE A TIME OUT TRAP

1697    ************************** TEST 2 **************************
        VERIFY THAT RUN CAN BE CLEARED

1714    ************************** TEST 3 **************************
        UNIBUS REGISTER WORD DUAL ADDRESSING TEST
        LOAD ALL REGISTERS WITH INCREMENTING PATTERN
        READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING

1756    ************************** TEST 4 **************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT0, VERIFY BIT0 WAS SET
        CLEAR BIT0, VERIFY BIT0 WAS CLEARED

1786    ************************** TEST 5 **************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT1, VERIFY BIT1 WAS SET
        CLEAR BIT1, VERIFY BIT1 WAS CLEARED

1816    ************************** TEST 6 **************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT2, VERIFY BIT2 WAS SET
        CLEAR BIT2, VERIFY BIT2 WAS CLEARED

1846    ************************** TEST 7 **************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT5, VERIFY BIT5 WAS SET
        CLEAR BIT5, VERIFY BIT5 WAS CLEARED

1876    ************************** TEST 10 *************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT6, VERIFY BIT6 WAS SET
        CLEAR BIT6, VERIFY BIT6 WAS CLEARED

1906    ************************** TEST 11 *************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT7, VERIFY BIT7 WAS SET
        CLEAR BIT7, VERIFY BIT7 WAS CLEARED

1936    ************************** TEST 12 *************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT9, VERIFY BIT9 WAS SET
        CLEAR BIT9, VERIFY BIT9 WAS CLEARED
```

```
1966    ************************** TEST 13 ****************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT11, VERIFY BIT11 WAS SET
        CLEAR BIT11, VERIFY BIT11 WAS CLEARED

1996    ************************** TEST 14 ****************************
        CONTROL STATUS REGISTER WRITE/READ TEST
        SET BIT12, VERIFY BIT12 WAS SET
        CLEAR BIT12, VERIFY BIT12 WAS CLEARED

2026    ************************** TEST 15 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT0, VERIFY BIT0 WAS SET
        CLEAR BIT0, VERIFY BIT0 WAS CLEARED

2056    ************************** TEST 16 ****************************

2057    CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT1, VERIFY BIT1 WAS SET
        CLEAR BIT1, VERIFY BIT1 WAS CLEARED

2086    ************************** TEST 17 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT2, VERIFY BIT2 WAS SET
        CLEAR BIT2, VERIFY BIT2 WAS CLEARED

2116    ************************** TEST 20 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT6, VERIFY BIT6 WAS SET
        CLEAR BIT6, VERIFY BIT6 WAS CLEARED

2146    ************************** TEST 21 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT7, VERIFY BIT7 WAS SET
        CLEAR BIT7, VERIFY BIT7 WAS CLEARED

2176    ************************** TEST 22 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT12, VERIFY BIT12 WAS SET
        CLEAR BIT12, VERIFY BIT12 WAS CLEARED

2206    ************************** TEST 23 ****************************
        CONTROL OUT REGISTER WRITE/READ TEST
        SET BIT13, VERIFY BIT13 WAS SET
        CLEAR BIT13, VERIFY BIT13 WAS CLEARED

2236    ************************** TEST 24 ****************************
        PORT4 REGISTER WRITE/READ TEST
        FLOAT A ONE THROUGH PORT4 REGISTER
        FLOAT A ZERO THROUGH PORT4 REGISTER
```

```
2279    ************************** TEST 25 ****************************
        PORT6 REGISTER WRITE/READ TEST

2281    FLOAT A ONE THROUGH PORT6 REGISTER
        FLOAT A ZERO THROUGH PORT6 REGISTER

2322    ************************** TEST 26 ****************************
        UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
        LOAD ALL REGISTERS WITH INCREMENTING PATTERN
        READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING

2364    ************************** TEST 27 ****************************
        MAINTENANCE INSTRUCTION REGISTER TEST
        VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
        AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.

2425    ************************** TEST 30 ****************************
        MAINTENANCE INSTRUCTION REGISTER TEST
        VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
        AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A MASTER RESET.

2446    ************************** TEST 31 ****************************
        MICRO PROCESSOR TEST
        LOAD DMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT

2449    VERIFY INSTRUCTION EXECUTED PROPERLY
        INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
        AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4

2473    ************************** TEST 32 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 0
        FLOAT A 0 THROUGH IBUS* REGISTER 0

2529    ************************** TEST 33 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 2
        FLOAT A 0 THROUGH IBUS* REGISTER 2

2585    ************************** TEST 34 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 4
        FLOAT A 0 THROUGH IBUS* REGISTER 4

2637    ************************** TEST 35 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 5
        FLOAT A 0 THROUGH IBUS* REGISTER 5
```

```
2689    ************************** TEST 36 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 10
        FLOAT A 0 THROUGH IBUS* REGISTER 10
        THE NPR RQ BIT (BIT 0) IS MASKED DURING THIS TEST

2746    ************************** TEST 37 ****************************
        MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS* REGISTER 11
        FLOAT A 0 THROUGH IBUS* REGISTER 11
        THE BR RQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
        (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST

2808    ************************** TEST 40 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 0
        FLOAT A 0 THROUGH IBUS REGISTER 0

2860    ************************** TEST 41 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 1
        FLOAT A 0 THROUGH IBUS REGISTER 1

2912    ************************** TEST 42 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 2
        FLOAT A 0 THROUGH IBUS REGISTER 2

2964    ************************** TEST 43 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 3
        FLOAT A 0 THROUGH IBUS REGISTER 3

3016    ************************** TEST 44 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 4
        FLOAT A 0 THROUGH IBUS REGISTER 4

3068    ************************** TEST 45 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 5
        FLOAT A 0 THROUGH IBUS REGISTER 5

3120    ************************** TEST 46 ****************************

3121    MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 6
        FLOAT A 0 THROUGH IBUS REGISTER 6
```

```
3172    ************************** TEST 47 ****************************
        MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
        FLOAT A 1 THROUGH IBUS REGISTER 7
        FLOAT A 0 THROUGH IBUS REGISTER 7

3224    ************************** TEST 50 ****************************
        MICRO PROCESSOR IBUS DUAL ADDRESS TEST
        WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
        READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING

3285    ************************** TEST 51 ****************************
        MICRO PROCESSOR BR REGISTER TEST
        FLOAT A 1 THROUGH THE BR
        FLOAT A 0 THROUGH THE BR

3336    ************************** TEST 52 ****************************
        SCRATCH PAD TEST
        FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
        FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION

3402    ************************** TEST 53 ****************************
        SCRATCH PAD DUAL ADDRESSING TEST
        WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
        READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING

3462    ************************** TEST 54 ****************************
        INTERRUPT TEST
        TEST THAT DEVICE CAN INTERRUPT TO VECTOR A

3491    ************************** TEST 55 ****************************
        INTERRUPT TEST
        TEST THAT DEVICE CAN INTERRUPT TO VECTOR B

3519    ************************** TEST 56 ****************************
        PRIORITY INTERRUPT TESTS
        SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
        THE DMC11 LEVEL,VERIFY THAT DMC11 DOES NOT INTERRUPT

3557    ************************** TEST 57 ****************************
        PRIORITY INTERRUPT TESTS
        SET PS TO ALL BR LEVELS LESS THAN THE DMC11 LEVEL
        VERIFY THAT THE DMC11 WILL INTERRUPT

3601    ************************** TEST 60 ****************************
        NPR TEST
        TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY

3634    ************************** TEST 61 ****************************
        NPR TEST
        TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
```

```
3670    ********************** TEST 62 **********************
        NPR TEST
        TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY

3702    ********************** TEST 63 **********************
        TEST OF EA BITS 16 AND 17
        DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
        VERIFY CORRECT RESULTS

3741    ********************** TEST 64 **********************
        TEST OF EA BITS 16 AND 17
        DO A DATI USING IN BA BITS 16 AND 17
        VERIFY CORRECT RESULTS

3777    ********************** TEST 65 **********************
        NPR NON-EXISTENT MEMORY TEST
        DO A DATO TO A NON-EXISTENT ADDRESS
        VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11

3812    ********************** TEST 66 **********************
        NPR NON-EXISTENT MEMORY TEST
        DO A DATI FROM A NON-EXISTENT ADDRESS
        VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11

3847    ********************** TEST 67 **********************
        NPR TEST

3849    USING DATO, NPR A BINARY COUNT (0-377 )
        FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY

3903    ********************** TEST 70 **********************
        MAIN MEMORY TEST

3905    FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS

3943    ********************** TEST 71 **********************
        MAIN MEMORY TEST
        FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS

3985    ********************** TEST 72 **********************
        MAIN MEMORY DUAL ADDRESSING TEST
        LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
        READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING

4041    ********************** TEST 73 **********************
        MAR TEST
        PERFORM DUAL ADDRESSING TEST
        USING MAR AUTO-INC FEATURE
```

```
4087    ************************** TEST 74 ****************************
        ALU C BIT TEST
        TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT

4125    ************************** TEST 75 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
        ALU FUNCTION (B)    CODE=11

4129    LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4174    ************************** TEST 76 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
        ALU FUNCTION (A)    CODE=10
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4223    ************************** TEST 77 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
        ALU FUNCTION (A OR NOTB)    CODE=12
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4272    ************************** TEST 100 ***************************
        ALU TEST
        TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
        ALU FUNCTION (A AND B)    CODE=13
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4321    ************************** TEST 101 ***************************
        ALU TEST
        TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
        ALU FUNCTION (A OR B)    CODE=14
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4370    ************************** TEST 102 ***************************
        ALU TEST
        TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
        ALU FUNCTION (A XOR B)    CODE=15
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4419    ************************** TEST 103 ***************************
        ALU TEST
        TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
        ALU FUNCTION (A PLUS B)    CODE=00
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
4468    ************************** TEST 104 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED
        ALU FUNCTION (A PLUS A PLUS C)    CODE=6
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4517    ************************** TEST 105 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
        ALU FUNCTION (A-B)    CODE=16

4521    LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4566    ************************** TEST 106 ****************************
        ALU TEST
        TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
        ALU FUNCTION (A PLUS B PLUS C)    CODE=01
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4615    ************************** TEST 107 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
        ALU FUNCTION (A-B-C)    CODE=2
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4664    ************************** TEST 110 ****************************
        ALU TEST
        TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
        ALU FUNCTION (A PLUS 1)    CODE=3
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4713    ************************** TEST 111 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
        ALU FUNCTION (A PLUS A)    CODE=5
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4762    ************************** TEST 112 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
        ALU FUNCTION (A PLUS C)    CODE=4
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
4811    ********************************* TEST 113 *******************************
        ALU TEST
        TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
        ALU FUNCTION (A-B-1)    CODE=17
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

486C    ********************************* TEST 114 *******************************
        ALU TEST
        TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
        ALU FUNCTION (A-1)    CODE=7
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4909    ********************************* TEST 115 *******************************
        ALU TEST
        TEST OF ALU FUNCTION SEL B WITH C BIT SET
        ALU FUNCTION (B)    CODE=11

4913    LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

4958    ********************************* TEST 116 *******************************
        ALU TEST
        TEST OF ALU FUNCTION SEL A WITH C BIT SET
        ALU FUNCTION (A)    CODE=10
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5007    ********************************* TEST 117 *******************************
        ALU TEST
        TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
        ALU FUNCTION (A OR NOTB)    CODE=12
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5056    ********************************* TEST 120 *******************************
        ALU TEST
        TEST OF ALU FUNCTION A AND B WITH C BIT SET
        ALU FUNCTION (A AND B)    CODE=13
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5105    ********************************* TEST 121 *******************************
        ALU TEST
        TEST OF ALU FUNCTION A OR B WITH C BIT SET
        ALU FUNCTION (A OR B)    CODE=14
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
5154    ************************** TEST 122 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A XOR B WITH C BIT SET
        ALU FUNCTION (A XOR B)    CODE=15
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION  VERIFY THE RESULTS

5203    ************************** TEST 123 ****************************
        ALU TEST
        TEST OF ALU FUNCTION ADD WITH C BIT SET
        ALU FUNCTION (A PLUS B)    CODE=00
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5252    ************************** TEST 124 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
        ALU FUNCTION (A PLUS A PLUS C)    CODE=6
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5301    ************************** TEST 125 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SUB WITH C BIT SET
        ALU FUNCTION (A-B)    CODE=16

5305    LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5350    ************************** TEST 126 ****************************
        ALU TEST
        TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
        ALU FUNCTION (A PLUS B PLUS C)    CODE=01
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5399    ************************** TEST 127 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
        ALU FUNCTION (A-B-C)    CODE=2
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5448    ************************** TEST 130 ****************************
        ALU TEST
        TEST OF ALU FUNCTION INC A WITH C BIT SET
        ALU FUNCTION (A PLUS 1)    CODE=3
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS
```

```
5497    ************************** TEST 131 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2A WITH C BIT SET
        ALU FUNCTION (A PLUS A)    CODE=5
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5546    ************************** TEST 132 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
        ALU FUNCTION (A PLUS C)    CODE=4
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5595    ************************** TEST 133 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
        ALU FUNCTION (A-B-1)    CODE=17
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5644    ************************** TEST 134 ****************************
        ALU TEST
        TEST OF ALU FUNCTION DEC A WITH C BIT SET
        ALU FUNCTION (A-1)    CODE=7
        LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
        PERFORM THE FUNCTION, VERIFY THE RESULTS

5693    ************************** TEST 135 ****************************
        TEST OF PROGRAM CLOCK BIT
        DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
        WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,

5697    AND THEN SETS SOME TIME LATER

5734    ************************** TEST 136 ****************************
        FORCE POWER FAIL TEST
        SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
        GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
        BLOCKED FROM GETTING TO THE DMC DURING THE POWER FAIL
        THIS TEST MAY HANG ON SOME PROCESSORS IF AN M9301 IS PRESENT.
        TO AVOID HANGING SW02 (POWER ON REBOOT ENABLE) ON THE M9301
        MUST BE IN THE OFF POSITION. THIS TEST WILL ALSO FAIL IF THE
        CPU POWER FAIL VECTOR IS SET TO ANY LOCATION OTHER THAN 24.
        IF THIS TEST HANGS OR FAILS DUE TO EITHER REASON ABOVE THE
        FOLLOWING PATCH MAY BE INSTALLED TO SKIP THIS TEST:

              LOC 33362      WAS 33532      SB 33724
```

```
5789    ************************** TEST 137 **************************
        MICRO-PROCESSOR NOISE TEST
        WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
        TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
        THEN GO BACK AND READ THE DATA PATERNS  TO VERIFY THAT
        READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
        DID NOT CHANGE THE DATA.
```

```
                        ;*MAINDEC-11-DZDMC-B   BASIC DMC11 CONTROLLER TEST
                        ;*COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*-----------------------------------------------------------------

                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;SWR=0  AUTOSIZE DMC11
                        ;SW07=1  USE CURRENT DMC11 PARAMETERS
                        ;SW00=1  INPUT NEW DMC11 PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-DZDMC-B   BASIC DMC11 CONTROLLER TEST"
                        ;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 DMC11 PARAMETERS, =0,AUTOSIZE DMC11
000100                  SW06=100                ;=1, HALT BEFORE CLOCKING MICRO-PROCESSOR INSTRUCTION
000040                  SW05=40
000020                  SW04=20
000010                  SW03=10                 ;RESELECT DMC11'S TO BE TESTED (ACTIVE.
000004                  SW02=4                  ;LOCK ON TEST SELECT
000002                  SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
000001                  SW00=1                  ;INPUT DMC11 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           ;PROCESSOR STACK POINTER
000006                  SP=%6           ;PROGRAM COUNTER
000007                  PC=%7

                        ;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  005336                          .PFAIL          ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT LEVEL 7
000030  004750                  .HLT                    ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT LEVEL 7
000034  004716                  .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  003522                  $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
000052                          .=52
000052  040000                  BIT14                   ;ACT-11 PROGRAM CHARACTERISTICS
                                                        ;BIT14=1 PROGRAM EXECUTION TIME
                                                        ;IS MEMORY SIZE DEPENDENT

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-DZDMC-B <377>
001025     102  051501  041511          .ASCIZ  /BASIC DMC11 CONTROLLER TEST/<377>

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
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         ;DMC11'S SELECTED ACTIVE.
001310  000001                         DMNUM:  .BLKW 1         ;OCTAL NUMBER OF DMC11'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 DMC11 ITERATIONS WILL BE
                                        .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  003576                          .SCOPE
        104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001332  003736                          .SCOP1
        104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001334  003766                          .TYPE
        104403                  INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROLTINE
001336  004050                          .INSTR
        104404                  INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001340  004154                          .INSTER
        104405                  PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001342  004174                          .PARAM
        104406                  SAVO5=TRAP+6            ;CALL TO REGISTER SAVE ROLTINE
001344  004374                          .SAVO5
        104407                  RESO5=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001346  004434                          .RESO5
        104410                  CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001350  004466                          .CONVRT
        104411                  CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001352  004472                          .CNVRT
        104412                  MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001354  005466                          .MSTCLR
        104413                  DELAY=TRAP+13           ;CALL TO DELAY
001356  005436                          .DELAY
        104414                  ROMCLK=TRAP+14          ;CALL TO CLOCk ROM ONCE
001360  005504                          .ROMCLK
        104415                  DATACLK=TRAP+15         ;CALL TO CLK DATA
001362  005552                          .DATACLK
        104416                  TIMER=TRAP+16           ;CALL TO DELAY A CLOCK TICk
001364  005618                          .TIMER

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

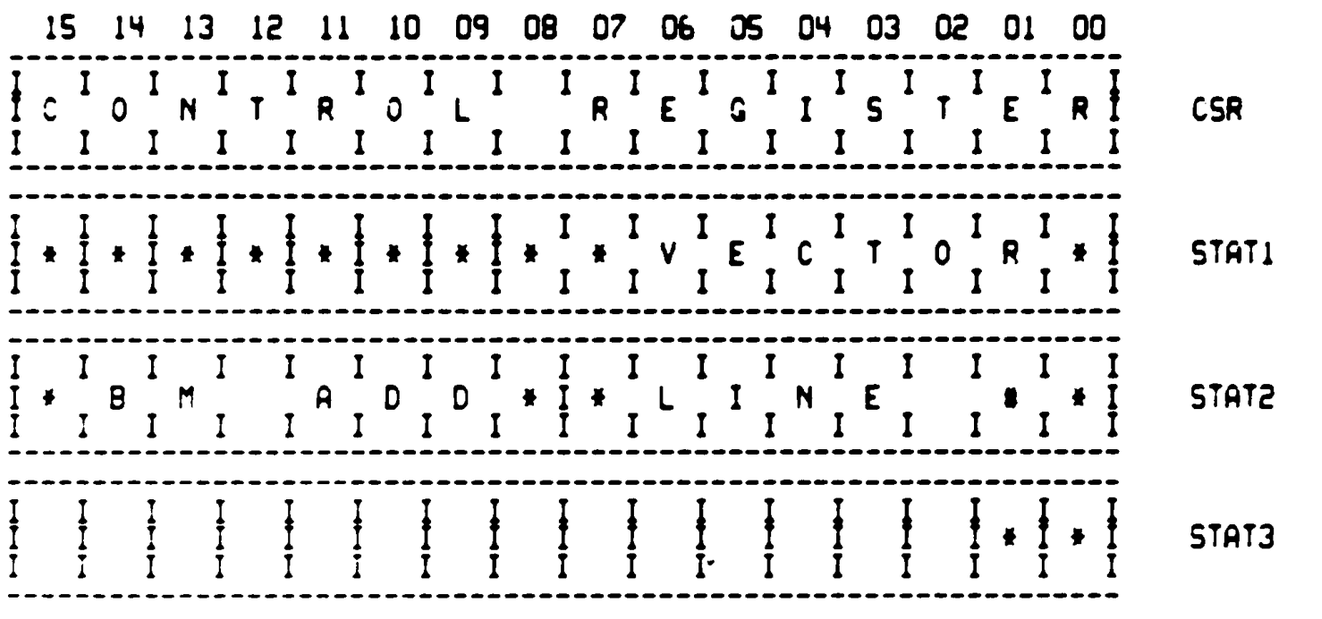
```
                                        ;DMC11 CONTROL INDICATORS FOR CURRENT DMC11 UNDER TEST
                                        ;-----------------------------------------------------

001366  000000                          STAT1:  0
001370  000000                          STAT2:  0
001372  000000                          STAT3:  0

                                        ;DMC11 VECTOR AND REGISTER INDIRECT POINTERS
                                        ;-------------------------------------------

001374  000000                          DMRVEC: 0               ;POINTER TO DMC11 RECEIVER INTERRUPT VECTOR
001376  000000                          DMRLVL: 0               ;POINTER TO DMC11 RECEIVER INTERRUPT SERVICE PS
001400  000000                          DMTVEC: 0               ;POINTER TO DMC11 TRANSMITTER INTERRUPT VECTOR
001402  000000                          DMTLVL: 0               ;POINTER TO DMC11 TRANSMITTER INTERRUPT SERVICE PS
001404  000000                          DMCSR:  0               ;POINTER TO DMC11 CONTROL STATUS REGISTER
001406  000000                          DMCSRH: 0               ;POINTER TO DMC11 CONTROL STATUS REGISTER HIGH BYTE.
001410  000000                          DMCTL:  0               ;POINTER TO DMC11 CONTOL OUT REGISTER
001412  000000                          DMP04:  0               ;POINTER TO DMC11 PORT REGISTER(SEL 4)
001414  000000                          DMP06:  0               ;POINTER TO DMC11 PORT REGISTER.SEL 6)

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

001416  000000                          TEMP:   0
        001460                          .=.+40

                                        ;DMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                        ;------------------------------------------

        001500                          .=1500
001500                                  DM.MAP:
001500  000001                          DMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 00
001502  000001                          DMS100: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 00
001504  000001                          DMS200: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 00
001506  000001                          DMS300: .BLKW   1               ;3RD STATUS WORD

001510  000001                          DMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 01
001512  000001                          DMS101: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 01
001514  000001                          DMS201: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 01
001516  000001                          DMS301: .BLKW   1               ;3RD STATUS WORD

001520  000001                          DMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 02
001522  000001                          DMS102: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 02
001524  000001                          DMS202: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 02
001526  000001                          DMS302: .BLKW   1               ;3RD STATUS WORD

001530  000001                          DMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 03
001532  000001                          DMS103: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 03
001534  000001                          DMS203: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 03
001536  000001                          DMS303: .BLKW   1               ;3RD STATUS WORD

001540  000001                          DMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 04
001542  000001                          DMS104: .BLKW   1               ;VECTOR FOR DMC11 NUMBER 04
001544  000001                          DMS204: .BLKW   1               ;DDCMP LINE# FOR DMC11 NUMBER 04
001546  000001                          DMS304: .BLKW   1               ;3RD STATUS WORD
```

```
306  001550  000001                DMCR05: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 05
307  001552  000001                DMS105: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 05
308  001554  000001                DMS205: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 05
309  001556  000001                DMS305: .BLKW   1       ;3RD STATUS WORD
310
311  001560  000001                DMCR06: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 06
312  001562  000001                DMS106: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 06
313  001564  000001                DMS206: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 05
314  001566  000001                DMS306: .BLKW   1       ;3RD STATUS WORD
315
316  001570  000001                DMCR07: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 07
317  001572  000001                DMS107: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 07
318  001574  000001                DMS207: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 07
319  001576  000001                DMS307: .BLKW   1       ;3RD STATUS WORD
320
321  001600  000001                DMCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 10
322  001602  000001                DMS110: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 10
323  001604  000001                DMS210: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 10
324  001606  000001                DMS310: .BLKW   1       ;3RD STATUS WORD
325
326  001610  000001                DMCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 11
327  001612  000001                DMS111: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 11
328  001614  000001                DMS211: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 11
329  001616  000001                DMS311: .BLKW   1       ;3RD STATUS WORD
330
331  001620  000001                DMCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 12
332  001622  000001                DMS112: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 12
333  001624  000001                DMS212: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 12
334  001626  000001                DMS312: .BLKW   1       ;3RD STATUS WORD
335
336  001630  000001                DMCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 13
337  001632  000001                DMS113: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 13
338  001634  000001                DMS213: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 13
339  001636  000001                DMS313: .BLKW   1       ;3RD STATUS WORD
340
341  001640  000001                DMCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 14
342  001642  000001                DMS114: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 14
343  001644  000001                DMS214: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 14
344  001646  000001                DMS314: .BLKW   1       ;3RD STATUS WORD
345
346  001650  000001                DMCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 15
347  001652  000001                DMS115: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 15
348  001654  000001                DMS215: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 15
349  001656  000001                DMS315: .BLKW   1       ;3RD STATUS WORD
350
351  001660  000001                DMCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 16
352  001662  000001                DMS116: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 16
353  001664  000001                DMS216: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 16
354  001666  000001                DMS316: .BLKW   1       ;3RD STATUS WORD
355
356  001670  000001                DMCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 17
357  001672  000001                DMS117: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 17
358  001674  000001                DMS217: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 17
359  001676  000001                DMS317: .BLKW   1       ;3RD STATUS WORD
360
361  001700  000000                DM.END: 000000
```

```
                                        ;DMC11 PASS COUNT AND ERROR COUNT TABLE
                                        ;--------------------------------------

001702                                  CNT.MAP:
001702  000000                          PACT00: 0               ;PASS COUNT FOR DMC11 NUMBER 00
001704  000000                          ERCT00: 0               ;ERROR COUNT FOR DMC11 NUMBER 00

001706  000000                          PACT01: 0               ;PASS COUNT FOR DMC11 NUMBER 01
001710  000000                          ERCT01: 0               ;ERROR COUNT FOR DMC11 NUMBER 01

001712  000000                          PACT02: 0               ;PASS COUNT FOR DMC11 NUMBER 02
001714  000000                          ERCT02: 0               ;ERROR COUNT FOR DMC11 NUMBER 02

001716  000000                          PACT03: 0               ;PASS COUNT FOR DMC11 NUMBER 03
001720  000000                          ERCT03: 0               ;ERROR COUNT FOR DMC11 NUMBER 03

001722  000000                          PACT04: 0               ;PASS COUNT FOR DMC11 NUMBER 04
001724  000000                          ERCT04: 0               ;ERROR COUNT FOR DMC11 NUMBER 04

001726  000000                          PACT05: 0               ;PASS COUNT FOR DMC11 NUMBER 05
001730  000000                          ERCT05: 0               ;ERROR COUNT FOR DMC11 NUMBER 05

001732  000000                          PACT06: 0               ;PASS COUNT FOR DMC11 NUMBER 06
001734  000000                          ERCT06: 0               ;ERROR COUNT FOR DMC11 NUMBER 06

001736  000000                          PACT07: 0               ;PASS COUNT FOR DMC11 NUMBER 07
001740  000000                          ERCT07: 0               ;ERROR COUNT FOR DMC11 NUMBER 07

001742  000000                          PACT10: 0               ;PASS COUNT FOR DMC11 NUMBER 10
001744  000000                          ERCT10: 0               ;ERROR COUNT FOR DMC11 NUMBER 10

001746  000000                          PACT11: 0               ;PASS COUNT FOR DMC11 NUMBER 11
001750  000000                          ERCT11: 0               ;ERROR COUNT FOR DMC11 NUMBER 11

001752  000000                          PACT12: 0               ;PASS COUNT FOR DMC11 NUMBER 12
001754  000000                          ERCT12: 0               ;ERROR COUNT FOR DMC11 NUMBER 12

001756  000000                          PACT13: 0               ;PASS COUNT FOR DMC11 NUMBER 13
001760  000000                          ERCT13: 0               ;ERROR COUNT FOR DMC11 NUMBER 13

001762  000000                          PACT14: 0               ;PASS COUNT FOR DMC11 NUMBER 14
001764  000000                          ERCT14: 0               ;ERROR COUNT FOR DMC11 NUMBER 14

001766  000000                          PACT15: 0               ;PASS COUNT FOR DMC11 NUMBER 15
001770  000000                          ERCT15: 0               ;ERROR COUNT FOR DMC11 NUMBER 15

001772  000000                          PACT16: 0               ;PASS COUNT FOR DMC11 NUMBER 16
001774  000000                          ERCT16: 0               ;ERROR COUNT FOR DMC11 NUMBER 16

001776  000000                          PACT17: 0               ;PASS COUNT FOR DMC11 NUMBER 17
002000  000000                          ERCT17: 0               ;ERROR COUNT FOR DMC11 NUMBER 17
```

415

FORMAT OF STATUS TABLE



DEFINITION OF FORMAT

CSR: CONTAINS DMC11 CSR ADDRESS

STAT1: BITS 00-08 IS DMC11 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 DMC11 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
(MUST BE SET TO A ONE MANUALLY [PROGRAM DZDMI ONLY])
KMC MUST HAVE MICRO-CODE WRITTEN FROM RUNNING
DZDMG TEST 2 FIRST
BIT1=1 DMC11-AL LOCAL HIGH SPEED MICRO-CODE
BIT1=0 DMC11-AR REMOTE LOW SPEED MICRO-CODE

```
                                                        ;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  005336  000024          MOV     #.PFAIL,@#24    ;SET UP POWER FAIL VECTOR
002022  013737  001310  001314          MOV     DMNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002030  005037  010016                  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$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002170  012737  000176  001202          MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002176  012737  000174  001200          MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002204  012637  000004          7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
002210  012637  000006                  MOV     (SP)+,@#6       ;
002214  105737  001324                  TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002220  001006                          BNE     20$             ;BR IF YES
002222  022737  003522  000042          CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002230  001402                          BEQ     20$
002232  104402  001000                  TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002236  004737  007606          20$:    JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
002242  017737  176734  001238          MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002250  005737  000042                  TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002254  001402                          BEQ     .+6             ;BR IF NO
002256  005037  001236                  CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002262  032737  000001  001236          BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002270  001012                          BNE     17$             ;BR IF SW00=1
002272  105737  001236                  TSTB    STRTSW          ;BIT7=1??
002276  100007                          BPL     17$             ;BR IF SW07=0
002300  005737  001306                  TST     DMACTV          ;ARE ANY DEVICES SELECTED?
002304  001006                          BNE     16$             ;BR IF YES
002306  104402  007154                  TYPE,   NOACT           ;NO DEVICES SELECTED.
```

```
002312  000000                          HALT                    ;STOP THE SHOW
002314  000776                          BR      .-2             ;DISQUALIFY CONTINUE SWITCH
002316  004737  010512          17$:    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
002322  105737  001324          16$:    TSTB    INIFLG          ;FIRST TIME?
002326  001410                          BEQ     21$             ;BR IF YES
002330  105737  001236                  TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
002334  100431                          BMI     1$
002336  032737  000006  001236          BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
002344  001403                          BEQ     24$             ;IF NO THEN TYPE STATUS
002346  000424                          BR      1$              ;IF YES DO NOT TYPE STATUS
002350  005137  001324          21$:    COM     INIFLG          ;SET FLAG
002354  104402  006224          24$:    TYPE    ,XHEAD          ;TYPE HEADER
002360  012704  001500                  MOV     #DM.MAP,R4      ;SET POINTER
002364  010437  001246          5$:     MOV     R4,TEMP1        ;SET ADDRESS
002370  012437  001250                  MOV     (R4)+,TEMP2     ;SET CSR
002374  001411                          BEQ     1$              ;ALL DONE IF ZERO
002376  012437  001252                  MOV     (R4)+,TEMP3     ;SET STAT1
002402  012437  001254                  MOV     (R4)+,TEMP4     ;SET STAT2
002406  012437  001256                  MOV     (R4)+,TEMP5     ;SET STAT3
002412  104410                          CONVRT                  ;TYPE OUT STATUS MAP
002414  007454                          XSTATQ                  ;
002416  000762                          BR      5$
002420  012700  001500          1$:     MOV     #DM.MAP,R0      ;R0 POINTS TO STATUS TABLE

                                ;;**************************************************************************
                                ;;*AUTO SIZE TEST
                                ;;*THIS TEST VERIFYS THAT THE DMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
                                ;;*ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
                                ;;*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC .
                                ;;*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE DMC11, THE FIRST
                                ;;*DMC11 ADDRESS IS 760070, KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
                                ;;*ADDRESS 760000. THIS TEST MAY REQUIRE 2 OR MORE ATTEMPTS TO GET THE
                                ;;*RIGHT ADDRESSES. AFTER YOU HAVE CHANGED THE ADDRESS TO WHAT IT TOLD
                                ;;*YOU THE FIRST TIME, IT MAY COME BACK AND TELL YOU A DIFFERENT ADDRESS
                                ;;*THE NEXT TIME YOU RUN IT. PLEASE HAVE PATIENCE, THE FINAL ADDRESS
                                ;;*WILL BE CORRECT (AS LONG AS ALL DEVICES IN FRONT OF THE DMC'S ARE
                                ;;*CORRECT).
                                ;;**************************************************************************

002424  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
002430  013746  000006                  MOV     @#6,-(SP)       ;SAVE LOC 6
002434  005037  000006                  CLR     @#6             ;CLEAR VEC+2
002440  005037  001252                  CLR     TEMP3           ;CLEAR FLAG
002444  005005                          CLR     R5              ;R5=0=DMC, R5=-1=KMC
002446  011037  001404          AUSTRT: MOV     (R0),DMCSR      ;GET NEXT DMC CSR
002452  001564                          BEQ     AUDONE          ;BR IF DONE
002454  005705                          TST     R5              ;DMC OR KMC?
002456  001005                          BNE     1$              ;BR IF KMC
002460  032760  100000  000002          BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
002466  001061                          BNE     SKIP            ;SKIP IF NOT DMC
002470  000404                          BR      2$              ;ITS A DMC SO CONTINUE
002472  032760  100000  000002  1$:     BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
002500  001454                          BEQ     SKIP            ;SKIP IF NOT KMC
002502  012737  002674  000004  2$:     MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
002510  005705                          TST     R5              ;DMC OR KMC?
002512  001003                          BNE     3$              ;BR IF KMC
```

```
582  002514  012703  000006                        MOV     #6,R3           ;R3 IS COUNT OF DEVICES BEFORE DMC
583  002520  000402                                BR      4$              ;GO ON
584  002522  012703  000010              3$:       MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
585  002526  012702  003010              4$:       MOV     #DEVTAB,R2      ;R2 IS DEVICE TABLE PONTER
586  002532  012701  160010                        MOV     #160010,R1      ;START WITH ADDRESS 160010
587  002536  005711                      FLOAT:    TST     (R1)            ;CHECK ADDRESS IN R1
588  002540  111204                                MOVB    (R2),R4         ;IF NO TIMEOUT, GET NEXT ADDRESS
589  002542  060401                                ADD     R4,R1           ;IN R1
590  002544  005201                                INC     R1              ;
591  002546  040401                                BIC     R4,R1           ;
592  002550  005703                                TST     R3              ;ANY MORE DEVICES TO CHECK FOR?
593  002552  001371                                BNE     FLOAT           ;BR IF YES
594  002554  012737  002700  000004                MOV     #ERR,@#4        ;OK ONLY DMC'S ARE LEFT, SET UP FOR TIMEOUT
595  002562  010137  003022                        MOV     R1,XLOC         ;SAVE FIRST DMC/KMC ADDRESS
596  002566  005705                      FY:       TST     R5              ;DMC OR KMC?
597  002570  001005                                BNE     1$              ;BR IF KMC
598  002572  032760  100000  000002                BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
599  002600  001014                                BNE     SKIP            ;SKIP IF NOT DMC
600  002602  000404                                BR      2$              ;ITS A DMC SO CONTINUE
601  002604  032760  100000  000002              1$:       BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
602  002612  001407                                BEQ     SKIP            ;SKIP IF NOT KMC
603  002614  005711                              2$:       TST     (R1)            ;CHECK DMC ADDRESS
604  002616  020137  001404                                CMP     R1,DMCSR        ;DOES IT MATCH
605  002622  001411                                BEQ     OK              ;BR IF YES
606  002624  062701  000010                        ADD     #10,R1          ;GET NEXT DMC ADDRESS
607  002630  000756                                BR      FY              ;DO IT AGAIN
608  002632  062700  000010              SKIP:     ADD     #10,R0          ;SKIP TO NEXT CSR IN TABLE
609  002636  011037  001404                        MOV     (R0),DMCSR      ;GET NEXT CSR
610  002642  001470                                BEQ     AUDONE          ;BR IF DONE
611  002644  000750                                BR      FY              ;ELSE CONTINUE
612  002646  062700  000010              OK:       ADD     #10,R0          ;SKIP TO NEXT DMC CSR
613  002652  062737  000010  003022                ADD     #10,XLOC        ;UPDATE EXPECTED DMC/KMC ADDRESS
614  002660  011037  001404                        MOV     (R0),DMCSR      ;GET NEXT DMC/KMC CSR
615  002664  001457                                BEQ     AUDONE          ;BR IF DONE
616  002666  013701  003022                        MOV     XLOC,R1         ;GET EXPECTED DMC/KMC ADDRESS
617  002672  000735                                BR      FY              ;CONTINUE
618  002674  122243                      NODEV:    CMPB    (R2)+,-(R3)     ;ON TIMEOUT, INC R2, DEC R3
619  002676  000002                                RTI                     ;RETURN
620  002700  005737  001252              ERR:      TST     TEMP3           ;CHECK FLAG IF = 0 TYPE HEADER
621  002704  001014                                BNE     1$              ;SKIP HEADER
622  002706  104402                                TYPE                    ;TYPEOUT HEADER MESSAGE
623  002710  007223                                CONERR                  ;CONFIGURATION ERROR!!!!
624  002712  012737  002700  001276                MOV     #ERR,SAVPC      ;SAVE PC FOR TYPEOUT
625  002720  104411                                CNVRT                   ;TYPE OUT ERROR PC
626  002722  002770                                ERRPC                   ;
627  002724  104402                                TYPE                    ;TYPE REST OF HEADER
628  002726  007277                                CNERR                   ;
629  002730  012737  177777  001252                MOV     #-1,TEMP3       ;SET FLAG SO IT ONLY GETS TYPED ONCE
630  002736  010137  001262              1$:       MOV     R1,SAVR1        ;SAVE R1 FOR TYPEOUT
631  002742  104410                                CONVRT
632  002744  002776                                CONTAB                  ;TYPE CSR VALUES
633  002746  005705                                TST     R5              ;DMC OR KMC ?
634  002750  001003                                BNE     3$              ;BR IF KMC
635  002752  104402                                TYPE
636  002754  007320                                DMCM
637  002756  000402                                BR      4$              ;CONTINUE
```

```
   638  002760  104402                        3$:     TYPE
   639  002762  007330                                KMCM
   640  002764  022626                        4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
   641  002766  000727                                BR      OK              ;BR TO GET OUT
   642  002770  000001                        ERRPC:  1
   643  002772     006      002                        .BYTE   6,2
   644  002774  001276                                SAVPC
   645  002776  000002                        CONTAB: 2
   646  003000     006      004                        .BYTE   6,4
   647  003002  003022                                XLOC
   648  003004     006      002                        .BYTE   6,2
   649  003006  001404                                DMCSR
   650  003010     007                         DEVTAB: .BYTE   7               ;DJ
   651  003011     017                                 .BYTE   17              ;DH
   652  003012     007                                 .BYTE   7               ;DQ
   653  003013     007                                 .BYTE   7               ;DU
   654  003014     007                                 .BYTE   7               ;DUP
   655  003015     007                                 .BYTE   7               ;LK
   656  003016     007                                 .BYTE   7               ;DMC
   657  003017     007                                 .BYTE   7               ;DZ
   658  003020     007                                 .BYTE   7               ;KMC
   659          003022                        .EVEN
   660  003022  000000                        XLOC:   0
   661  003024  005705                        AUDONE: TST     R5              ;DMC?
   662  003026  001005                                BNE     1$              ;BR IF KMC AND ALL DONE
   663  003030  012705  177777                        MOV     #-1,R5          ;SET R5 TO -1 (KMC)
   664  003034  012700  001500                        MOV     #DM.MAP,R0      ;RESET R0 TO START OF TABLE
   665  003040  000602                                BR      AUSTRT          ;GO DO KMC'S
   666  003042  012637  000006                1$:     MOV     (SP)+,@#6       ;RESTORE LOC 6
   667  003046  012637  000004                        MOV     (SP)+,@#4       ;RESTORE LOC 4
   668  003052  032737  000010  001236                BIT     #SW03,STRTSW    ;SELECT SPECIFIC DEVICES??
   669  003060  001422                                BEQ     3$              ;BR IF NO.
   670  003062  104402  006144                        TYPE    ,MNEW           ;TYPE THE MESSAGE.
   671  003066  005000                                CLR     R0              ;ZERO DATA LIGHTS
   672  003070  000000                                HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
   673  003072  027737  176104  001312                CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
   674  003100  101404                                BLOS    2$              ;BR IF NUMBER IS OK.
   675  003102  104402  006005                        TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
   676  003106  000000                                HALT                    ;STOP EVERY THING.
   677  003110  000776                                BR      .-2             ;RESTART THE PROGRAM AGAIN.
   678  003112  017737  176064  001306        2$:     MOV     @SWR,DMACTV     ;GET NEW DEVICE PATTERN
   679  003120  013700  001306                        MOV     DMACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
   680  003124  000000                                HALT                    ;CONTINUE DYNAMIC SWITCHES.
   681  003126  012700  000300                3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
   682  003132  012701  000302                        MOV     #302,R1         ;VECTOR AREA. 300-776
   683  003136  010120                        4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
   684  003140  005021                                CLR     (R1)+           ;IN VECTOR AREA.
   685  003142  022021                                CMP     (R0)+,(R1)+     ;POP POINTERS
   686  003144  022700  001000                        CMP     #1000,R0        ;ALL DONE??
   687  003150  001372                                BNE     4$              ;BR IF NO.
   688
   689                                        ;TEST START AND RESTART
   690                                        ;----------------------
   691
   692  003152  012706  001200                .BEGIN: MOV     #STACK,SP       ;SET UP STACK
   693  003156  013746  000006                        MOV     @#6,-(SP)       ;SAVE LOC 6
```

```
694  003162  013746  000004                    MOV     @#4,-(SP)       ;SAVE LOC 4
695  003166  005000                            CLR     R0              ;START AT 0
696  003170  012737  003234  000004            MOV     #2$,@#4         ;SET UP FOR TIME OUT
697  003176  005037  000006                    CLR     @#6             ;TO AUTOSIZE MEMORY
698  003202  005720                    6$:     TST     (R0)+           ;CHECK ADDRESS IN R0
699  003204  022700  157776                    CMP     #157776,R0      ;IS IT AT LEAST 28K
700  003210  001374                            BNE     6$              ;BR IF NO
701  003212  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
702  003216  010037  001304            7$:     MOV     R0,MEMLIM       ;STORE MEMORY LIMIT
703  003222  012637  000004                    MOV     (SP)+,@#4       ;RESTORE LOC 4
704  003226  012637  000006                    MOV     (SP)+,@#6       ;RESTORE LOC 6
705  003232  000413                            BR      10$             ;CONTINUE
706  003234  022626                    2$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
707  003236  162700  000004                    SUB     #4,R0           ;GET LAST GOOD ADDRESS
708  003242  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
709  003246  022700  030000                    CMP     #30000,R0       ;IS IT 8K?
710  003252  001361                            BNE     7$              ;BR IF NO
711  003254  012700  037400                    MOV     #37400,R0       ;IF 8K DON'T SAVE 2K
712  003260  000756                            BR      7$              ;
713  003262  012737  000340  177776    10$:    MOV     #340,PS         ;LOCK OUT INTERRUPTS
714  003270  032737  000004  001236            BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
715  003276  001411                            BEQ     1$              ;BR IF NO LOCK DESIRED.
716  003300  104402  006043                    TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
717  003304  012737  000240  003612            MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
718  003312  012737  000240  003614            MOV     #NOP,TTST+2     ;SET UP TO LOCK
719  003320  000406                            BR      3$              ;CONTINUE ALONG.
720  003322  013737  003730  003612    1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
721  003330  013737  003732  003614            MOV     BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
722  003336  012737  010060  001214    3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DE.ICE TO TEST
723  003344  032737  000002  001236    4$:     BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED?
724  003352  001002                            BNE     5$              ;BR IF YES
725  003354  104402  005755                    TYPE    ,MR             ;TYPE R
726  003360  000177  175630            5$:     JMP     @RETURN         ;START TESTING
```

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

003364  000005                                  .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
003366  005037  001234                                  CLR     LSTERR          ;CLEAR LAST ERROR PC
003372  105037  001325                                  CLRB    ERRFLG          ;CLEAR ERROR FLAG
003376  005237  001230                                  INC     PASCNT          ;UPDATE PASS COUNT
003402  013777  001230  175570                         MOV     PASCNT,@DISPLAY ;DISPLAY PASS COUNT
003410  104402  005733                                  TYPE    ,MEPASS         ;TYPE END PASS
003414  104402  006072                                  TYPE    ,MCSRX          ;TYPE CSR
003420  104411  003546                                  CNVRT   ,XCSR           ;SHOW IT
003424  104402  006100                                  TYPE    ,MVECX          ;TYPE VECTOR
003430  104411  003554                                  CNVRT   ,XVEC           ;SHOW IT
003434  104402  006106                                  TYPE    ,MPASSX         ;TYPE PASSES
003440  104411  003562                                  CNVRT   ,XPASS          ;SHOW IT
003444  104402  006117                                  TYPE    ,MERRX          ;TYPE ERRORS
003450  104411  003570                                  CNVRT   ,XERR           ;SHOW IT
003454  013700  001322                                  MOV     MILK,R0         ;GET POINTER TO PASS COUNT
003460  013720  001230                                  MOV     PASCNT,(R0)+    ;STORE PASS COUNT FOR THIS DMC11
003464  013720  001232                                  MOV     ERRCNT,(R0)+    ;STORE ERROR COUNT FOR THIS DMC11
003470  005337  001314                                  DEC     SAVNUM          ;ARE ALL DEVICES TESTED?
003474  001017                                          BNE     RESTRT          ;BR IF NO.
003476  112737  000377  001327                         MOVB    #377,QV.FLG     ;SET THE QUICK VERIFY FLAG.
003504  013737  001310  001314                         MOV     DMNUM,SAVNUM    ;RESTORE THE COUNT
003512  013701  000042                                  MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
003516  001406                                          BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
003520  000005                                          RESET                   ;STOP THE SHOW--CLEAR THE WORLD
003522                                          $ENDAD:
003522  004711                                          JSR     PC,(R1)
003524  000240                                          NOP
003526  000240                                          NOP
003530  000240                                          NOP
003532  000240                                          NOP
003534  012737  010060  001214                 RESTRT: MOV     #CYCLE,RETURN
003542  000137  010060                                  JMP     CYCLE
003546  000001                                  XCSR:   1
003550     006      002                                 .BYTE   6,2
003552  001404                                          DMCSR
003554  000001                                  XVEC:   1
003556     004      002                                 .BYTE   4,2
003560  001374                                          DMRVEC
003562  000001                                  XPASS:  1
003564     006      002                                 .BYTE   6,2
003566  001230                                          PASCNT
003570  000001                                  XERR:   1
003572     006      002                                 .BYTE   6,2
003574  001232                                          ERRCNT

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

003576  004737  007606                          .SCOPE: JSR     PC,CKSWR        ;CKECK FOR SOFT SWR
003602  010016                                          MOV     R0,(SP)         ;SAVE R0 ON THE STACK
```

```
 783  003604  032777  040000  175370          BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
 784  003612  001407                  TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
 785  003614  000437                          BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
 786  003616  005737  003734                  TST     DONE            ;WAS TKCSR DONE SET?
 787  003622  001434                          BEQ     3$              ;BR IF NO (LOCKED ON TEST)
 788  003624  005037  003734                  CLR     DONE            ;YES, CLEAR FLAG
 789  003630  000415                          BR      2$              ;GO TO NEXT TEST
 790  003632  032777  004000  175342  1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
 791  003640  001011                          BNE     2$              ;BR IF YES
 792  003642  105737  001327                  TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
 793  003646  001406                          BEQ     2$              ;BR IF QUICK PASS.
 794  003650  005237  001224                  INC     LPCNT           ;UPDATE ITERATION COUNTER
 795  003654  023737  001224  001222          CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
 796  003662  101414                          BLOS    3$              ;BR IF NOT YET
 797  003664  105037  001325          2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
 798  003670  005037  001224                  CLR     LPCNT           ;START ICOUNTER AT 0
 799  003674  005037  001220                  CLR     LOCK
 800  003700  012737  000020  001222          MOV     #20,ICOUNT      ;RESET ITERATIONS
 801  003706  013737  001216  001214          MOV     NEXT,RETURN     ;GET NEXT TEST
 802  003714  011600                  3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
 803  003716  022626                          POP2SP                  ;FAKE AN "RTI"
 804  003720  013701  001404                  MOV     DMCSR,R1        ;R1 CONTAINS BASE DMC ADDRESS
 805  003724  000177  175264                  JMP     @RETURN         ;GO DO THE TEST
 806  003730  001407                  BRW:    1407
 807  003732  000437                  BRX:    437
 808  003734  000000                  DONE:   0
 809
 810                                          ;CHECK FOR FREEZE ON CURRENT DATA
 811                                          ;-------------------------------
 812
 813  003736  004737  007606          .SCOP1: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
 814  003742  032777  001000  175232          BIT     #SW09,@SWR      ;IS SW09=1(SET)?
 815  003750  001405                          BEQ     1$              ;BR IF NOT SET.
 816  003752  005737  001220                  TST     LOCK
 817  003756  001402                          BEQ     1$
 818  003760  013716  001220                  MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
 819  003764  000002                  1$:     RTI                     ;GO BACK.
 820
 821                                          ;TELETYPE OUTPUT ROUTINE
 822                                          ;-----------------------
 823
 824  003766  010546                  .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
 825  003770  017605  000002                  MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
 826  003774  062766  000002  000002          ADD     #2,2(SP)        ;POP OVER ADDRESS.
 827  004002  005737  010016          4$:     TST     SWFLG           ;SOFT SWR MESSAGE?
 828  004006  001004                          BNE     1$              ;IF YES TYPE IT OUT REGARDLESS OF SW12
 829  004010  032777  010000  175164          BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
 830  004016  001012                          BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
 831  004020  105715                  1$:     TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
 832  004022  100002                          BPL     2$              ;BR IF NUMBER IS PLUS
 833  004024  104402  005672                  TYPE    ,MCRLF          ;TYPE A CR/LF!
 834  004030  105777  175154          2$:     TSTB    @TPCSR          ;TTY READY?
 835  004034  100375                          BPL     2$              ;BR IF NO.
 836  004036  112577  175150                  MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
 837  004042  001357                          BNE     4$              ;IF NOT ZERO KEEP PRINTING!
 838  004044  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
                                        MOV     @4(SP),.MSG
                                        ADD     #2,4(SP)
                                .INST1: TYPE
                                .MSG:   0
                                        MOV     #INBUF,R4
                                        MOV     #7,R3
                                1$:     TSTB    @TKCSR
                                        BPL     1$
                                        MOVB    @TKDBR,(R4)+
                                        BICB    #200,(R4)
                                        CMPB    (R4)+,#15
                                        BEQ     INSTR2
                                2$:     TSTB    @TPCSR
                                        BPL     2$
                                        MOV     @TKDBR,@TPDBR
                                        DEC     R3
                                        BNE     1$
                                        MOV     (SP)+,R4
                                        MOV     (SP)+,R3
                                .INSTE: TYPE    .MQM
                                        MOV     R3,-(SP)
                                        MOV     R4,-(SP)
                                        BR      .INST1
                                INSTR2: MOV     (SP)+,R4        ;RESTORE R4
                                        MOV     (SP)+,R3        ;RESTORE R3
                                        RTI

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

                                .PARAM: MOV     R5,-(SP)
                                        MOV     R4,-(SP)
                                        MOV     4(SP),R5
                                        MOV     (R5)+,LOLIM
                                        MOV     (R5)+,HILIM
                                        MOV     (R5)+,DEVADR
                                        MOVB    (R5)+,LOBITS
                                        MOVB    (R5)+,ADRCNT
                                        MOV     R5,4(SP)
                                PARAM1: CLR     R5
                                        MOV     #INBUF,R4
                                        CMPB    #15,(R4)
                                        BEQ     PARERR
                                1$:     CMPB    (R4),#60
                                        BLT     PARERR
                                        CMPB    (R4),#67
                                        BGT     PARERR
                                        BICB    #60,(R4)
                                        BISB    (R4)+,R5
                                        CMPB    #15,(R4)
                                        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
                                                ;-------------------------------------

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

                                                ;SAVE R0-R5

                                        SVOS:   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

                                        .RESOS: 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
```

```
  951  004464  000002                          RTI                     ;LEAVE
  952
  953                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
  954                                          ;-------------------------------------------------------
   55
  956  004466  104402  005672          .CONVR: TYPE    ,MCRLF
  957  004472  010046                  .CNVRT: MOV     R0,-(SP)
  958  004474  010146                          MOV     R1,-(SP)
  959  004476  010346                          MOV     R3,-(SP)
  960  004500  010446                          MOV     R4,-(SP)
  961  004502  010546                          MOV     R5,-(SP)
  962  004504  017601  000012                  MOV     @12(SP),R1
  963  004510  062766  000002  000012          ADD     #2,12(SP)
  964  004516  012137  004710                  MOV     (R1)+,WRDCNT
  965  004522  112137  004712          1$:     MOVB    (R1)+,CHRCNT
  966  004526  112137  004713                  MOVB    (R1)+,SPACNT
  967  004532  013137  004714                  MOV     @(R1)+,BINWRD
  968  004536  122737  000003  004712          CMPB    #3,CHRCNT
  969  004544  001003                          BNE     2$
  970  004546  042737  177400  004714          BIC     #177400,BINWRD
  971  004554  013704  004714          2$:     MOV     BINWRD,R4
  972  004560  113705  004712                  MOVB    CHRCNT,R5
  973  004564  012700  001416                  MOV     #TEMP,R0
  974  004570  010403                  3$:     MOV     R4,R3
  975  004572  042703  177770                  BIC     #177770,R3
  976  004576  062703  000060                  ADD     #060,R3
  977  004602  110320                          MOVB    R3,(R0)+
  978  004604  000241                          CLC
  979  004606  006004                          ROR     R4
  980  004610  000241                          CLC
  981  004612  006004                          ROR     R4
  982  004614  000241                          CLC
  983  004616  006004                          ROR     R4
  984  004620  005305                          DEC     R5
  985  004622  001362                          BNE     3$
  986  004624  012703  007544                  MOV     #MDATA,R3
  987  004630  114023                  4$:     MOVB    -(R0),(R3)+
  988  004632  105337  004712                  DECB    CHRCNT
  989  004636  001374                          BNE     4$
  990  004640  105737  004713                  TSTB    SPACNT
  991  004644  001405                          BEQ     6$
  992  004646  112723  000040          5$:     MOVB    #040,(R3)+
  993  004652  105337  004713                  DECB    SPACNT
  994  004656  001373                          BNE     5$
  995  004660  105013                  6$:     CLRB    (R3)
  996  004662  104402  007544                  TYPE    ,MDATA
  997  004666  005337  004710                  DEC     WRDCNT
  998  004672  001313                          BNE     1$
  999  004674  012605                          MOV     (SP)+,R5
 1000  004676  012604                          MOV     (SP)+,R4
 1001  004700  012603                          MOV     (SP)+,R3
 1002  004702  012601                          MOV     (SP)+,R1
 1003  004704  012600                          MOV     (SP)+,R0
 1004  004706  000002                          RTI
 1005  004710  000000          WRDCNT: 0
 1006  004712  000000          CHRCNT: 0
```

```
                                        SPACNT=CHRCNT+1
                                        BINWRD: 0


                                                ;TRAP DISPATCH SERVICE
                                                ;ARGUMENT OF TRAP IS EXTRACTED
                                                ;AND USED AS OFFSET TO OBTAIN POINTER
                                                ;TO SELECTED SUBROUTINE

                                        .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
                                                SUB     #2,(SP)         ;=PC OF TRAP
                                                MOV     @(SP),(SP)      ;GET TRP
                                        TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
                                                BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
                                                ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
                                                MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
                                                JMP     @(SP)+          ;GO TO SUBRCUTINE

                                                ;ERROR HANDLER
                                                ;-------------

                                        .HLT:   JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
                                                BIT     #SW12,@SWR      ;BELL ON ERROR?
                                                BEQ     XBX             ;BR IF NO BELL
                                                TSTB    @TPCSR          ;TTY READY.
                                                BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
                                                MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
                                        XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
                                                BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
                                                CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND  AST TIME?
                                                BEQ     1$              ;BR IF YES
                                                MOV     (SP),LSTERR     ;RECORD BEING HERE
                                                CLRB    ERRFLG          ;PREPARE HEADER
                                        1$:     SAVDS                   ;SAVE ALL PROC REGISTERS
                                                MOV     (SP),R5         ;GET THE PC OF ERROR
                                                SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
                                                MOV     (R5),R4         ;GET HLT INSTRUCTION
                                                ASL     R4              ;MULT BY TWO
                                                ADD     (R5),R4         ;DOUBLE IT
                                                ASL     R4              ;MULT AGAIN
                                                BIC     #177001,R4      ;CLEAR JUNK
                                                ADD     #.ERRTAB,R4     ;GET POINTER
                                                MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
                                                MOV     (R4)+,DATAHD    ;GET DATA HEADRER
                                                MOV     (R4),DATABP     ;GET DATA TABLE
                                                TSTB    ERRFLG          ;TYPE HEADREER
                                                BEQ     TYPMSG          ;BR IF YES
                                                TST     DATABP          ;DOES DATA TABLE EXIST?
                                                BNE     TYPDAT          ;BR IF YES.
                                        TYPMSG: TYPE    .MCRLF
                                                TYPE    .MCRLF
                                                TST     LOCK
                                                BEQ     1$
                                                TYPE    .MASTEK
                                        1$:     TYPE    .MTSTN
                                                CNVRT   .XTSTN          ;SHOW IT
```

```
1063  005140  104402  006217                        TYPE    ,MERRPC         ;TYPE PC.
1064  005144  104411  005322                        CNVRT   ,ERTABO         ;SHOW IT
1065  005150  104402  005672                        TYPE    ,MCRLF          ;GIVE A CR/LF
1066  005154  112737  177777  001325                MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1067  005162  005737  005172                        TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1068  005166  001402                                BEQ     WRK0.FM         ;BR IF NO.
1069  005170  104402                                TYPE                    ;TYPE
1070  005172  000000                    ERRMSG:     0                       ;    ERROR MESSAGE
1071  005174                             WRK0.FM:                            ;
1072  005174  005737  005204                        TST     DATAHD          ;DATA HEADER?
1073  005200  001402                                BEQ     TYPDAT          ;BR IF NO
1074  005202  104402                                TYPE                    ;TYPE
1075  005204  000000                    DATAHD:     0                       ;    DATA HEADER
1076  005206  005737  005216          TYPDAT:     TST     DATABP          ;DATA TABLE?
1077  005212  001402                                BEQ     RESREG          ;BR IF NO.
1078  005214  104410                                CONVRT                  ;SHOW
1079  005216  000000                    DATABP:     0                       ;    DATA TABLE
1080  005220  104407                    RESREG:     RESOS                   ;RESTORE PROC REGISTERS
1081  005222  022737  003522  000042    HALTS:      CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, HALT!!
1082  005230  001403                                BEQ     1$
1083  005232  005777  173744                        TST     @SWR            ;HALT ON ERROR?
1084  005236  100005                                BPL     EXITER          ;BR IF NO HALT ON ERROR
1085  005240  010046                    1$:         PUSHRO                  ;SAVE RO
1086  005242  016600  000002                        MOV     2(SP),RO        ;SHOW ERROR PC IN DATA LIGHTS
1087  005246  000000                                HALT                    ;HALT
1088  005250  012600                                POPRO                   ;GET RO
1089  005252  005237  001232          EXITER:     INC     ERRCNT          ;UPDATE ERROR COUNT
1090  005256  032777  000400  173716                BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1091  005264  001007                                BNE     1$              ;BR IF YES
1092  005266  032777  002000  173706                BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1093  005274  001411                                BEQ     2$              ;BR IF NO
1094  005276  013737  001216  001214                MOV     NEXT,RETURN     ;SET FOR NEXT TEST
1095  005304  012706  001200          1$:         MOV     #STACK,SP       ;RESET SP
1096  005310  013701  001404                        MOV     DMCSR,R1        ;SET UP R1
1097  005314  000177  173674                        JMP     @RETURN         ;GOTO SPECIFIED TEST
1098  005320  000002                    2$:         RTI                     ;RETURN
1099  005322  000001                    ERTABO:     1
1100  005324     006     002                        .BYTE   6,2
1101  005326  001276                                SAVPC
1102  005330  000001                    XTSTN:      1
1103  005332     003     002                        .BYTE   3,2
1104  005334  001226                                TSTNO
1105                                               ;ENTER HERE ON POWER FAILURE
1106                                               ;--------------------------
1107
1108
1109  005336                             .PFAIL:
1110  005336  012737  005350  000024                MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
1111  005344  000000                                HALT                            ;HALT ON POWER DOWN NORMAL
1112  005346  000777                                BR      .
1113
1114                                               ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1115
1116  005350                             RESTAR:
1117  005350  012737  005336  000024                MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
1118  005356  012706  001200                        MOV     #STACK,SP       ;RESET THE STACK POINTER
```

```
1119  005362  013701  001404                        MOV     DMCSR,R1        ;RESTORE R1
1120  005366  005037  001416                        CLR     TEMP            ;READY FOR TIMMER
1121  005372  005237  001416                        INC     TEMP            ;PLUS ONE TO THE TIMER!
1122  005376  001375                                BNE     .-4             ;BR IF MORE TO GO
1123  005400  104402  005675                        TYPE    ,MPFAIL         ;TYPE THE MESSAGE
1124  005404  104411  005430                        CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
1125  005410  105037  001325                        CLRB    ERRFLG          ;START CLEAN
1126  005414  005037  001234                        CLR     LSTERR          ;
1127  005420  005011                                CLR     (R1)            ;CLEAR MAINT BITS
1128  005422  104412                                MSTCLR                  ;START CLEAN UP OF DEVICE
1129  005424  000177  173564                        JMP     @RETURN         ;START DOING THAT TEST AGAIN.
1130  005430  000001                        PFTAB:  1
1131  005432     003     002                .BYTE   3,2
1132  005434  001226                                TSTNO
1133
1134  005436                                 .DELAY:
1135  005436  012777  000020  173746                MOV     #20,@DMP04
1136  005444  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1137  005446  121111                                121111                  ;POKE CLOCK DELAY BIT
1138  005450                                 1$:
1139  005450  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1140  005452  121224                                121224                  ;PORT4+IBUS*11
1141  005454  032777  000020  173730                BIT     #BIT4,@DMP04    ;IS CLOCK BIT SET?
1142  005462  001772                                BEQ     1$              ;BR IF NO
1143  005464  000002                                RTI
1144
1145  005466                                 .MSTCLR:
1146  005466  152777  000100  173712                BISB    #BIT6,@DMCSRH   ;SET MASTER CLEAR
1147  005474  142777  000300  173704                BICB    #BIT6!BIT7,@DMCSRH  ;CLEAR MASTER CLEAR AND RUN
1148  005502  000002                                RTI                     ;RETURN
1149
1150  005504                                 .ROMCLK:
1151  005504  152777  000002  173674                BISB    #BIT1,@DMCSRH   ;SET ROMI
1152  005512  013677  173676                        MOV     @(SP)+,@DMP06   ;LOAD INSTRUCTION IN SEL6
1153  005516  062746  000002                        ADD     #2,-(SP)        ;ADJUST STACK
1154  005522  032777  000100  173452                BIT     #SW06,@SWR      ;HALT IF SW06 =1
1155  005530  001401                                BEQ     1$              ;BR IF SW06 =0
1156  005532  000000                                HALT                    ;HALT BEFORE CLOCKING INSTRUCTION
1157  005534  152777  000003  173644        1$:     BISB    #BIT1!BIT0,@DMCSRH ;CLOCK INSTRUCTION
1158  005542  142777  000007  173636                BICB    #BIT2!BIT1!BIT0,@DMCSRH  ;CLEAR ROM0, ROMI, STEP
1159  005550  000002                                RTI
1160
1161  005552                                 .DATACLK:
1162  005552  013637  001416                        MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
1163  005556  062746  000002                        ADD     #2,-(SP)        ;ADJUST STACK
1164  005562  152777  000020  173616        1$:     BISB    #BIT4,@DMCSRH   ;SET STEP LU
1165  005570  027777  173610  173606                CMP     @DMCSR,@DMCSR   ;WASTE TIME
1166  005576  142777  000020  173602                BICB    #BIT4,@DMCSRH   ;CLEAR STEP LU
1167  005604  005337  001416                        DEC     TEMP            ;DEC TICK COUNT
1168  005610  001364                                BNE     1$              ;BR IF NOT DONE
1169  005612  000002                                RTI                     ;RETURN
1170  005614  000001                        3$:     .BLKW 1
1171
1172  005616                                 .TIMER:
1173  005616  013637  001416                        MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
1174  005622  062746  000002                        ADD     #2,-(SP)        ;ADJUST STACK
```

```
1175  005626
1176  005626  104414                          1$:     ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1177  005630  021364                                  021364                  ;PORT4←IBUS* REG11
1178  005632  032777  000002  173552                  BIT     #2,@DMP04       ;IS PGM CLOCK BIT CLEAR?
1179  005640  001772                                  BEQ     1$              ;BR IF YES
1180  005642                                2$:
1181  005642  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1182  005644  021364                                  021364                  ;PORT4←IBUS* REG11
1183  005646  032777  000002  173536                  BIT     #2,@DMP04       ;IS PGM CLOCK BIT SET?
1184  005654  001372                                  BNE     2$              ;BR IF YES
1185  005656  005337  001416                          DEC     TEMP            ;DEC COUNT
1186  005662  001361                                  BNE     1$              ;BR IF NOT DONE
1187  005664  000002                                  RTI                     ;RETURN
1188
1189  005666  020040  000077               MQM:    .ASCIZ  /  ?/
 (2)  005672  005015     000               MCRLF:  .ASCIZ  <15><12>
 (2)  005675     377  053520  020122       MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
 (2)  005733     377  047105  020104       MEPASS: .ASCIZ  <377>/END PASS DZDMC  /
 (2)  005755     377  000122               MR:     .ASCIZ  <377>/R/
 (2)  005760  047377  020117  042504       MERR2:  .ASCIZ  <377>/NO DEVICES PRESENT./
 (2)  006005     377  047111  052523       MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
 (2)  006031     377  042524  052123       MTSTPC: .ASCIZ  <377>/TEST PC-/
 (2)  006043     377  047514  045503       MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
 (2)  006072  051503  035122  000040       MCSRX:  .ASCIZ  /CSR: /
 (2)  006100  042526  035103  000040       MVECX:  .ASCIZ  /VEC: /
 (2)  006106  040520  051523  051505       MPASSX: .ASCIZ  /PASSES: /
 (2)  006117     105  051122  051117       MERRX:  .ASCIZ  /ERRORS: /
 (2)  006130  042524  052123  047040       MTSTN:  .ASCIZ  /TEST NO: /
 (2)  006142  000052                        MASTEK: .ASCIZ  /*/
 (2)  006144  051777  052105  051440       MNEW:   .ASCIZ  <377>/SET SWITCH REG TO DMC11'S DESIRED ACTIVE./
 (2)  006217     120  035103  000040       MERRPC: .ASCIZ  /PC: /
 (2)  006224  020212  020040  020040       XHEAD:  .ASCII  <212>/          MAP OF DMC11 STATUS/
 (2)  006263     377  020040  020040               .ASCII  <377>/          ------------------/
 (2)  006322  020212  050040  020103               .ASCII  <212>/  PC      CSR     STAT1   STAT2   STAT3/
 (2)  006374  026777  026455  026455               .ASCIZ  <377>/------  ------  ------  ------  ------/
 (2)  006450  044377  053517  046440       NUM:    .ASCIZ  <377>/HOW MANY DMC11'S TO BE TESTED?/
 (2)  006510  041777  051123  040440       CSR:    .ASCIZ  <377>/CSR ADDRESS?/
 (2)  006526  053377  041505  047524       VEC:    .ASCIZ  <377>/VECTOR ADDRESS?/
 (2)  006547     377  051102  050040       PRIO:   .ASCIZ  <377>/BR PRIORITY LEVEL? (4,5,6,7)?/
 (2)  006606  044777  020106  046504       CRAM:   .ASCIZ  <377>/IF DMC HAS CRAM (M8204) TYPE "Y", IF CROM (M8200) TYPE "N"
 (2)  006704  053777  044510  044103       MODU:   .ASCIZ  <377>/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M
 (2)  007016  051777  044527  041524       LINE:   .ASCIZ  <377>/SWITCH PAC#1 (DDCMP LINE #)?/
 (2)  007054  051777  044527  041524       BM:     .ASCIZ  <377>/SWITCH PAC#2 (BM873 BOOT ADD)?/
 (2)  007114  044777  020123  044124       CONN:   .ASCIZ  <377>/IS THE LOOP BACK CONNECTOR ON?/
 (2)  007154  047377  020117  042504       NOACT:  .ASCIZ  <377>/NO DEVICES ARE SELECTED/
 (2)  007205     377  051412  051127       SWMES:  .ASCIZ  <377><12>/SWR= /
 (2)  007215     116  053505  020077       SWMES1: .ASCIZ  /NEW? /
 (2)  007223     377  042377  041515       CONERR: .ASCIZ  <377><377>/DMC11 FOUND AT NON-STANDARD ADDRESS  PC: /
 (2)  007277     377  054105  042520       CNERR:  .ASCIZ  <377>/EXPECTED  FOUND/
 (2)  007320  024040  046504  024503       DMCM:   .ASCIZ  / (DMC) /
 (2)  007330  024040  046513  024503       KMCM:   .ASCIZ  / (KMC) /
 (2)  007340  042377  041515  030461       SPEED:  .ASCIZ  <377>/DMC11-AR(REMOTE,LOW SPEED) OR DMC11-AL(LOCAL,HIGH SPEED) T
 (2)                                       .EVEN
 (2)  007454  000005                       XSTATG: 5
1190  007456     006     003                       .BYTE   6,3
1191  007460  001246                               TEMP1
```

```
1192  007462  006  003                          .BYTE   6,3
1193  007464  001250                           TEMP2
1194  007466  006  003                          .BYTE   6,3
1195  007470  001252                           TEMP3
1196  007472  006  003                          .BYTE   6,3
1197  007474  001254                           TEMP4
1198  007476  006  002                          .BYTE   6,2
1199  007500  001256                           TEMP5
1200                                  .EVEN
1201
1202                                          ;BUFFERS FOR INPUT-OUTPUT
1203
1204  007502  000000                  INBUF:  0
1205          007544                  .=.+40
1206  007544  000000                  MDATA:  0
1207          007606                  .=.+40
1208
1209
1210                                          ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1211                                          ;REGISTER USING THE CONSOLE TERMINAL
1212                                          ;-----------------------------------
1213
1214  007606  022737  000176  001202  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
1215  007614  001077                          BNE     CKSWR5          ;BR IF NO
1216  007616  105777  171362                  TSTB    @TKCSR          ;IS DONE SET?
1217  007622  100003                          BPL     2$              ;GO ON IF NOT SET
1218  007624  012737  177777  003734          MOV     #-1,DONE        ;IF DONE SET, SET FLAG
1219  007632  022777  000007  171346  2$:     CMP     #7,@TKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
1220  007640  001404                          BEQ     1$              ;BR IF YES
1221  007642  022777  000207  171336          CMP     #207,@TKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII
1222  007650  001061                          BNE     CKSWR5          ;BR IF NO
1223  007652  010246                  1$:     MOV     R2,-(SP)        ;STORE R2
1224  007654  010346                          MOV     R3,-(SP)        ;STORE R3
1225  007656  010446                          MOV     R4,-(SP)        ;STORE R4
1226  007660  012737  177777  010016          MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
1227  007666  005002                  CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
1228  007670  012704  177777                  MOV     #-1,R4          ;SET FLAG TO ALL ONES
1229  007674  104402  007205                  TYPE    ,SWMES          ;TYPE "SWR= "
1230  007700  104411                  CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
1231  007702  010052                          SOFTSW                  ;OF SOFT SWITCH REGISTER
1232  007704  104402  007215          CKSWR3: TYPE    ,SWMES1         ;TYPE "NEW? "
1233  007710  004737  010020          CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
1234  007714  022703  000015                  CMP     #15,R3          ;WAS IT A CR?
1235  007720  001424                          BEQ     5$              ;BR IF YES
1236  007722  022703  000012                  CMP     #12,R3          ;WAS IT A LF?
1237  007726  001416                          BEQ     4$              ;BR IF YES
1238  007730  022703  000025                  CMP     #25,R3          ;WAS IT CTRL U?
1239  007734  001754                          BEQ     CKSWR1          ;BR IF YES(START OVER)
1240  007736  022703  000007                  CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
1241  007742  001762                          BEQ     CKSWR4
1242  007744  005004                          CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
1243  007746  042703  177770                  BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1244  007752  006302                          ASL     R2              ;SHIFT R2 3 TIMES
1245  007754  006302                          ASL     R2
1246  007756  006302                          ASL     R2
1247  007760  050302                          BIS     R3,R2           ;ADD LAST DIGIT
```

```
1248  007762  000752                          BR      CKSWR4          ;GET NEXT CHARACTER
1249  007764  012766  002002  000006  4$:     MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
1250  007772  005704                  5$:     TST     R4              ;IS FLAG CLEAR?
1251  007774  001002                          BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
1252  007776  010277  171200                  MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
1253  010002  005037  010016          6$:     CLR     SWFLG           ;CLEAR TYPEOUT FLAG
1254  010006  012604                          MOV     (SP)+,R4        ;RESTORE R4
1255  010010  012603                          MOV     (SP)+,R3        ;RESTORE R3
1256  010012  012602                          MOV     (SP)+,R2        ;RESTORE R2
1257  010014  000207                  CKSWR5: RTS     PC              ;RETURN
1258
1259  010016  000000                  SWFLG:  0
1260
1261  010020  105777  171160          INCHAR: TSTB    @TKCSR
1262  010024  100375                          BPL     .-4
1263  010026  017703  171154                  MOV     @TKDBR,R3
1264  010032  105777  171152                  TSTB    @TPCSR
1265  010036  100375                          BPL     .-4
1266  010040  010377  171146                  MOV     R3,@TPDBR
1267  010044  042703  000200                  BIC     #BIT7,R3
1268  010050  000207                          RTS     PC
1269
1270  010052  000001                  SOFTSW: 1
1271  010054     006     002                  .BYTE   6,2
1272  010056  000176                          SWREG
```

```
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282  010060  005737  001306                CYCLE:  TST     DMACTV          ;ARE ANY DMC11'S TO BE TESTED?
1283  010064  001004                                BNE     1$              ;BR IF OK.
1284  010066  104402  007154                        TYPE    ,NOACT          ;NO DMC11'S SELECTED!!
1285  010072  000000                                HALT                    ;STOP THE SHOW.
1286  010074  000776                                BR      .-2             ;DISQUALIFY CONT. SW.
1287  010076  000241                        1$:     CLC                     ;CLEAR PROC. CARRY BIT.
1288  010100  006137  001316                        ROL     RUN             ;UPDATE POINTER
1289  010104  005537  001316                        ADC     RUN             ;CATCH CARRY FROM RUN
1290  010110  062737  000004  001322                ADD     #4,MILK         ;UPDATE POINTER
1291  010116  062737  000010  001320                ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
1292  010124  022737  001700  001320                CMP     #DM.MAP+200,CREAM
1293  010132  001006                                BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
1294  010134  012737  001500  001320                MOV     #DM.MAP,CREAM   ;RESET ADDRESS POINTER.
1295  010142  012737  001702  001322                MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
1296  010150  033737  001316  001306        2$:     BIT     RUN,DMACTV      ;IS THIS ONE ACTIVE?
1297  010156  001747                                BEQ     1$              ;BR IF NO
1298  010160  013700  001320                        MOV     CREAM,R0        ;GET ADDRESS POINTER
1299  010164  013702  001322                        MOV     MILK,R2         ;GET PASS COUNT POINTER
1300  010170  012037  001404                        MOV     (R0)+,DMCSR     ;LOAD SYSTEM CTRL. REG
1301  010174  011037  001374                        MOV     (R0),DMRVEC     ;LOAD VECTOR
1302  010200  042737  177000  001374                BIC     #177000,DMRVEC  ;CLEAR UNWANTED BITS
1303  010206  012037  001366                        MOV     (R0)+,STAT1     ;LOAD STAT1
1304  010212  012037  001370                        MOV     (R0)+,STAT2     ;LOAD STAT2
1305  010216  012037  001372                        MOV     (R0)+,STAT3     ;LOAD STAT3
1306  010222  012237  001230                        MOV     (R2)+,PASCNT    ;LOAD PASS COUNT
1307  010226  012237  001232                        MOV     (R2)+,ERRCNT    ;LOAD ERROR COUNT
1308  010232  012700  000002                        MOV     #2,R0           ;SAVE CORE THIS WAY!
1309  010236  013737  001404  001406                MOV     DMCSR,DMCSRH
1310  010244  005237  001406                        INC     DMCSRH
1311  010250  013737  001406  001410                MOV     DMCSRH,DMCTL
1312  010256  005237  001410                        INC     DMCTL
1313  010262  013737  001410  001412                MOV     DMCTL,DMP04
1314  010270  060037  001412                        ADD     R0,DMP04
1315  010274  013737  001412  001414                MOV     DMP04,DMP06
1316  010302  060037  001414                        ADD     R0,DMP06
1317
1318  010306  013737  001374  001376                MOV     DMRVEC,DMRLVL   ;PTY LVL
1319  010314  060037  001376                        ADD     R0,DMRLVL       ;
1320  010320  013737  001376  001400                MOV     DMRLVL,DMTVEC   ;TX VEC
1321  010326  060037  001400                        ADD     R0,DMTVEC       ;
1322  010332  013737  001400  001402                MOV     DMTVEC,DMTLVL   ;TX LVL
1323  010340  060037  001402                        ADD     R0,DMTLVL
1324
1325  010344  032737  000002  001236                BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
1326  010352  001450                                BEQ     7$              ;BR IF NO
1327  010354                                4$:
1328  010354  005737  000042                        TST     @#42            ;RUNNING IN AUTO MODE?
```

```
1329  010360  001045                          BNE     7$              ;BR IF YFS
1330  010362  104402  005672                  TYPE    ,MCRLF
1331  010366  104403                          INSTR                   ;GET TEST NO.
1332  010370  006130                          MTSTN
1333  010372  104405                          PARAM
1334  010374  0000C1                          1
1335  010376  001000                          1000
1336  010400  001226                          TSTNO
1337  010402     000                  .BYTE   0
1338  010403     001                  .BYTE   1
1339  010404  012700  012320                  MOV     #TST1,R0
1340  010410  022710                  5$:     CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
1341  010412  012737                          MOV     (PC)+,@(PC)+
1342  010414  001020                          BNE     6$              ;BR IF NOT SAME
1343  010416  023760  001226  000002          CMP     TSTNO,2(R0)     ;DOES TSTNO MATCH?
1344  010424  001014                          BNE     6$              ;BR IF NO
1345  010426  022760  001226  000004          CMP     #TSTNO,4(R0)    ;IS LAST WORD OK?
1346  010434  001010                          BNE     6$              ;BR IF NO
1347  010436  010037  001214                  MOV     R0,RETURN       ;IT IS A LEGAL TEST SO DO IT
1348  010442  104402  005755                  TYPE    ,MR
1349  010446  042737  000002  001236          BIC     #SW01,STRTSW
1350  010454  000412                          BR      8$
1351  010456  005720                  6$:     TST     (R0)+           ;POP R0
1352  010460  020027  033734                  CMP     R0,#TLAST+10    ;AT END YET?
1353  010464  001351                          BNE     5$              ;BR IF NO
1354  010466  104402  005666                  TYPE    ,MQM            ;YES ILLEGAL TEST NO.
1355  010472  000730                          BR      4$              ;TRY AGAIN
1356
1357  010474  012737  012320  001214  7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
1358  010502  013701  001404          8$:     MOV     DMCSR,R1        ;R1 = BASE DMC11 ADDRESS
1359  010506  000177  170502                  JMP     @RETURN         ;GO START TESTING.
1360
1361
1362                                          ;ROUTINE USED TO "AUTO SIZE" THE DMC11
1363                                          ;CSR AND VECTOR.
1364                                          ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
1365                                          ;       ADDRESS RANGE (160000:164000)
1366                                          ;       AND THE VECTOR MAY BE ANY WHERE IN THE
1367                                          ;       FLOATING VECTOR RANGE (300:770)
1368                                          ;
1369
1370  010512                          AUTO.SIZE:
1371  010512  000005                          RESET                   ;INSURE A BUS INIT.
1372  010514  012702  001500          CSRMAP: MOV     #DM.MAP,R2      ;LOAD MAP POINTER.
1373  010520  005022                  1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1374  010522  022702  001700                  CMP     #DM.END,R2      ;ALL DONE?
1375  010526  001374                          BNE     1$              ;BR IF NO
1376  010530  005037  001310                  CLR     DMNUM           ;SET OCTAL NUMBER OF DMC11'S TO 0
1377  010534  012702  001500                  MOV     #DM.MAP,R2      ;R2 POINTS TO DMC MAP
1378  010540  005037  001306                  CLR     DMACTV          ;CLEAR ACTIVE
1379  010544  032737  000001  001236          BIT     #SW00,STRTSW    ;QUESTIONS?
1380  010552  001002                          BNE     .+6             ;BR IF YES
1381  010554  000137  011252                  JMP     7$              ;IF NO SKIP QUESTIONS
1382  010560  012737  000001  001256          MOV     #1,TEMP5        ;START WITH 1
1383  010566  104403                          INSTR
1384  010570  006450                          NUM
```

```
1385  010572  104405                          PARAM
1386  010574  000001                          1
1387  010576  000C30                          16.
1388  010600  001252                          TEMP3
1389  010602     000                          .BYTE   0
1390  010603     001                          .BYTE   1
1391  010604  013737  001252  001310          MOV     TEMP3,DMNUM     ;DMNUM = HOW MANY
1392  010612  104402  005672            12$:  TYPE    ,MCRLF
1393  010616  104410                          CONVRT                  ;TYPE WHICH DMC IS BEING DONE
1394  010620  012002                          WHICH                   ;TEMP5 IS WHICH DMC
1395  010622  005237  001256                  INC     TEMP5
1396  010626  104403                          INSTR
1397  010630  006510                          CSR
1398  010632  104405                          PARAM
1399  010634  160000                          160000
1400  010636  164000                          164000
1401  010640  001254                          TEMP4
1402  010642     000                          .BYTE   0
1403  010643     001                          .BYTE   1
1404  010644  013722  001254                  MOV     TEMP4,(R2)+     ;STORE CSR IN MAP
1405  010650  104403                          INSTR
1406  010652  006526                          VEC
1407  010654  104405                          PARAM
1408  010656  000000                          0
1409  010660  000776                          776
1410  010662  001254                          TEMP4
1411  010664     000                          .BYTE   0
1412  010665     001                          .BYTE   1
1413  010666  013712  001254                  MOV     TEMP4,(R2)      ;STORE VECTOR IN MAP
1414  010672  104402                  10$:    TYPE
1415  010674  006547                          PRIO                    ;ASK WHAT BR LEVEL
1416  010676  004737  012266                  JSR     PC,INTTY        ;GET RESPONSE
1417  010702  022703  000024                  CMP     #24,R3          ;
1418  010706  101014                          BHI     50$             ;BR IF LESS THAN 4
1419  010710  022703  000027                  CMP     #27,R3          ;
1420  010714  103411                          BLO     50$             ;BR IF GREATER THAN 7
1421  010716  012704  000011                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
1422  010722  006303                          ASL     R3              ;SHIFT R3 LEFT
1423  010724  005304                          DEC     R4              ;DEC SHIFT COUNT
1424  010726  001375                          BNE     .-4             ;BR IF NOT DONE
1425  010730  042703  170777                  BIC     #170777,R3      ;BIC UNWANTED BITS
1426  010734  050312                          BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
1427  010736  000403                          BR      8$              ;CONTINUE
1428  010740  104402                  50$:    TYPE
1429  010742  005666                          MQM                     ;RESPONSE IS OUT OF LIMITS
1430  010744  000752                          BR      10$             ;TRY AGAIN
1431  010746  104402                  8$:     TYPE
1432  010750  006606                          CRAM                    ;DOES DMC HAVE CRAM?
1433  010752  004737  012266                  JSR     PC,INTTY        ;GET REPLY
1434  010756  022703  000131                  CMP     #131,R3
1435  010762  001427                          BEQ     9$              ;YES
1436  010764  022703  000116                  CMP     #116,R3         ;NO
1437  010770  001403                          BEQ     40$             ;NOT A Y OR N
1438  010772  104402                          TYPE
1439  010774  005666                          MQM                     ;TYPE "?"
1440  010776  000763                          BR      8$              ;ASK AGAIN
```

```
1441  011000  104402                    40$:  TYPE
1442  011002  007340                          SPEED                   ;DMC11-AR OR DMC11-AL?
1443  011004  004737  012266                 JSR     PC,INTTY        ;GET RESPONSE
1444  011010  022703  000122                 CMP     #122,R3         ;IS IT R
1445  011014  001414                          BEQ     16$             ;BR IF REMOTE
1446  011016  022703  000114                 CMP     #114,R3         ;IS IT L
1447  011022  001403                          BEQ     41$             ;BR IF LOCAL
1448  011024  104402                          TYPE
1449  011026  005666                          MQM
1450  011030  000763                          BR      40$             ;TRY AGAIN
1451  011032  052762  000002  000004    41$:  BIS     #BIT1,4(R2)     ;SET BIT1 IN STAT3
1452  011040  000402                          BR      16$             ;CONTINUE
1453  011042  052712  100000             9$:  BIS     #BIT15,(R2)     ;SET BIT 15 IF CRAM
1454  011046  104402                    16$:  TYPE
1455  011050  006704                          MODU                    ;ASK WHICH LINE UNIT
1456  011052  004737  012266                 JSR     PC,INTTY        ;GET REPLY
1457  011056  022703  000021                 CMP     #21,R3          ;"1"
1458  011062  001417                          BEQ     30$
1459  011064  022703  000022                 CMP     #22,R3          ;"2"
1460  011070  001412                          BEQ     31$
1461  011072  022703  000116                 CMP     #116,R3         ;"N"
1462  011076  001403                          BEQ     32$
1463  011100  104402                          TYPE
1464  011102  005666                          MQM                     ;IF NOT A 1,2 OR N TYPE "?"
1465  011104  000760                          BR      16$             ;TRY AGIAN
1466  011106  052722  010000             32$:  BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
1467  011112  022222                          CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
1468  011114  000447                          BR      33$
1469  011116  052712  020000             31$:  BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF M8202
1470  011122  104402                    30$:  TYPE
1471  011124  007114                          CONN                    ;ASK IF LOOP-BACK IS ON
1472  011126  004737  012266                 JSR     PC,INTTY        ;GET REPLY
1473  011132  022703  000131                 CMP     #131,R3         ;Y
1474  011136  001406                          BEQ     17$
1475  011140  022703  000116                 CMP     #116,R3         ;N
1476  011144  001406                          BEQ     18$
1477  011146  104402                          TYPE
1478  011150  005666                          MQM                     ;IF NOT Y OR N TYPE "?"
1479  011152  000763                          BR      30$             ;TRY AGAIN
1480  011154  052722  040000             17$:  BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
1481  011160  000402                          BR      19$
1482  011162  042722  040000             18$:  BIC     #BIT14,(R2)+    ;NO TURNAROUND
1483  011166                             19$:
1484  011166  104403                          INSTR
1485  011170  007016                          LINE
1486  011172  104405                          PARAM
1487  011174  000000                          0
1488  011176  000377                          377
1489  011200  001254                          TEMP4
1490  011202     000                          .BYTE   0
1491  011203     001                          .BYTE   1
1492  011204  113722  001254                 MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
1493  011210  104403                          INSTR
1494  011212  007054                          BM
1495  011214  104405                          PARAM
1496  011216  000000                          0
```

```
1497  011220  000377                                    377
1498  011222  001254                                    TEMP4
1499  011224     000                                    .BYTE   0
1500  011225     001                                    .BYTE   1
1501  011226  113722  001254                           MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
1502  011232  005722                                    TST     (R2)+           ;POP OVER STAT3
1503  011234  005337  001252                 33$:      DEC     TEMP3           ;DEC DMC COUNT
1504  011240  001402                                    BEQ     34$             ;BR IF DONE
1505  011242  000137  010612                           JMP     12$             ;JUMP IF NOT
1506  011246  000137  011702                 34$:      JMP     13$             ;CONTINUE
1507  011252  012701  160000                 7$:       MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
1508  011256  012737  011774  000004                    MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
1509  011264  005011                          2$:       CLR     (R1)            ;CLEAR SEL0
1510  011266  005711                                    TST     (R1)            ;IF DMC11 DMCSR S/B 0
1511  011270  001172                                    BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DMC1
1512  011272  005061  000006                           CLR     6(R1)           ;CLEAR SEL6
1513  011276  005761  000006                           TST     6(R1)           ;IF DMC11 THEN DMRIC S/B =0!
1514  011302  001165                                    BNE     3$              ;BR IF NOT DMC11
1515  011304  012711  002000                           MOV     #BIT10,(R1)     ;SET ROM0
1516  011310  005061  000004                           CLR     4(R1)           ;CLEAR SEL4
1517  011314  012761  125252  000006                   MOV     #125252,6(R1)   ;WRITE THIS TO SEL6
1518  011322  052711  020000                           BIS     #BIT13,(R1)     ;WRITE IT!
1519  011326  022761  125252  000004                   CMP     #125252,4(R1)   ;WAS IT WRITTEN?
1520  011334  001004                                    BNE     21$             ;IF NO IT IS NOT CRAM
1521  011336  052762  100000  000002                   BIS     #BIT15,2(R2)    ;SET BIT15 IF CRAM
1522  011344  000431                                    BR      22$
1523  011346  012711  001000                 21$:      MOV     #BIT9,(R1)      ;SET ROM1
1524  011352  012761  100417  000006                   MOV     #100417,6(R1)   ;PUT INSTRUCTION IN SEL6
1525  011360  012711  001400                           MOV     #BIT9!BIT8,(R1) ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
1526  011364  012711  002000                           MOV     #BIT10,(R1)     ;SET ROM0
1527  011370  022761  000626  000006                   CMP     #626,6(R1)      ;IS IT LOCAL CROM
1528  011376  001411                                    BEQ     23$             ;BR IF YES
1529  011400  022761  016520  000006                   CMP     #16520,6(R1)    ;IS IT REMOTE CROM?
1530  011406  001410                                    BEQ     22$             ;BR IF YES
1531  011410  022761  177777  000006                   CMP     #-1,6(R1)       ;NO CROM?
1532  011416  001404                                    BEQ     22$             ;BR IF YES
1533  011420  000516                                    BR      3$              ;NOT A DMC
1534  011422  052762  000002  000006         23$:      BIS     #BIT1,6(R2)     ;SET BIT 1 IN STAT3
1535                                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DMC11 CSR ADDRESS.
1536  011430  010122                          22$:      MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
1537  011432  012711  001000                 15$:      MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
1538  011436  005061  000004                           CLR     4(R1)           ;CLEAR PORT4
1539  011442  012761  122113  000006                   MOV     #122113,6(R1)   ;LOAD INSTRUCTION (CLR DTR)
1540  011450  052711  000400                           BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
1541  011454  012761  021264  000006                   MOV     #021264,6(R1)   ;LOAD INSTRUCTION
1542  011462  052711  000400                           BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
1543  011466  122761  000377  000004                   CMPB    #377,4(R1)      ;IS IT ALL ONES?
1544  011474  001003                                    BNE     .+10            ;BR IF NO
1545  011476  052712  010000                           BIS     #BIT12,(R2)     ;IF YES, NO LINE UNIT, SET STATUS BIT
1546  011502  000436                                    BR      20$
1547  011504  032761  000002  000004                   BIT     #BIT1,4(R1)     ;IS SWITCH A ONE?
1548  011512  001403                                    BEQ     .+10            ;BR IF M8201
1549  011514  052712  060000                           BIS     #BIT13!BIT14,(R2) ;M8202 ASSUME CONNECTOR
1550  011520  000427                                    BR      20$             ;CONNECTOR ON)
1551  011522  032761  000010  000004                   BIT     #BIT3,4(R1)     ;IS MRDY SET
1552  011530  001023                                    BNE     20$             ;BR IF M8201 NO CONNECTOR (ON LINE)
```

```
1553  011532  012761  000100  000004          MOV     #BIT6,4(R1)     ;LOAD PORT4
1554  011540  012761  122113  000006          MOV     #122113,6(R1)   ;LOAD INSTRUCTION
1555  011546  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(SET DTR)
1556  011552  012761  021264  000006          MOV     #021264,6(R1)   ;LOAD INSTRUCTION
1557  011560  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(READ MODEM REG)
1558  011564  032761  000010  000004          BIT     #BIT3,4(R1)     ;IS MRDY SET NOW?
1559  011572  001402                          BEQ     20$             ;BR IF NO CONNECTOR
1560  011574  052712  040000                  BIS     #BIT14,(R2)     ;SET STATUS BIT FOR CONNECTOR
1561  011600  005722                  20$:    TST     (R2)+           ;POP POINTER
1562  011602  012761  021324  000006          MOV     #021324,6(R1)   ;PUT INSTRUCTION IN PORT6
1563  011610  012711  001400                  MOV     #BIT9!BIT8,(R1) ;PORT4=LU 15
1564  011614  156122  000004                  BISB    4(R1),(R2)+     ;STORE DDCMP LINE # IN TABLE
1565  011620  012761  021344  000006          MOV     #021344,6(R1)   ;PORT6=INSTRUCTION
1566  011626  012711  001400                  MOV     #BIT8!BIT9,(R1) ;CLOCK INSTR.
1567  011632  156122  000004                  BISB    4(R1),(R2)+     ;STORE BM873 ADD IN TABLE
1568  011636  005722                          TST     (R2)+           ;POP OVER STAT3
1569  011640  005011                          CLR     (R1)            ;CLEAR ROMI
1570  011642  005237  001310                  INC     DMNUM           ;UPDATE DEVICE COUNTER
1571  011646  022737  000020  001310          CMP     #20,DMNUM       ;ARE MAX. NO. OF DEV FOUND?
1572  011654  001412                          BEQ     13$             ;YES DON'T LOOK FOR ANY MORE.
1573  011656  005011                  3$:     CLR     (R1)            ;CLEAR BIT 10
1574  011660  005061  000006                  CLR     6(R1)           ;CLEAR SEL 6
1575  011664  062701  000010          14$:    ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
1576  011670  022701  164000                  CMP     #164000,R1
1577  011674  001402                          BEQ     13$             ;BR IF DONE
1578  011676  000137  011264                  JMP     2$              ;JUMP IF NOT
15    011702  005037  001306          13$:    CLR     DMACTV
1580  011706  005737  001310                  TST     DMNUM           ;WERE ANY DMC11'S FOUND AT ALL?
1581  011712  001423                          BEQ     5$              ;ERROR AUTO SIZER FOUND NO DMC11'S IN THIS SYS.
1582  011714  013701  001310                  MOV     DMNUM,R1        ;
1583  011720  010137  001314                  MOV     R1,SAVNUM       ;SAVE NUMBER OF DEVICES
1584  011724  000241                  4$:     CLC
1585  011726  006137  001306                  ROL     DMACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
1586  011732  005237  001306                  INC     DMACTV          ;SET THE BIT
1587  011736  005301                          DEC     R1
1588  011740  001371                          BNE     4$              ;BR IF MORE TO GENERATE
1589  011742  012737  000006  000004          MOV     #6,@#4          ;RESTORE TRAP VECTOR
1590  011750  013737  001306  001312          MOV     DMACTV,SAVACT   ;SAVE ACTIVE REGISTER
1591  011756  000137  012010                  JMP     VECMAP          ;GO FIND THE VECTOR NOW.
1592  011762  104402  005760          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO DMC11'S FOUND.
1593  011766  005000                          CLR     R0              ;MAKE DATA LIGHTS ZERO
1594  011770  000000                          HALT                    ;STOP THE SHOW
1595  011772  000776                          BR      .-2             ;DISABLE CONT. SW.
1596  011774  012716  011664          6$:     MOV     #14$,(SP)       ;ENTERED BY NON-EXISTANT TIME-OUT.
1597  012000  000002                          RTI                     ;RETURN TO MAINSTREAM
1598
1599  012002  000001          WHICH:  1
1600  012004     002     002                  .BYTE   2,2
1601  012006  001256                          TEMP5
1602
1603  012010  032737  000001  001236  VECMAP: BIT     #SW00,STRTSW
1604  012016  001114                          BNE     5$
1605  012020  012737  000340  000022          MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
1606  012026  012737  012202  000020          MOV     #4$,@#20        ;SET IOT TRAP VECTOR
1607  012034  012702  001500                  MOV     #DM.MAP,R2      ;SET SOFTWARE POINTER
1608  012040  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
```

```
1609  012044  012701  000302                    MOV     #302,R1         ;PC OF IOT INSTR.
1610  012050  010120                    1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
1611  012052  012721  000004                    MOV     #4,(R1)+        ;WITH .+2; IOT
1612  012056  022021                            CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
1613  012060  020127  001000                    CMP     R1,#1000
1614  012064  101771                            BLOS    1$              ;BR IF MORE TO FILL
1615  012066  013737  001306  001246            MOV     DMACTV,TEMP1    ;STORE TEMPORALLY
1616  012074  006037  001246            2$:     ROR     TEMP1           ;BRING OUT A BIT
1617  012100  103063                            BCC     5$              ;BR IF ALL DONE
1618  012102  012704  000012                    MOV     #12,R4          ;R4 IS INDEX REGISTER
1619  012106  016437  012252  177776            MOV     BRLVL(R4),PS    ;SET PS TO 7
1620  012114  011201                            MOV     (R2),R1
1621  012116  012761  000200  000004            MOV     #200,4(R1)
1622  012124  012711  001000                    MOV     #BIT9,(R1)      ;SET ROMI
1623  012130  012761  121111  000006            MOV     #121111,6(R1)   ;PUT INSTRUCTION IN PORT6
1624  012136  012711  001400                    MOV     #BIT9!BIT8,(R1) ;FORCE AN INTERRUPT
1625  012142  105200                    7$:     INCB    R0              ;STALL
1626  012144  001376                            BNE     .-2             ;FOR TIME TO INTERUPT
1627  012146  162704  000002                    SUB     #2,R4           ;GET NEXT LOWEST PS LEVEL
1628  012152  001404                            BEQ     6$              ;BR IF R4 = 0
1629  012154  016437  012252  177776            MOV     BRLVL(R4),PS    ;MOVE NEXT LOWER LEVEL IN PS
1630  012162  000767                            BR      7$              ;BR TO DELAY
1631  012164  052762  005300  000002    6$:     BIS     #5300,2(R2)     ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX DMC11
1632  012172  005011            3$:             CLR     (R1)            ;CLEAR ROMI
1633  012174  062702  000010                    ADD     #10,R2          ;POP SOFTWARE POINTER
1634  012200  000735                            BR      2$              ;KEEP GOING
1635  012202  051662  000002            4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
1636  012206  042762  000007  000002            BIC     #7,2(R2)        ;CLEAR JUNK
1637  012214  016405  012254                    MOV     BRLVL+2(R4),R5  ;GET BR LEVEL OF DMC11
1638  012220  006305                            ASL     R5              ;SHIFT LEVEL 4 PLACES
1639  012222  006305                            ASL     R5              ;TO THE LEFT FOR THE
1640  012224  006305                            ASL     R5              ;STATUS TABLE
1641  012226  006305                            ASL     R5
1642  012230  042705  170777                    BIC     #170777,R5      ;CLEAR UNWANTED BITS
1643  012234  050562  000002                    BIS     R5,2(R2)        ;PUT BR LEVEL IN STATUS TABLE
1644  012240  022626                            CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
1645  012242  012716  012172                    MOV     #3$,(SP)        ;SET FOR RETURN
1646  012246  000002                            RTI
1647  012250  000207            5$:             RTS     PC              ;ALL DONE WITH "AUTO SIZING"
1648
1649  012252  000000            BRLVL:  0       ;LEVEL 0
1650  012254  000000                    0       ;LEVEL 0
1651  012256  000200                    200     ;LEVEL 4
1652  012260  000240                    240     ;LEVEL 5
1653  012262  000300                    300     ;LEVEL 6
1654  012264  000340                    340     ;LEVEL 7
1655
1656
1657  012266  105777  166712            INTTY:  TSTB    @TKCSR          ;WAIT FOR DONE
1658  012272  100375                            BPL     .-4
1659  012274  017703  166706                    MOV     @TKDBR,R3       ;PUT CHAR IN R3
1660  012300  105777  166704                    TSTB    @TPCSR          ;WAIT UNTIL PRINTER IS READY
1661  012304  100375                            BPL     .-4
1662  012306  010377  166700                    MOV     R3,@TPDBR       ;ECHO CHAR
1663  012312  042703  000240                    BIC     #BIT7!BIT5,R3   ;MASK OFF LOWER CASE
1664  012316  000207                            RTS     PC              ;RETURN
```

```
1665
1666
1667
1668                                            ;*************************** TEST 1 ****************************
1669                                            ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
1670                                            ;*DOES NOT CAUSE A TIME OUT TRAP
1671                                            ;:***********************************************************
1672
1673                                            ;  TEST 1
1674                                            ;--------------
1675  012320  012737  000001  001226    TST1:   MOV     #1,TSTNO
1676  012326  012737  012426  001216            MOV     #TST2,NEXT
1677  012334  012737  012366  001220            MOV     #1$,LOCK
1678                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
1679  012342  013701  001404                    MOV     DMCSR,R1        ;R1 CONTAINS BASE DMC11 ADDRESS
1680  012346  012700  000004                    MOV     #4,R0           ;4 REGISTERS TO BE TESTED
1681  012352  012737  012420  000004            MOV     #2$,4           ;SET UP TIMEOUT TRAP
1682  012360  012737  000340  000006            MOV     #340,6          ;LEVEL 7
1683  012366  005711                    1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
1684  012370  000240                            NOP
1685  012372  104401                            SCOP1                   ;SW09=1?
1686  012374  062701  000002                    ADD     #2,R1           ;NEXT REGISTER
1687  012400  005300                            DEC     R0              ;DEC REGISTER COUNT
1688  012402  001371                            BNE     1$              ;BR IF NOT LAST REGISTER
1689  012404  012737  000006  000004            MOV     #6,4            ;RESTORE LOC 4
1690  012412  005037  000006                    CLR     6               ;RESTORE LOC 6
1691  012416  104400                            SCOPE                   ;SCOPE THIS TEST
1692  012420  011602                    2$:     MOV     (SP),R2         ;GET PC OF TRAP
1693  012422  104001                            HLT     1               ;TIME-OUT ERROR
1694  012424  000002                            RTI
1695
1696
1697                                            ;*************************** TEST 2 ****************************
1698                                            ;*VERIFY THAT RUN CAN BE CLEARED
1699                                            ;:***********************************************************
1700
1701                                            ;  TEST 2
1702                                            ;--------------
1703  012426  012737  000002  001226    TST2:   MOV     #2,TSTNO
1704  012434  012737  012456  001216            MOV     #TST3,NEXT
1705                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
1706  012442  005011                            CLR     (R1)            ;CLEAR DMCSR
1707  012444  005005                            CLR     R5              ;CLEAR "EXPECTED"
1708  012446  011104                            MOV     (R1),R4         ;PUT DMCSR IN "FOUND"
1709  012450  001401                            BEQ     1$              ;BR IF CLEARED
1710  012452  104002                            HLT     2               ;ERROR DMCSR NOT CLEARED
1711  012454  104400                    1$:     SCOPE                   ;SCOPE THIS TEST
1712
1713
1714                                            ;*************************** TEST 3 ****************************
1715                                            ;*UNIBUS REGISTER WORD DUAL ADDRESSING TEST
1716                                            ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
1717                                            ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
1718                                            ;:***********************************************************
1719
1720                                            ;  TEST 3
```

```
1721                                                    ;--------------
1722  012456  012737  000003  001226          TST3:     MOV     #3,TSTNO
1723  012464  012737  012606  001216                    MOV     #TST4,NEXT
1724  012472  012737  012506  001220                    MOV     #1$,LOCK
1725                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
1726  012500  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
1727  012502  012700  000001                           MOV     #1,R0           ;START PATTERN AT 1
1728  012506  005011                           1$:     CLR     (R1)            ;CLEAR REGISTER
1729  012510  010005                                   MOV     R0,R5           ;PUT DATA IN "EXPECTED"
1730  012512  010011                                   MOV     R0,(R1)         ;WRITE DMC REGISTER WITH PATTERN
1731  012514  011104                                   MOV     (R1),R4         ;READ DMC REGISTER INTO "FOUND"
1732  012516  020504                                   CMP     R5,R4           ;IS DATA CORRECT
1733  012520  001401                                   BEQ     2$              ;BR IF YES
1734  012522  104002                                   HLT     2               ;DATA ERROR
1735  012524  104401                           2$:     SCOP1                   ;SW09=1?
1736  012526  005721                                   TST     (R1)+           ;NEXT REGISTER
1737  012530  005200                                   INC     R0              ;INCREMENT DATA PATTERN
1738  012532  022700  000005                           CMP     #5,R0           ;LAST REGISTER?
1739  012536  001363                                   BNE     1$              ;BR IF NO
1740  012540  013701  001404                           MOV     DMCSR,R1        ;BASE DMC11  ADDRESS TO R1
1741  012544  012700  000001                           MOV     #1,R0           ;RESTART PATTERN AT 1
1742  012550  012737  012556  001220                    MOV     #3$,LOCK        ;NEW SCOP1
1743  012556  010005                           3$:     MOV     R0,R5           ;PUT DATA IN "EXPECTED"
1744  012560  011104                                   MOV     (R1),R4         ;READ DMC REGISTER INTO "FOUND"
1745  012562  020504                                   CMP     R5,R4           ;IS DATA CORRECT
1746  012564  001401                                   BEQ     4$              ;BR IF YES
1747  012566  104002                                   HLT     2               ;DUAL ADDRESSING ERROR
1748  012570  104401                           4$:     SCOP1                   ;SW09=1?
1749  012572  005721                                   TST     (R1)+           ;NEXT REGISTER
1750  012574  005200                                   INC     R0              ;INCREMENT PATTERN
1751  012576  022700  000005                           CMP     #5,R0           ;LAST REGISTER?
1752  012602  001365                                   BNE     3$              ;BR IF NO
1753  012604  104400                                   SCOPE                   ;SCOPE THIS TEST
1754
1755
1756                                                    ;************************** TEST 4 **************************
1757                                                    ;*CONTROL STATUS REGISTER WRITE/READ TEST
1758                                                    ;*SET BIT0, VERIFY BIT0 WAS SET
1759                                                    ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
1760                                                    ;;************************************************************
1761
1762                                                    ;  TEST 4
1763                                                    ;--------------
1764  012606  012737  000004  001226          TST4:     MOV     #4,TSTNO
1765  012614  012737  012704  001216                    MOV     #TST5,NEXT
1766  012622  012737  012632  001220                    MOV     #1$,LOCK
1767  012630  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
1768  012632  013701  001404                   1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1769  012636  012705  000001                           MOV     #BIT0,R5        ;PUT DATA IN "EXPECTED"
1770  012642  010511                                   MOV     R5,(R1)         ;WRITE BIT 0
1771  012644  011104                                   MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1772  012646  020504                                   CMP     R5,R4           ;IS DATA CORRECT
1773  012650  001401                                   BEQ     2$              ;BR IF YES
1774  012652  104002                                   HLT     2               ;DATA ERROR
1775  012654  104401                           2$:     SCOP1                   ;SW09 UP?
1776  012656  012737  012664  001220                    MOV     #3$,LOCK        ;NEW SCOP1
```

```
1777  012664  042711  000001          3$:     BIC     #BIT0,(R1)      ;CLEAR BIT 0
1778  012670  005005                          CLR     R5              ;CLEAR "EXPECTED"
1779  012672  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1780  012674  001402                          BEQ     4$              ;BR IF ZERO
1781  012676  104002                          HLT     2               ;DATA ERROR BIT0 NOT CLEARED
1782  012700  104401                          SCOP1                   ;SW09 UP?
1783  012702  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
1784
1785
1786                                          ;*************************** TEST 5 ****************************
1787                                          ;*CONTROL STATUS REGISTER WRITE/READ TEST
1788                                          ;*SET BIT1, VERIFY BIT1 WAS SET
1789                                          ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED
1790                                          ;:***********************************************************
1791
1792                                          ;  TEST 5
1793                                          ;--------------
1794  012704  012737  000005  001226  TST5:   MOV     #5,TSTNO
1795  012712  012737  013002  001216          MOV     #TST6,NEXT
1796  012720  012737  012730  001220          MOV     #1$,LOCK
1797  012726  104412                          MSTCLR                  ;MASTER CLEAR DMC11
1798  012730  013701  001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1799  012734  012705  000002                  MOV     #BIT1,R5        ;PUT DATA IN "EXPECTED"
1800  012740  010511                          MOV     R5,(R1)         ;WRITE BIT 1
1801  012742  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1802  012744  020504                          CMP     R5,R4           ;IS DATA CORRECT
1803  012746  001401                          BEQ     2$              ;BR IF YES
1804  012750  104002                          HLT     2               ;DATA ERROR
1805  012752  104401                  2$:     SCOP1                   ;SW09 UP?
1806  012754  012737  012762  001220          MOV     #3$,LOCK        ;NEW SCOP1
1807  012762  042711  000002          3$:     BIC     #BIT1,(R1)      ;CLEAR BIT 1
1808  012766  005005                          CLR     R5              ;CLEAR "EXPECTED"
1809  012770  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1810  012772  001402                          BEQ     4$              ;BR IF ZERO
1811  012774  104002                          HLT     2               ;DATA ERROR BIT1 NOT CLEARED
1812  012776  104401                          SCOP1                   ;SW09 UP?
1813  013000  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
1814
1815
1816                                          ;*************************** TEST 6 ****************************
1817                                          ;*CONTROL STATUS REGISTER WRITE/READ TEST
1818                                          ;*SET BIT2, VERIFY BIT2 WAS SET
1819                                          ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED
1820                                          ;:***********************************************************
1821
1822                                          ;  TEST 6
1823                                          ;--------------
1824  013002  012737  000006  001226  TST6:   MOV     #6,TSTNO
1825  013010  012737  013100  001216          MOV     #TST7,NEXT
1826  013016  012737  013026  001220          MOV     #1$,LOCK
1827  013024  104412                          MSTCLR                  ;MASTER CLEAR DMC11
1828  013026  013701  001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1829  013032  012705  000004                  MOV     #BIT2,R5        ;PUT DATA IN "EXPECTED"
1830  013036  010511                          MOV     R5,(R1)         ;WRITE BIT 2
1831  013040  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1832  013042  020504                          CMP     R5,R4           ;IS DATA CORRECT
```

```
1833  013044  001401                              BEQ     2$              ;BR IF YES
1834  013046  104002                              HLT     2               ;DATA ERROR
1835  013050  104401                      2$:     SCOP1                   ;SW09 UP?
1836  013052  012737  013060  001220              MOV     #3$,LOCK        ;NEW SCOP1
1837  013060  042711  000004              3$:     BIC     #BIT2,(R1)      ;CLEAR BIT 2
1838  013064  005005                              CLR     R5              ;CLEAR "EXPECTED"
1839  013066  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1840  013070  001402                              BEQ     4$              ;BR IF ZERO
1841  013072  104002                              HLT     2               ;DATA ERROR BIT2 NOT CLEARED
1842  013074  104401                              SCOP1                   ;SW09 UP?
1843  013076  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1844
1845
1846                                              ;*************************** TEST     ************************
1847                                              ;*CONTROL STATUS REGISTER WRITE/REA  EST
1848                                              ;*SET BITS, VERIFY BITS WAS SET
1849                                              ;*CLEAR BITS, VERIFY BITS WAS CLEARED
1850                                              ;;*************************************************************
1851
1852                                              ;  TEST 7
1853                                              ;-------------
1854  013100  012737  000007  001226      TST7:   MOV     #7,TSTNO
1855  013106  012737  013176  001216              MOV     #TST10,NEXT
1856  013114  012737  013124  001220              MOV     #1$,LOCK
1857  013122  104412                              MSTCLR                  ;MASTER CLEAR DMC11
1858  013124  013701  001404              1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1859  013130  012705  000040                      MOV     #BIT5,R5        ;PUT DATA IN "EXPECTED"
1860  013134  010511                              MOV     R5,(R1)         ;WRITE BIT 5
1861  013136  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1862  013140  020504                              CMP     R5,R4           ;IS DATA CORRECT
1863  013142  001401                              BEQ     2$              ;BR IF YES
1864  013144  104002                              HLT     2               ;DATA ERROR
1865  013146  104401                      2$:     SCOP1                   ;SW09 UP?
1866  013150  012737  013156  001220              MOV     #3$,LOCK        ;NEW SCOP1
1867  013156  042711  000040              3$:     BIC     #BIT5,(R1)      ;CLEAR BIT 5
1868  013162  005005                              CLR     R5              ;CLEAR "EXPECTED"
1869  013164  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1870  013166  001402                              BEQ     4$              ;BR IF ZERO
1871  013170  104002                              HLT     2               ;DATA ERROR BITS NOT CLEARED
1872  013172  104401                              SCOP1                   ;SW09 UP?
1873  013174  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1874
1875
1876                                              ;*************************** TEST 10 ***************************
1877                                              ;*CONTROL STATUS REGISTER WRITE/READ TEST
1878                                              ;*SET BIT6, VERIFY BIT6 WAS SET
1879                                              ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED
1880                                              ;;*************************************************************
1881
1882                                              ;  TEST 10
1883                                              ;-------------
1884  013176  012737  000010  001226      TST10:  MOV     #10,TSTNO
1885  013204  012737  013274  001216              MOV     #TST11,NEXT
1886  013212  012737  013222  001220              MOV     #1$,LOCK
1887  013220  104412                              MSTCLR                  ;MASTER CLEAR DMC11
1888  013222  013701  001404              1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
```

```
1889  013226  012705  000100                      MOV     #BIT6,R5        ;PUT DATA IN "EXPECTED"
1890  013232  010511                              MOV     R5,(R1)         ;WRITE BIT 6
1891  013234  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1892  013236  020504                              CMP     R5,R4           ;IS DATA CORPECT
1893  013240  001401                              BEQ     2$              ;BR IF YES
1894  013242  104002                              HLT     2               ;DATA ERROR
1895  013244  104401                      2$:     SCOP1                   ;SW09 UP?
1896  013246  012737  013254  001220              MOV     #3$,LOCK        ;NEW SCOP1
1897  013254  042711  000100              3$:     BIC     #BIT6,(R1)      ;CLEAR BIT 6
1898  013260  005005                              CLR     R5              ;CLEAR "EXPECTED"
1899  013262  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1900  013264  001402                              BEQ     4$              ;BR IF ZERO
1901  013266  104002                              HLT     2               ;DATA ERROR BIT6 NOT CLEARED
1902  013270  104401                              SCOP1                   ;SW09 UP?
1903  013272  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1904
1905
1906                                              ;*************************** TEST 11 ****************************
1907                                              ;*CONTROL STATUS REGISTER WRITE/READ TEST
1908                                              ;*SET BIT7, VERIFY BIT7 WAS SET
1909                                              ;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED
1910                                              ;:************************************************************
1911
1912                                              ;  TEST 11
1913                                              ;--------------
1914  013274  012737  000011  001226      TST11:  MOV     #11,TSTNO
1915  013302  012737  013372  001216              MOV     #TST12,NEXT
1916  013310  012737  013320  001220              MOV     #1$,LOCK
1917  013316  104412                              MSTCLR                  ;MASTER CLEAR DMC11
1918  013320  013701  001404              1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1919  013324  012705  000200                      MOV     #BIT7,R5        ;PUT DATA IN "EXPECTED"
1920  013330  010511                              MOV     R5,(R1)         ;WRITE BIT 7
1921  013332  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1922  013334  020504                              CMP     R5,R4           ;IS DATA CORRECT
1923  013336  001401                              BEQ     2$              ;BR IF YES
1924  013340  104002                              HLT     2               ;DATA ERROR
1925  013342  104401                      2$:     SCOP1                   ;SW09 UP?
1926  013344  012737  013352  001220              MOV     #3$,LOCK        ;NEW SCOP1
1927  013352  042711  000200              3$:     BIC     #BIT7,(R1)      ;CLEAR BIT 7
1928  013356  005005                              CLR     R5              ;CLEAR "EXPECTED"
1929  013360  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1930  013362  001402                              BEQ     4$              ;BR IF ZERO
1931  013364  104002                              HLT     2               ;DATA ERROR BIT7 NOT CLEARED
1932  013366  104401                              SCOP1                   ;SW09 UP?
1933  013370  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1934
1935
1936                                              ;*************************** TEST 12 ****************************
1937                                              ;*CONTROL STATUS REGISTER WRITE/READ TEST
1938                                              ;*SET BIT9, VERIFY BIT9 WAS SET
1939                                              ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED
1940                                              ;:************************************************************
1941
1942                                              ;  TEST 12
1943                                              ;--------------
1944  013372  012737  000012  001226      TST12:  MOV     #12,TSTNO
```

```
1945  013400  012737  013470  001216              MOV     #TST13,NEXT
1946  013406  012737  013416  001220              MOV     #1$,LOCK
1947  013414  104412                              MSTCLR                  ;MASTER CLEAR DMC11
1948  013416  013701  001404              1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1949  013422  012705  001000                      MOV     #BIT9,R5        ;PUT DATA IN "EXPECTED"
1950  013426  010511                              MOV     R5,(R1)         ;WRITE BIT 9
1951  013430  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1952  013432  020504                              CMP     R5,R4           ;IS DATA CORRECT
1953  013434  001401                              BEQ     2$              ;BR IF YES
1954  013436  104002                              HLT     2               ;DATA ERROR
1955  013440  104401                      2$:     SCOP1                   ;SW09 UP?
1956  013442  012737  013450  001220              MOV     #3$,LOCK        ;NEW SCOP1
1957  013450  042711  001000              3$:     BIC     #BIT9,(R1)      ;CLEAR BIT 9
1958  013454  005005                              CLR     R5              ;CLEAR "EXPECTED"
1959  013456  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1960  013460  001402                              BEQ     4$              ;BR IF ZERO
1961  013462  104002                              HLT     2               ;DATA ERROR BIT9 NOT CLEARED
1962  013464  104401                              SCOP1                   ;SW09 UP?
1963  013466  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1964
1965
1966                                              ;*************************** TEST 13 ***************************
1967                                              ;*CONTROL STATUS REGISTER WRITE/READ TEST
1968                                              ;*SET BIT11, VERIFY BIT11 WAS SET
1969                                              ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED
1970                                              ;;*************************************************************
1971
1972                                              ;  TEST 13
1973                                              ;--------------
1974  013470  012737  000013  001226      TST13:  MOV     #13,TSTNO
1975  013476  012737  013566  001216              MOV     #TST14,NEXT
1976  013504  012737  013514  001220              MOV     #1$,LOCK
1977  013512  104412                              MSTCLR                  ;MASTER CLEAR DMC11
1978  013514  013701  001404              1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
1979  013520  012705  004000                      MOV     #BIT11,R5       ;PUT DATA IN "EXPECTED"
1980  013524  010511                              MOV     R5,(R1)         ;WRITE BIT 11
1981  013526  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1982  013530  020504                              CMP     R5,R4           ;IS DATA CORRECT
1983  013532  001401                              BEQ     2$              ;BR IF YES
1984  013534  104002                              HLT     2               ;DATA ERROR
1985  013536  104401                      2$:     SCOP1                   ;SW09 UP?
1986  013540  012737  013546  001220              MOV     #3$,LOCK        ;NEW SCOP1
1987  013546  042711  004000              3$:     BIC     #BIT11,(R1)     ;CLEAR BIT 11
1988  013552  005005                              CLR     R5              ;CLEAR "EXPECTED"
1989  013554  011104                              MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
1990  013556  001402                              BEQ     4$              ;BR IF ZERO
1991  013560  104002                              HLT     2               ;DATA ERROR BIT11 NOT CLEARED
1992  013562  104401                              SCOP1                   ;SW09 UP?
1993  013564  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
1994
1995
1996                                              ;*************************** TEST 14 ***************************
1997                                              ;*CONTROL STATUS REGISTER WRITE/READ TEST
1998                                              ;*SET BIT12, VERIFY BIT12 WAS SET
1999                                              ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
2000                                              ;;*************************************************************
```

```
2001
2002
2003                                                    ;  TEST 14
2004  013566  012737  000014  001226                    ;-------------
2005  013574  012737  013664  001216          TST14:    MOV     #14,TSTNO
2006  013602  012737  013612  001220                    MOV     #TST15,NEXT
2007  013610  104412                                    MOV     #1$,LOCK
2008  013612  013701  001404                            MSTCLR                  ;MASTER CLEAR DMC11
2009  013616  012705  010000                    1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
2010  013622  010511                                    MOV     #BIT12,R5       ;PUT DATA IN "EXPECTED"
2011  013624  011104                                    MOV     R5,(R1)         ;WRITE BIT 12
2012  013626  020504                                    MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
2013  013630  001401                                    CMP     R5,R4           ;IS DATA CORRECT
2014  013632  104002                                    BEQ     2$              ;BR IF YES
2015  013634  104401                                    HLT     2               ;DATA ERROR
2016  013636  012737  013644  001220            2$:     SCOP1                   ;SW09 UP?
2017  013644  042711  010000                            MOV     #3$,LOCK        ;NEW SCOP1
2018  013650  005005                    3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
2019  013652  011104                            CLR     R5              ;CLEAR "EXPECTED"
2020  013654  001402                            MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
2021  013656  104002                            BEQ     4$              ;BR IF ZERO
2022  013660  104401                            HLT     2               ;DATA ERROR BIT12 NOT CLEARED
2023  013662  104400                            SCOP1                   ;SW09 UP?
2024                            4$:     SCOPE                   ;SCOPE THIS TEST
2025
2026                    ;************************** TEST 15 **************************
2027                    ;*CONTROL OUT REGISTER WRITE/READ TEST
2028                    ;*SET BIT0, VERIFY BIT0 WAS SET
2029                    ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED
2030                    ;:*********************************************************
2031
2032                                                    ;  TEST 15
2033                                                    ;-------------
2034  013664  012737  000015  001226          TST15:    MOV     #15,TSTNO
2035  013672  012737  013762  001216                    MOV     #TST16,NEXT
2036  013700  012737  013710  001220                    MOV     #1$,LOCK
2037  013706  104412                                    MSTCLR                  ;MASTER CLEAR DMC11
2038  013710  013701  001410                    1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
2039  013714  012705  000001                            MOV     #BIT0,R5        ;PUT DATA IN "EXPECTED"
2040  013720  010511                                    MOV     R5,(R1)         ;WRITE BIT 0
2041  013722  011104                                    MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2042  013724  020504                                    CMP     R5,R4           ;IS DATA CORRECT
2043  013726  001401                                    BEQ     2$              ;BR IF YES
2044  013730  104002                                    HLT     2               ;DATA ERROR
2045  013732  104401                            2$:     SCOP1                   ;SW09 UP?
2046  013734  012737  013742  001220                    MOV     #3$,LOCK        ;NEW SCOP1
2047  013742  042711  000001                    3$:     BIC     #BIT0,(R1)      ;CLEAR BIT 0
2048  013746  005005                                    CLR     R5              ;CLEAR "EXPECTED"
2049  013750  011104                                    MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2050  013752  001402                                    BEQ     4$              ;BR IF ZERO
2051  013754  104002                                    HLT     2               ;DATA ERROR BIT0 NOT CLEARED
2052  013756  104401                                    SCOP1                   ;SW09 UP?
2053  013760  104400                            4$:     SCOPE                   ;SCOPE THIS TEST
2054
2055
2056                    ;************************** TEST 16 **************************
```

```
                                                        ;*CONTROL OUT REGISTER WRITE/READ TEST
                                                        ;*SET BIT1, VERIFY BIT1 WAS SET
                                                        ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED
                                                        ;;*****************************************************************

                                                        ;  TEST 16
                                                        ;--------------
013762  012737  000016  001226          TST16:  MOV     #16,TSTNO
013770  012737  014060  001216                  MOV     #TST17,NEXT
014006  012737  014006  001220                  MOV     #1$,LOCK
014014  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
014006  013701  001410                  1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014012  012705  000002                          MOV     #BIT1,R5        ;PUT DATA IN "EXPECTED"
014016  010511                                  MOV     R5,(R1)         ;WRITE BIT 1
014020  011104                                  MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014022  020504                                  CMP     R5,R4           ;IS DATA CORRECT
014024  001401                                  BEQ     2$              ;BR IF YES
014026  104002                                  HLT     2               ;DATA ERROR
014030  104401                          2$:     SCOP1                   ;SW09 UP?
014032  012737  014040  001220                  MOV     #3$,LOCK        ;NEW SCOP1
014040  042711  000002                  3$:     BIC     #BIT1,(R1)      ;CLEAR BIT 1
014044  005005                                  CLR     R5              ;CLEAR "EXPECTED"
014046  011104                                  MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014050  001402                                  BEQ     4$              ;BR IF ZERO
014052  104002                                  HLT     2               ;DATA ERROR BIT1 NOT CLEARED
014054  104401                                  SCOP1                   ;SW09 UP?
014056  104400                          4$:     SCOPE                   ;SCOPE THIS TEST

                                                        ;************************** TEST 17 **************************
                                                        ;*CONTROL OUT REGISTER WRITE/READ TEST
                                                        ;*SET BIT2, VERIFY BIT2 WAS SET
                                                        ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED
                                                        ;;*****************************************************************

                                                        ;  TEST 17
                                                        ;--------------
014060  012737  000017  001226          TST17:  MOV     #17,TSTNO
014066  012737  014156  001216                  MOV     #TST20,NEXT
014074  012737  014104  001220                  MOV     #1$,LOCK
014102  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
014104  013701  001410                  1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014110  012705  000004                          MOV     #BIT2,R5        ;PUT DATA IN "EXPECTED"
014114  010511                                  MOV     R5,(R1)         ;WRITE BIT 2
014116  011104                                  MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014120  020504                                  CMP     R5,R4           ;IS DATA CORRECT
014122  001401                                  BEQ     2$              ;BR IF YES
014124  104002                                  HLT     2               ;DATA ERROR
014126  104401                          2$:     SCOP1                   ;SW09 UP?
014130  012737  014136  001220                  MOV     #3$,LOCK        ;NEW SCOP1
014136  042711  000004                  3$:     BIC     #BIT2,(R1)      ;CLEAR BIT 2
014142  005005                                  CLR     R5              ;CLEAR "EXPECTED"
014144  011104                                  MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014146  001402                                  BEQ     4$              ;BR IF ZERO
014150  104002                                  HLT     2               ;DATA ERROR BIT2 NOT CLEARED
014152  104401                                  SCOP1                   ;SW09 UP?
```

```
2113  014154  104400                        4$:     SCOPE                   ;SCOPE THIS TEST
2114
2115
2116                                                ;*************************** TEST 20 ****************************
2117                                                ;*CONTROL OUT REGISTER WRITE/READ TEST
2118                                                ;*SET BIT6, VERIFY BIT6 WAS SET
2119                                                ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED
2120                                                ;:************************************************************
2121
2122                                                ;  TEST 20
2123                                                ;--------------
2124  014156  012737  000020  001226        TST20:  MOV     #20,TSTNO
2125  014164  012737  014254  001216                MOV     #TST21,NEXT
2126  014172  012737  014202  001220                MOV     #1$,LOCK
2127  014200  104412                                MSTCLR                  ;MASTER CLEAR DMC11
2128  014202  013701  001410                1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
2129  014206  012705  000100                        MOV     #BIT6,R5        ;PUT DATA IN "EXPECTED"
2130  014212  010511                                MOV     R5,(R1)         ;WRITE BIT 6
2131  014214  011104                                MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2132  014216  020504                                CMP     R5,R4           ;IS DATA CORRECT
2133  014220  001401                                BEQ     2$              ;BR IF YES
2134  014222  104002                                HLT     2               ;DATA ERROR
2135  014224  104401                        2$:     SCOP1                   ;SW09 UP?
2136  014226  012737  014234  001220                MOV     #3$,LOCK        ;NEW SCOP1
2137  014234  042711  000100                3$:     BIC     #BIT6,(R1)      ;CLEAR BIT 6
2138  014240  005005                                CLR     R5              ;CLEAR "EXPECTED"
2139  014242  011104                                MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2140  014244  001402                                BEQ     4$              ;BR IF ZERO
2141  014246  104002                                HLT     2               ;DATA ERROR BIT6 NOT CLEARED
2142  014250  104401                                SCOP1                   ;SW09 UP?
2143  014252  104400                        4$:     SCOPE                   ;SCOPE THIS TEST
2144
2145
2146                                                ;*************************** TEST 21 ****************************
2147                                                ;*CONTROL OUT REGISTER WRITE/READ TEST
2148                                                ;*SET BIT7, VERIFY BIT7 WAS SET
2149                                                ;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED
2150                                                ;:************************************************************
2151
2152                                                ;  TEST 21
2153                                                ;--------------
2154  014254  012737  000021  001226        TST21:  MOV     #21,TSTNO
2155  014262  012737  014352  001216                MOV     #TST22,NEXT
2156  014270  012737  014300  001220                MOV     #1$,LOCK
2157  014276  104412                                MSTCLR                  ;MASTER CLEAR DMC11
2158  014300  013701  001410                1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
2159  014304  012705  000200                        MOV     #BIT7,R5        ;PUT DATA IN "EXPECTED"
2160  014310  010511                                MOV     R5,(R1)         ;WRITE BIT 7
2161  014312  011104                                MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2162  014314  020504                                CMP     R5,R4           ;IS DATA CORRECT
2163  014316  001401                                BEQ     2$              ;BR IF YES
2164  014320  104002                                HLT     2               ;DATA ERROR
2165  014322  104401                        2$:     SCOP1                   ;SW09 UP?
2166  014324  012737  014332  001220                MOV     #3$,LOCK        ;NEW SCOP1
2167  014332  042711  000200                3$:     BIC     #BIT7,(R1)      ;CLEAR BIT 7
2168  014336  005005                                CLR     R5              ;CLEAR "EXPECTED"
```

```
2169  014340  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2170  014342  001402                          BEQ     4$              ;BR IF ZERO
2171  014344  104002                          HLT     2               ;DATA ERROR BIT7 NOT CLEARED
2172  014346  104401                          SCOP1                   ;SW09 UP?
2173  014350  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
2174
2175                                          ;*************************** TEST 22 ****************************
2176                                          ;*CONTROL OUT REGISTER WRITE/READ TEST
2177                                          ;*SET BIT12, VERIFY BIT12 WAS SET
2178                                          ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED
2179                                          ;:****************************************************************
2180
2181                                          ;  TEST 22
2182                                          ;----------
2183
2184  014352  012737  000022  001226  TST22:  MOV     #22,TSTNO
2185  014360  012737  014450  001216          MOV     #TST23,NEXT
2186  014366  012737  014376  001220          MOV     #1$,LOCK
2187  014374  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2188  014376  013701  001410          1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
2189  014402  012705  010000                  MOV     #BIT12,R5       ;PUT DATA IN "EXPECTED"
2190  014406  010511                          MOV     R5,(R1)         ;WRITE BIT 12
2191  014410  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2192  014412  020504                          CMP     R5,R4           ;IS DATA CORRECT
2193  014414  001401                          BEQ     2$              ;BR IF YES
2194  014416  104002                          HLT     2               ;DATA ERROR
2195  014420  104401                  2$:     SCOP1                   ;SW09 UP?
2196  014422  012737  014430  001220          MOV     #3$,LOCK        ;NEW SCOP1
2197  014430  042711  010000          3$:     BIC     #BIT12,(R1)     ;CLEAR BIT 12
2198  014434  005005                          CLR     R5              ;CLEAR "EXPECTED"
2199  014436  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2200  014440  001402                          BEQ     4$              ;BR IF ZERO
2201  014442  104002                          HLT     2               ;DATA ERROR BIT12 NOT CLEARED
2202  014444  104401                          SCOP1                   ;SW09 UP?
2203  014446  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
2204
2205                                          ;*************************** TEST 23 ****************************
2206                                          ;*CONTROL OUT REGISTER WRITE/READ TEST
2207                                          ;*SET BIT13, VERIFY BIT13 WAS SET
2208                                          ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED
2209                                          ;:****************************************************************
2210
2211                                          ;  TEST 23
2212                                          ;----------
2213
2214  014450  012737  000023  001226  TST23:  MOV     #23,TSTNO
2215  014456  012737  014546  001216          MOV     #TST24,NEXT
2216  014464  012737  014474  001220          MOV     #1$,LOCK
2217  014472  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2218  014474  013701  001410          1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
2219  014500  012705  020000                  MOV     #BIT13,R5       ;PUT DATA IN "EXPECTED"
2220  014504  010511                          MOV     R5,(R1)         ;WRITE BIT 13
2221  014506  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2222  014510  020504                          CMP     R5,R4           ;IS DATA CORRECT
2223  014512  001401                          BEQ     2$              ;BR IF YES
2224  014514  104002                          HLT     2               ;DATA ERROR
```

```
2225  014516  104401                          2$:     SCOP1                   ;SW09 UP?
2226  014520  012737  014526  001220                  MOV     #3$,LOCK        ;NEW SCOP1
2227  014526  042711  020000                  3$:     BIC     #BIT13,(R1)     ;CLEAR BIT 13
2228  014532  005005                                  CLR     R5              ;CLEAR "EXPECTED"
2229  014534  011104                                  MOV     (R1),R4         ;READ CONTROL OUT REGISTER
2230  014536  001402                                  BEQ     4$              ;BR IF ZERO
2231  014540  104002                                  HLT     2               ;DATA ERROR BIT13 NOT CLEARED
2232  014542  104401                                  SCOP1                   ;SW09 UP?
2233  014544  104400                          4$:     SCOPE                   ;SCOPE THIS TEST
2234
2235
2236                                                  ;************************** TEST 24 **************************
2237                                                  ;*PORT4 REGISTER WRITE/READ TEST
2238                                                  ;*FLOAT A ONE THROUGH PORT4 REGISTER
2239                                                  ;*FLOAT A ZERO THROUGH PORT4 REGISTER
2240                                                  ;:************************************************************
2241
2242                                                  ;  TEST 24
2243                                                  ;--------------
2244  014546  012737  000024  001226          TST24:  MOV     #24,TSTNO
2245  014554  012737  014672  001216                  MOV     #TST25,NEXT
2246  014562  012737  014602  001220                  MOV     #64$,LOCK
2247  014570  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
2248  014572  013701  001412                          MOV     DMP04,R1                ;PUT REGISTER ADDRESS IN R1
2249  014576  012700  000001                          MOV     #1,R0           ;START WITH BIT0
2250  014602                                  64$:
2251  014602  010005                                  MOV     R0,R5           ;PUT "EXPECTED" IN R5
2252  014604  010511                                  MOV     R5,(R1)         ;WRITE PORT4 REGISTER
2253  014606  011104                                  MOV     (R1),R4         ;READ PORT4 REGISTER
2254  014610  020504                                  CMP     R5,R4           ;COMPARE EXPECTED AND FOUND
2255  014612  001401                                  BEQ     65$             ;BR IF OK
2256  014614  104002                                  HLT     2               ;WRITE/READ ERROR
2257  014616  104401                          65$:    SCOP1                   ;LOOP TO 64$ IF SW09=1
2258  014620  000241                                  CLC                     ;CLEAR CARRY
2259  014622  006100                                  ROL     R0              ;SHIFT TO NEXT BIT
2260  014624  001366                                  BNE     64$             ;BR IF NOT DONE YET
2261  014626  012737  014640  001220                  MOV     #66$,LOCK       ;NEW SCOP1
2262  014634  012700  000001                          MOV     #1,R0           ;START WITH BIT0
2263  014640                                  66$:
2264  014640  005100                                  COM     R0              ;CHANGE TO A FLOATING ZERO
2265  014642  010005                                  MOV     R0,R5           ;PUT "EXPECTED" IN R5
2266  014644  010511                                  MOV     R5,(R1)         ;WRITE PORT4 REGISTER
2267  014646  011104                                  MOV     (R1),R4         ;READ PORT4 REGISTER
2268  014650  020504                                  CMP     R5,R4           ;COMPARE EXPECTED AND FOUND
2269  014652  001401                                  BEQ     67$             ;BR IF OK
2270  014654  104002                                  HLT     2               ;WRITE/READ ERROR
2271  014656  104401                          67$:    SCOP1                   ;LOOP TO 66$ IF SW09=1
2272  014660  005100                                  COM     R0              ;CHANGE BACK TO A FLOATING ONE
2273  014662  000241                                  CLC                     ;CLEAR CARRY
2274  014664  006100                                  ROL     R0              ;SHIFT TO NEXT BIT
2275  014666  001364                                  BNE     66$             ;BR IF NOT DONE YET
2276  014670  104400                                  SCOPE                   ;SCOPE THIS TEST
2277
2278
2279                                                  ;************************** TEST 25 **************************
2280                                                  ;*PORT6 REGISTER WRITE/READ TEST
```

```
2281                                            ;*FLOAT A ONE THROUGH PORT6 REGISTER
2282                                            ;*FLOAT A ZERO THROUGH PORT6 REGISTER
2283                                            ;;*****************************************************************
2284
2285                                            ;  TEST 25
2286                                            ;--------------
2287  014672  012737  000025  001226    TST25:  MOV     #25,TSTNO
2288  014700  012737  015016  001216            MOV     #TST26,NEXT
2289  014706  012737  014726  001220            MOV     #64$,LOCK
2290  014714  104412                            MSTCLR                          ;MASTER CLEAR DMC11
2291  014716  013701  001414                    MOV     DMP06,R1                ;PUT REGISTER ADDRESS IN R1
2292  014722  012700  000001                    MOV     #1,R0                   ;START WITH BIT0
2293  014726                            64$:
2294  014726  010005                            MOV     R0,R5                   ;PUT "EXPECTED" IN R5
2295  014730  010511                            MOV     R5,(R1)                 ;WRITE PORT6 REGISTER
2296  014732  011104                            MOV     (R1),R4                 ;READ PORT6 REGISTER
2297  014734  020504                            CMP     R5,R4                   ;COMPARE EXPECTED AND FOUND
2298  014736  001401                            BEQ     65$                     ;BR IF OK
2299  014740  104002                            HLT     2                       ;WRITE/READ ERROR
2300  014742  104401                    65$:    SCOP1                           ;LOOP TO 64$ IF SW09=1
2301  014744  000241                            CLC                             ;CLEAR CARRY
2302  014746  006100                            ROL     R0                      ;SHIFT TO NEXT BIT
2303  014750  001366                            BNE     64$                     ;BR IF NOT DONE YET?
2304  014752  012737  014764  001220            MOV     #66$,LOCK               ;NEW SCOP1
2305  014760  012700  000001                    MOV     #1,R0                   ;START WITH BIT0
2306  014764                            66$:
2307  014764  005100                            COM     R0                      ;CHANGE TO A FLOATING ZERO
2308  014766  010005                            MOV     R0,R5                   ;PUT "EXPECTED" IN R5
2309  014770  010511                            MOV     R5,(R1)                 ;WRITE PORT6 REGISTER
2310  014772  011104                            MOV     (R1),R4                 ;READ PORT6 REGISTER
2311  014774  020504                            CMP     R5,R4                   ;COMPARE EXPECTED AND FOUND
2312  014776  001401                            BEQ     67$                     ;BR IF OK
2313  015000  104002                            HLT     2                       ;WRITE/READ ERROR
2314  015002  104401                    67$:    SCOP1                           ;LOOP TO 66$ IF SW09=1
2315  015004  005100                            COM     R0                      ;CHANGE BACK TO A FLOATING ONE
2316  015006  000241                            CLC                             ;CLEAR CARRY
2317  015010  006100                            ROL     R0                      ;SHIFT TO NEXT BIT
2318  015012  001364                            BNE     66$                     ;BR IF NOT DONE YET?
2319  015014  104400                            SCOPE                           ;SCOPE THIS TEST
2320
2321
2322                                            ;*************************** TEST 26 ****************************
2323                                            ;*UNIBUS REGISTER BYTE DUAL ADDRESSING TEST
2324                                            ;*LOAD ALL REGISTERS WITH INCREMENTING PATTERN
2325                                            ;*READ BACK ALL REGISTERS TO VERIFY CORRECT ADDRESSING
2326                                            ;;*****************************************************************
2327
2328                                            ;  TEST 26
2329                                            ;--------------
2330  015016  012737  000026  001226    TST26:  MOV     #26,TSTNO
2331  015024  012737  015146  001216            MOV     #TST27,NEXT
2332  015032  012737  015046  001220            MOV     #1$,LOCK
2333                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
2334  015040  104412                            MSTCLR                          ;MASTER CLEAR DMC11
2335  015042  012700  000001                    MOV     #1,R0                   ;START PATTERN AT 1
2336  015046  105011                    1$:     CLRB    (R1)                    ;CLEAR REGISTER
```

```
2337  015050  110005                          MOVB    R0,R5           ;PUT DATA IN "EXPECTED"
2338  015052  110011                          MOVB    R0,(R1)         ;WRITE DMC REGISTER WITH PATTERN
2339  015054  111104                          MOVB    (R1),R4         ;READ DMC REGISTER INTO "FOUND"
2340  015056  020504                          CMP     R5,R4           ;IS DATA CORRECT
2341  015060  001401                          BEQ     2$              ;BR IF YES
2342  015062  104002                          HLT     2               ;DATA ERROR
2343  015064  104401                  2$:     SCOP1                   ;SW09=1?
2344  015066  105721                          TSTB    (R1)+           ;NEXT REGISTER
2345  015070  005200                          INC     R0              ;INCREMENT DATA PATTERN
2346  015072  022700  000011                  CMP     #11,R0          ;LAST REGISTER?
2347  015076  001363                          BNE     1$              ;BR IF NO
2348  015100  013701  001404                  MOV     DMCSR,R1        ;BASE DMC11  ADDRESS TO R1
2349  015104  012700  000001                  MOV     #1,R0           ;RESTART PATTERN AT 1
2350  015110  012737  015116  001220          MOV     #3$,LOCK        ;NEW SCOP1
2351  015116  110005                  3$:     MOVB    R0,R5           ;PUT DATA IN "EXPECTED"
2352  015120  111104                          MOVB    (R1),R4         ;READ DMC REGISTER INTO "FOUND"
2353  015122  020504                          CMP     R5,R4           ;IS DATA CORRECT
2354  015124  001401                          BEQ     4$              ;BR IF YES
2355  015126  104002                          HLT     2               ;DUAL ADDRESSING ERROR
2356  015130  104401                  4$:     SCOP1                   ;SW09=1?
2357  015132  105721                          TSTB    (R1)+           ;NEXT REGISTER
2358  015134  005200                          INC     R0              ;INCREMENT PATTERN
2359  015136  022700  000011                  CMP     #11,R0          ;LAST REGISTER?
2360  015142  001365                          BNE     3$              ;BR IF NO
2361  015144  104400                          SCOPE                   ;SCOPE THIS TEST
2362
2363
2364                                          ;*************************** TEST 27 ****************************
2365                                          ;*MAINTENANCE INSTRUCTION REGISTER TEST
2366                                          ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
2367                                          ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A BUS RESET.
2368                                          ;*****************************************************************
2369
2370                                          ;  TEST 27
2371                                          ;--------------
2372  015146  012737  000027  001226  TST27:  MOV     #27,TSTNO
2373  015154  012737  015306  001216          MOV     #TST30,NEXT
2374  015162  012737  015200  001220          MOV     #1$,LOCK
2375                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2376  015170  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2377  015172  012711  003000                  MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
2378  015176  005005                          CLR     R5              ;PUT "EXPECTED" IN R5
2379  015200  010561  000006          1$:     MOV     R5,6(R1)        ;CLEAR THE IR
2380  015204  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2381  015210  020504                          CMP     R5,R4           ;IS IT CLEARED?
2382  015212  001401                          BEQ     2$              ;BR IF YES
2383  015214  104023                          HLT     23              ;ERROR IR IS NOT CLEAR
2384  015216  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2385  015220  012737  015232  001220          MOV     #3$,LOCK        ;NEW SCOP1
2386  015226  012705  177777                  MOV     #-1,R5          ;PUT "EXPECTED" IN R5
2387  015232  010561  000006          3$:     MOV     R5,6(R1)        ;WRITE ALL ONES TO THE IR
2388  015236  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2389  015242  020504                          CMP     R5,R4           ;IS IT ALL ONES?
2390  015244  001401                          BEQ     4$              ;BR IF YES
2391  015246  104023                          HLT     23              ;ERROR IR IS NOT = ALL ONES
2392  015250  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
```

```
2393  015252  012737  015262  001220          MOV     #5$,LOCK        ;NEW SCOPE
2394  015260  005005                          CLR     R5              ;PUT "EXPECTED" IN R5
2395  015262  000005                  5$:     RESET                   ;BUS RESET
2396  015264  012711  003000                  MOV     #BIT9!BIT10,(R1) ;SEL6 IS IR
2397  015270  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2398  015274  020504                          CMP     R5,R4           ;IS IT CLEARED?
2399  015276  001401                          BEQ     6$              ;BR IF YES
2400  015300  104023                          HLT     23              ;ERROR, IR IS NOT CLEARED
2401  015302  104401                  6$:     SCOPE                   ;LOOP TO 5$ IF SW09=1
2402  015304  104400                          SCOPE                   ;SCOPE THIS TEST
2403
2404
2405                                          ;************************** TEST 30 **************************
2406                                          ;*MAINTENANCE INSTRUCTION REGISTER TEST
2407                                          ;*VERIFY THAT THE MAINT IR CAN BE WRITTEN TO ALL ZEROS'
2408                                          ;*AND ALL ONES'. VERIFY THAT IT IS CLEARED ON A MASTER RESET.
2409                                          ;;*********************************************************************
2410
2411                                          ;  TEST 30
2412                                          ;-------------
2413  015306  012737  000030  001226  TST30:  MOV     #30,TSTNO
2414  015314  012737  015450  001216          MOV     #TST31,NEXT
2415  015322  012737  015340  001220          MOV     #1$,LOCK
2416                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2417  015330  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2418  015332  012711  003000                  MOV     #BIT9!BIT10,(R1) ;SEL6 IS NOW THE IR
2419  015336  005005                          CLR     R5              ;PUT "EXPECTED" IN R5
2420  015340  010561  000006          1$:     MOV     R5,6(R1)        ;CLEAR THE IR
2421  015344  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2422  015350  020504                          CMP     R5,R4           ;IS IT CLEARED?
2423  015352  001401                          BEQ     2$              ;BR IF YES
2424  015354  104023                          HLT     23              ;ERROR IR IS NOT CLEAR
2425  015356  104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
2426  015360  012737  015372  001220          MOV     #3$,LOCK        ;NEW SCOP1
2427  015366  012705  177777                  MOV     #-1,R5          ;PUT "EXPECTED" IN R5
2428  015372  010561  000006          3$:     MOV     R5,6(R1)        ;WRITE ALL ONES TO THE IR
2429  015376  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2430  015402  020504                          CMP     R5,R4           ;IS IT ALL ONES?
2431  015404  001401                          BEQ     4$              ;BR IF YES
2432  015406  104023                          HLT     23              ;ERROR IR IS NOT = ALL ONES
2433  015410  104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
2434  015412  012737  015422  001220          MOV     #5$,LOCK        ;NEW SCOP1
243   015420  005005                          CLR     R5              ;PUT "EXPECTED" IN R5
2436  015422  052711  040000          5$:     BIS     #BIT14,(R1)     ;MASTER CLEAR
2437  015426  012711  003000                  MOV     #BIT9!BIT10,(R1) ;SEL6 IS IR
2438  015432  016104  000006                  MOV     6(R1),R4        ;READ THE IR
2439  015436  020504                          CMP     R5,R4           ;IS IT CLEARED?
2440  015440  001401                          BEQ     6$              ;BR IF YES
2441  015442  104023                          HLT     23              ;ERROR, IR IS NOT CLEARED
2442  015444  104401                  6$:     SCOP1                   ;LOOP TO 5$ IF SW09=1
2443  015446  104400                          SCOPE                   ;SCOPE THIS TEST
2444
2445
2446                                          ;************************** TEST 31 **************************
2447                                          ;*MICRO PROCESSOR TEST
2448                                          ;*LOAD DMP06 WITH A MICRO-PROCESSOR INSTRUCTION, CLOCK IT
```

```
2449                                                    ;*VERIFY INSTRUCTION EXECUTED PROPERLY
2450                                                    ;*INSTRUCTION SHOULD MOVE IBUS*4 TO IBUS*5, IBUS*4 IS ALL 1'S
2451                                                    ;*AND IBUS*5 IS ALL 0'S. RESULT SHOULD BE ALL 1'S IN SEL4
2452                                                    ;;**************************************************************
2453
2454                                                    ;  TEST 31
2455                                                    ;--------------
2456  015450  012737  000031  001226          TST31:  MOV     #31,TSTNO
2457  015456  012737  015534  001216                  MOV     #TST32,NEXT
2458                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
2459  015464  104412                                   MSTCLR                ;MASTER CLEAR DMC11
2460  015466  012761  000377  000004                  MOV     #377,4(R1)    ;PORT4 HI-BYTE=0'S LO-BYTE=1'S
2461  015474  012711  001000                          MOV     #BIT9,(R1)    ;SET ROMI
2462  015500  012761  121105  000006                  MOV     #121105,6(R1) ;INSTRUCTION TO PORT6
2463  015506  052711  001400                          BIS     #BIT8!BIT9,(R1) ;CLK INTSRUCTION, MOVE IBUS*4 TO IBUS*5
2464  015512  000240                                   NOP                   ;
2465  015514  012705  177777                          MOV     #-1,R5        ;PUT "EXPECTED" IN R5
2466  015520  016104  000004                          MOV     4(R1),R4      ;PUT "FOUND" INTO R4
2467  015524  020504                                   CMP     R5,R4         ;IS DATA CORRECT
2468  015526  001401                                   BEQ     1$            ;BR IF YES
2469  015530  104003                                   HLT     3             ;ERROR
2470  015532  104400                          1$:     SCOPE                 ;SCOPE THIS TEST
2471
2472                                                    ;*************************** TEST 32 ****************************
2473                                                    ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2474                                                    ;*FLOAT A 1 THROUGH IBUS* REGISTER 0
2475                                                    ;*FLOAT A 0 THROUGH IBUS* REGISTER 0
2476                                                    ;;**************************************************************
2477
2478                                                    ;  TEST 32
2479                                                    ;--------------
2480  015534  012737  000032  001226          TST32:  MOV     #32,TSTNO
2481  015542  012737  015734  001216                  MOV     #TST33,NEXT
2482  015550  012737  015570  001220                  MOV     #64$,LOCK
2483                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
2484  015556  104412                                   MSTCLR                ;MASTER CLEAR DMC11
2485  015560  012702  000000                          MOV     #0,R2         ;SAVE REGISTER ADDRESS FOR TYPEOUT
2486  015564  012700  000001                          MOV     #1,R0         ;START WITH BIT 0
2487  015570                                  64$:
2488  015570  010061  000004                          MOV     R0,4(R1)      ;PUT PATTERN INTO PORT4
2489  015574  042761  000030  000004                  BIC     #30,4(R1)     ;CLEAR UNWANTED BITS
2490  015602  104414                                   ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2491  015604  121100                                   121100!0              ;MOV DATA TO IBUS* REGISTER 0
2492  015606  104414                                   ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2493  015610  121005                                   121005!(0*20)         ;READ FROM IBUS* REGISTER 0
2494  015612  010005                                   MOV     R0,R5         ;PUT EXPECTED IN R5
2495  015614  042705  000030                          BIC     #30,R5        ;CLEAR UNWANTED BITS
2496  015620  116104  000005                          MOVB    5(R1),R4      ;PUT "FOUND" INTO R4
2497  015624  120504                                   CMPB    R5,R4         ;DATA CORRECT?
2498  015626  001401                                   BEQ     65$           ;BR IF YES
2499  015630  104004                                   HLT     4             ;ERROR
2500  015632  104401                          65$:     SCOP1                 ;SW09=1?
2501  015634  000241                                   CLC                   ;CLEAR CARRY
2502  015636  106100                                   ROLB    R0            ;SHIFT BIT IN R0
2503  015640  001353                                   BNE     64$           ;IF R0=0 THEN DONE
```

```
2505  015642  012737  015656  001220          MOV     #67$,LOCK       ;NEW SCOPE
2506  015650  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2507  015654  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2508  015656                          67$:
2509  015656  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2510  015662  042761  000030  000004          BIC     #30,4(R1)       ;CLEAR UNWANTED BITS
2511  015670  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2512  015672  121100                          121100!0                ;MOV DATA TO IBUS* REGISTER 0
2513  015674  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2514  015676  121005                          121005!(0*20)           ;READ FROM IBUS* REGISTER 0
2515  015700  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2516  015702  042705  000030                  BIC     #30,R5          ;CLEAR UNWANTED BITS
2517  015706  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2518  015712  120504                          CMPB    R5,R4           ;DATA CORRECT?
2519  015714  001401                          BEQ     68$             ;BR IF YES
2520  015716  104004                          HLT     4               ;ERROR
2521  015720  104401                  68$:    SCOP1                   ;SW09=1?
2522  015722  005100                          COM     R0              ;CHANGE TO FLOATING 1
2523  015724  000241                          CLC                     ;CLEAR CARRY
2524  015726  106100                          ROLB    R0              ;SHIFT BIT IN R0
2525  015730  001351                          BNE     69$             ;IF R0=0 THEN DONE
2526  015732  104400                          SCOPE                   ;SCOPE THIS TEST
2527
2528
2529                                          ;*************************** TEST 33 ****************************
2530                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2531                                          ;*FLOAT A 1 THROUGH IBUS* REGISTER 2
2532                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 2
2533                                          ;;*************************************************************
2534
2535                                          ;  TEST 33
2536                                          ;--------------
2537  015734  012737  000033  001226  TST33:  MOV     #33,TSTNO
2538  015742  012737  016134  001216          MOV     #TST34,NEXT
2539  015750  012737  015770  001220          MOV     #64$,LOCK
2540                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2541  015756  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2542  015760  012702  000002                  MOV     #2,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2543  015764  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2544  015770                          64$:
2545  015770  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2546  015774  042761  000070  000004          BIC     #70,4(R1)       ;CLEAR UNWANTED BITS
2547  016002  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2548  016004  121102                          121100!2                ;MOV DATA TO IBUS* REGISTER 2
2549  016006  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2550  016010  121045                          121005!(2*20)           ;READ FROM IBUS* REGISTER 2
2551  016012  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2552  016014  042705  000070                  BIC     #70,R5          ;CLEAR UNWANTED BITS
2553  016020  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2554  016024  120504                          CMPB    R5,R4           ;DATA CORRECT?
2555  016026  001401                          BEQ     65$             ;BR IF YES
2556  016030  104004                          HLT     4               ;ERROR
2557  016032  104401                  65$:    SCOP1                   ;SW09=1?
2558  016034  000241                          CLC                     ;CLEAR CARRY
2559  016036  106100                          ROLB    R0              ;SHIFT BIT IN R0
2560  016040  001353                          BNE     64$             ;IF R0=0 THEN DONE
```

```
2561  016042  012737  016056  001220                  MOV     #67$,LOCK       ;NEW SCOP1
2562  016050  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
2563  016054  005100                          69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2564  016056                                  67$:
2565  016056  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2566  016062  042761  000070  000004                  BIC     #70,4(R1)       ;CLEAR UNWANTED BITS
2567  016070  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2568  016072  121102                                  121100!2                ;MOV DATA TO IBUS* REGISTER 2
2569  016074  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2570  016076  121045                                  121005!<2*20>           ;READ FROM IBUS* REGISTER 2
2571  016100  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
2572  016102  042705  000070                          BIC     #70,R5          ;CLEAR UNWANTED BITS
2573  016106  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2574  016112  .20504                                  CMPB    R5,R4           ;DATA CORRECT?
2575  016114  001401                                  BEQ     68$             ;BR IF YES
2576  016116  104004                                  HLT     4               ;ERROR
2577  016120  104401                          68$:    SCOP1                   ;SW09=1?
2578  016122  005100                                  COM     R0              ;CHANGE TO FLOATING 1
2579  016124  000241                                  CLC                     ;CLEAR CARRY
2580  016126  106100                                  ROLB    R0              ;SHIFT BIT IN R0
2581  016130  001351                                  BNE     69$             ;IF R0=0 THEN DONE
2582  016132  104400                                  SCOPE                   ;SCOPE THIS TEST
2583
2584
2585                                                  ;************************** TEST 34 **************************
2586                                                  ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2587                                                  ;*FLOAT A 1 THROUGH IBUS* REGISTER 4
2588                                                  ;*FLOAT A 0 THROUGH IBUS* REGISTER 4
2589                                                  ;:************************************************************
2590
2591                                                  ;  TEST 34
2592                                                  ;--------------
2593  016134  012737  000034  001226          TST34:  MOV     #34,TSTNO
2594  016142  012737  016310  001216                  MOV     #TST35,NEXT
2595  016150  012737  016170  001220                  MOV     #64$,LOCK
2596                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
2597  016156  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
2598  016160  012702  000004                          MOV     #4,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2599  016164  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
2600  016170                                  64$:
2601  016170  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2602  016174  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2603  016176  121104                                  121100!4                ;MOV DATA TO IBUS* REGISTER 4
2604  016200  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2605  016202  121105                                  121005!<4*20>           ;READ FROM IBUS* REGISTER 4
2606  016204  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
2607  016206  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2608  016212  120504                                  CMPB    R5,R4           ;DATA CORRECT?
2609  016214  001401                                  BEQ     65$             ;BR IF YES
2610  016216  104004                                  HLT     4               ;ERROR
2611  016220  104401                          65$:    SCOP1                   ;SW09=1?
2612  016222  000241                                  CLC                     ;CLEAR CARRY
2613  016224  106100                                  ROLB    R0              ;SHIFT BIT IN R0
2614  016226  001360                                  BNE     64$             ;IF R0=0 THEN DONE
2615  016230  012737  016244  001220                  MOV     #67$,LOCK       ;NEW SCOP1
2616  016236  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
```

```
2617  016242  005100                          69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2618  016244                                  67$:
2619  016244  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2620  016250  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2621  016252  121104                                121100!4                ;MOV DATA TO IBUS* REGISTER 4
2622  016254  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2623  016256  121105                                121005!(4*20)           ;READ FROM IBUS* REGISTER 4
2624  016260  010005                                MOV     R0,R5           ;PUT EXPECTED IN R5
2625  016262  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2626  016266  120504                                CMPB    R5,R4           ;DATA CORRECT?
2627  016270  001401                                BEQ     68$             ;BR IF YES
2628  016272  104004                                HLT     4               ;ERROR
2629  016274  10440:                         68$:    SCOP1                   ;SW09=1?
2630  016276  005100                                COM     R0              ;CHANGE TO FLOATING 1
2631  016300  000241                                CLC                     ;CLEAR CARRY
2632  016302  106100                                ROLB    R0              ;SHIFT BIT IN R0
2633  016304  001356                                BNE     69$             ;IF R0=0 THEN DONE
2634  016306  104400                                SCOPE                   ;SCOPE THIS TEST
2635
2636
2637                                                ;************************** TEST 35 **************************
2638                                                ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2639                                                ;*FLOAT A 1 THROUGH IBUS* REGISTER 5
2640                                                ;*FLOAT A 0 THROUGH IBUS* REGISTER 5
2641                                                ;;*********************************************************************
2642
2643                                                ;  TEST 35
2644                                                ;--------------
2645  016310  012737  000035  001226        TST35:  MOV     #35,TSTNO
2646  016316  012737  016464  001216                MOV     #TST36,NEXT
2647  016324  012737  016344  001220                MOV     #64$,LOCK
2648                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
2649  016332  104412                                MSTCLR                  ;MASTER CLEAR DMC11
2650  016334  012702  000005                        MOV     #5,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2651  016340  012700  000001                        MOV     #1,R0           ;START WITH BIT 0
2652  016344                                  64$:
2653  016344  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2654  016350  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2655  016352  121105                                121100!5                ;MOV DATA TO IBUS* REGISTER 5
2656  016354  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2657  016356  121125                                121005!(5*20)           ;READ FROM IBUS* REGISTER 5
2658  016360  010005                                MOV     R0,R5           ;PUT EXPECTED IN R5
2659  016362  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2660  016366  120504                                CMPB    R5,R4           ;DATA CORRECT?
2661  016370  001401                                BEQ     65$             ;BR IF YES
2662  016372  104004                                HLT     4               ;ERROR
2663  016374  104401                         65$:    SCOP1                   ;SW09=1?
2664  016376  000241                                CLC                     ;CLEAR CARRY
2665  016400  106100                                ROLB    R0              ;SHIFT BIT IN R0
2666  016402  001360                                BNE     64$             ;IF R0=0 THEN DONE
2667  016404  012737  016420  001220                MOV     #67$,LOCK       ;NEW SCOP1
2668  016412  012700  000001                        MOV     #1,R0           ;START WITH BIT 0
2669  016416  005100                          69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2670  016420                                  67$:
2671  016420  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2672  016424  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
2673  016426  121105                          121100!5            ;MOV DATA TO IBUS* REGISTER 5
2674  016430  104414                          ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2675  016432  121125                          121005!(5*20)       ;READ FROM IBUS* REGISTER 5
2676  016434  010005                          MOV     R0,R5       ;PUT EXPECTED IN R5
2677  016436  116104  000005                  MOVB    5(R1),R4    ;PUT "FOUND" INTO R4
2678  016442  120504                          CMPB    R5,R4       ;DATA CORRECT?
2679  016444  001401                          BEQ     68$         ;BR IF YES
2680  016446  104004                          HLT     4           ;ERROR
2681  016450  104401                  68$:    SCOP1               ;SW09=1?
2682  016452  005100                          COM     R0          ;CHANGE TO FLOATING 1
2683  016454  000241                          CLC                 ;CLEAR CARRY
2684  016456  106100                          ROLB    R0          ;SHIFT BIT IN R0
2685  016460  001356                          BNE     69$         ;IF R0=0 THEN DONE
2686  016462  104400                          SCOPE               ;SCOPE THIS TEST
2687
2688
2689                                          ;*************************** TEST 36 ***************************
2690                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2691                                          ;*FLOAT A 1 THROUGH IBUS* REGISTER 10
2692                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 10
2693                                          ;*THE NPR RQ BIT (BIT 0) IS MASKED DURING THIS TEST
2694                                          ;:***********************************************************
2695
2696                                          ;  TEST 36
2697                                          ;--------------
2698  016464  012737  000036  001226  TST36:  MOV     #36,TSTNO
2699  016472  012737  016664  001216          MOV     #TST37,NEXT
2700  016500  012737  016520  001220          MOV     #64$,LOCK
2701                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
2702  016506  104412                          MSTCLR              ;MASTER CLEAR DMC11
2703  016510  012702  000010                  MOV     #10,R2      ;SAVE REGISTER ADDRESS FOR TYPEOUT
2704  016514  012700  000001                  MOV     #1,R0       ;START WITH BIT 0
2705  016520                          64$:
2706  016520  010061  000004                  MOV     R0,4(R1)    ;PUT PATTERN INTO PORT4
2707  016524  042761  000141  000004          BIC     #141,4(R1)  ;CLEAR UNWANTED BITS
2708  016532  104414                          ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2709  016534  121110                          121100!10           ;MOV DATA TO IBUS* REGISTER 10
2710  016536  104414                          ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2711  016540  121205                          121005!(10*20)      ;READ FROM IBUS* REGISTER 10
2712  016542  010005                          MOV     R0,R5       ;PUT EXPECTED IN R5
2713  016544  042705  000141                  BIC     #141,R5     ;CLEAR UNWANTED BITS
2714  016550  116104  000005                  MOVB    5(R1),R4    ;PUT "FOUND" INTO R4
2715  016554  120504                          CMPB    R5,R4       ;DATA CORRECT?
2716  016556  001401                          BEQ     65$         ;BR IF YES
2717  016560  104004                          HLT     4           ;ERROR
2718  016562  104401                  65$:    SCOP1               ;SW09=1?
2719  016564  000241                          CLC                 ;CLEAR CARRY
2720  016566  106100                          ROLB    R0          ;SHIFT BIT IN R0
2721  016570  001353                          BNE     64$         ;IF R0=0 THEN DONE
2722  016572  012737  016606  001220          MOV     #67$,LOCK   ;NEW SCOP1
2723  016600  012700  000001                  MOV     #1,R0       ;START WITH BIT 0
2724  016604  005100                  69$:    COM     R0          ;CHANGE TO FLOATING ZERO
2725  016606                          67$:
2726  016606  010061  000004                  MOV     R0,4(R1)    ;PUT PATTERN INTO PORT4
2727  016612  042761  000141  000004          BIC     #141,4(R1)  ;CLEAR UNWANTED BITS
2728  016620  104414                          ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
2729  016622  121110                          121100!10               ;MOV DATA TO IBUS* REGISTER 10
2730  016624  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2731  016626  121205                          121005!<10*20>          ;READ FROM IBUS* REGISTER 10
2732  016630  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2733  016632  042705  000141                  BIC     #141,R5         ;CLEAR UNWANTED BITS
2734  016636  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2735  016642  120504                          CMPB    R5,R4           ;DATA CORRECT?
2736  016644  001401                          BEQ     68$             ;BR IF YES
2737  016646  104004                          HLT     4               ;ERROR
2738  016650  104401                  68$:    SCOP1                   ;SW09=1?
2739  016652  005100                          COM     R0              ;CHANGE TO FLOATING 1
2740  016654  000241                          CLC                     ;CLEAR CARRY
2741  016656  106100                          ROLB    R0              ;SHIFT BIT IN R0
2742  016660  001351                          BNE     69$             ;IF R0=0 THEN DONE
2743  016662  104400                          SCOPE                   ;SCOPE THIS TEST
2744
2745                                          ;*************************** TEST 37 ****************************
2746                                          ;*MICRO PROCESSOR IBUS* REGISTER WRITE/READ TEST
2747                                          ;*FLOAT A 1 THROUGH IBUS* REGISTER 11
2748                                          ;*FLOAT A 0 THROUGH IBUS* REGISTER 11
2749                                          ;*THE BR RQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
2750                                          ;*(BITS 7,4,1) ARE ALL MASKED DURING THIS TEST
2751                                          ;:****************************************************************
2752
2753                                          ;  TEST 37
2754                                          ;--------------
2755
2756  016664  012737  000037  001226  TST37:  MOV     #37,TSTNO
2757  016672  012737  017104  001216          MOV     #TST40,NEXT
2758  016700  012737  016720  001220          MOV     #64$,LOCK
2759
2760  016706  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2761  016710  012702  000011                  MOV     #11,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
2762  016714  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2763  016720                          64$:
2764  016720  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2765  016724  042761  000262  000004          BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
2766  016732  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2767  016734  121111                          121100!11               ;MOV DATA TO IBUS* REGISTER 11
2768  016736  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2769  016740  121225                          121005!<11*20>          ;READ FROM IBUS* REGISTER 11
2770  016742  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2771  016744  042705  000262                  BIC     #262,R5         ;CLEAR UNWANTED BITS
2772  016750  052705  000020                  BIS     #20,R5          ;ADD THESE BITS
2773  016754  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2774  016760  052704  000020                  BIS     #20,R4          ;ADD THIS BIT
2775  016764  120504                          CMPB    R5,R4           ;DATA CORRECT?
2776  015766  001401                          BEQ     65$             ;BR IF YES
2777  016770  104004                          HLT     4               ;ERROR
2778  016772  104401                  65$:    SCOP1                   ;SW09=1?
2779  016774  000241                          CLC                     ;CLEAR CARRY
2780  016776  106100                          ROLB    R0              ;SHIFT BIT IN R0
2781  017000  001347                          BNE     64$             ;IF R0=0 THEN DONE
2782  017002  012737  017016  001220          MOV     #67$,LOCK       ;NEW SCOP1
2783  017010  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2784  017014  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
```

```
2785  017016                                  67$:
2786  017016  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2787  017022  042761  000262  000004                BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
2788  017030  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2789  017032  121111                                121100!11               ;MOV DATA TO IBUS* REGISTER 11
2790  017034  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2791  017036  121225                                121005!(11*20)          ;READ FROM IBUS* REGISTER 11
2792  017040  010005                                MOV     R0,R5           ;PUT EXPECTED IN R5
2793  017042  042705  000262                       BIC     #262,R5         ;CLEAR UNWANTED BITS
2794  017046  052705  000020                       BIS     #20,R5          ;ADD THESE BITS
2795  017052  116104  000005                       MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2796  017056  052704  000020                       BIS     #20,R4          ;ADD THIS BIT
2797  017062  120504                                CMPB    R5,R4           ;DATA CORRECT?
2798  017064  001401                                BEQ     68$             ;BR IF YES
2799  017066  104004                                HLT     4               ;ERROR
2800  017070  104401                          68$:  SCOP1                   ;SW09=1?
2801  017072  005100                                COM     R0              ;CHANGE TO FLOATING 1
2802  017074  000241                                CLC                     ;CLEAR CARRY
2803  017076  106100                                ROLB    R0              ;SHIFT BIT IN R0
2804  017100  001345                                BNE     69$             ;IF R0=0 THEN DONE
2805  017102  104400                                SCOPE                   ;SCOPE THIS TEST
2806
2807
2808                                          ;************************** TEST 40 **************************
2809                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
2810                                          ;*FLOAT A 1 THROUGH IBUS REGISTER 0
2811                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 0
2812                                          ;;**********************************************************
2813
2814                                          ;  TEST 40
2815                                          ;--------------
2816  017104  012737  000040  001226  TST40:  MOV     #40,TSTNO
2817  017112  012737  017260  001216          MOV     #TST41,NEXT
2818  017120  012737  017140  001220          MOV     #64$,LOCK
2819                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2820  017126  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2821  017130  012702  000000                  MOV     #0,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2822  017134  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2823  017140                            64$:
2824  017140  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2825  017144  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2826  017146  122100                          122100!0                ;MOV DATA TO IBUS REGISTER 0
2827  017150  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2828  017152  021005                          21005!(0*20)            ;READ FROM IBUS REGISTER 0
2829  017154  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2830  017156  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2831  017162  120504                          CMPB    R5,R4           ;DATA CORRECT?
2832  017164  001401                          BEQ     65$             ;BR IF YES
2833  017166  104005                          HLT     5               ;ERROR
2834  017170  104401                    65$:  SCOP1                   ;SW09=1?
2835  017172  000241                          CLC                     ;CLEAR CARRY
2836  017174  106100                          ROLB    R0              ;SHIFT BIT IN R0
2837  017176  001360                          BNE     64$             ;IF R0=0 THEN DONE
2838  017200  012737  017214  001220          MOV     #67$,LOCK       ;NEW SCOP1
2839  017206  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2840  017212  005100                    69$:  COM     R0              ;CHANGE TO FLOATING ZERO
```

```
2841  017214                                  67$:
2842  017214  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2843  017220  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2844  017222  122100                                122100!0                ;MOV DATA TO IBUS REGISTER 0
2845  017224  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2846  017226  021005                                21005!<0*20>            ;READ FROM IBUS REGISTER 0
2847  017230  010005                                MOV     R0,R5           ;PUT EXPECTED IN R5
2848  017232  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2849  017236  120504                                CMPB    R5,R4           ;DATA CORRECT?
2850  017240  001401                                BEQ     68$             ;BR IF YES
2851  017242  104005                                HLT     5               ;ERROR
2852  017244  104401                          68$:  SCOP1                   ;SW09=1?
2853  017246  005100                                COM     R0              ;CHANGE TO FLOATING 1
2854  017250  000241                                CLC                     ;CLEAR CARRY
2855  017252  106100                                ROLB    R0              ;SHIFT BIT IN R0
2856  017254  001356                                BNE     69$             ;IF R0=0 THEN DONE
2857  017256  104400                                SCOPE                   ;SCOPE THIS TEST
2858
2859
2860                                                ;************************** TEST 41 **************************
2861                                                ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
2862                                                ;*FLOAT A 1 THROUGH IBUS REGISTER 1
2863                                                ;*FLOAT A 0 THROUGH IBUS REGISTER 1
2864                                                ;:**********************************************************
2865
2866                                                ;  TEST 41
2867                                                ;--------------
2868  017260  012737  000041  001226          TST41:  MOV     #41,TSTNO
2869  017266  012737  017434  001216                  MOV     #TST42,NEXT
2870  017274  012737  017314  001220                  MOV     #64$,LOCK
2871                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
2872  017302  104412                                MSTCLR                  ;MASTER CLEAR DMC11
2873  017304  012702  000001                        MOV     #1,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2874  017310  012700  000001                        MOV     #1,R0           ;START WITH BIT 0
2875  017314                                  64$:
2876  017314  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2877  017320  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2878  017322  122101                                122100!1                ;MOV DATA TO IBUS REGISTER 1
2879  017324  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2880  017326  021025                                21005!<1*20>            ;READ FROM IBUS REGISTER 1
2881  017330  010005                                MOV     R0,R5           ;PUT EXPECTED IN R5
2882  017332  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2883  017336  120504                                CMPB    R5,R4           ;DATA CORRECT?
2884  017340  001401                                BEQ     65$             ;BR IF YES
2885  017342  104005                                HLT     5               ;ERROR
2886  017344  104401                          65$:  SCOP1                   ;SW09=1?
2887  017346  000241                                CLC                     ;CLEAR CARRY
2888  017350  106100                                ROLB    R0              ;SHIFT BIT IN R0
2889  017352  001360                                BNE     64$             ;IF R0=0 THEN DONE
2890  017354  012737  017370  001220                MOV     #67$,LOCK       ;NEW SCOP1
2891  017362  012700  000001                        MOV     #1,R0           ;START WITH BIT 0
2892  017366  005100                          69$:  COM     R0              ;CHANGE TO FLOATING ZERO
2893  017370                                  67$:
2894  017370  010061  000004                        MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2895  017374  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2896  017376  122101                                122100!1                ;MOV DATA TO IBUS REGISTER 1
```

```
2897  017400  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2898  017402  021025                          21005!(1*20)            ;READ FROM IBUS REGISTER 1
2899  017404  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2900  017406  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2901  017412  120504                          CMPB    R5,R4           ;DATA CORRECT?
2902  017414  001401                          BEQ     68$             ;BR IF YES
2903  017416  104005                          HLT     5               ;ERROR
2904  017420  104401                  68$:    SCOP1                   ;SW09=1?
2905  017422  005100                          COM     R0              ;CHANGE TO FLOATING 1
2906  017424  000241                          CLC                     ;CLEAR CARRY
2907  017426  106100                          ROLB    R0              ;SHIFT BIT IN R0
2908  017430  001356                          BNE     69$             ;IF R0=0 THEN DONE
2909  017432  104400                          SCOPE                   ;SCOPE THIS TEST
2910
2911
2912                                          ;*************************** TEST 42 ****************************
2913                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
2914                                          ;*FLOAT A 1 THROUGH IBUS REGISTER 2
2915                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 2
2916                                          ;:************************************************************
2917
2918                                          ;  TEST 42
2919                                          ;-------------
2920  017434  012737  000042  001226  TST42:  MOV     #42,TSTNO
2921  017442  012737  017610  001216          MOV     #TST43,NEXT
2922  017450  012737  017470  001220          MOV     #64$,LOCK
2923                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2924  017456  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2925  017460  012702  000002                  MOV     #2,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2926  017464  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2927  017470                          64$:
2928  017470  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2929  017474  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2930  017476  122102                          122100!2                ;MOV DATA TO IBUS REGISTER 2
2931  017500  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2932  017502  021045                          21005!(2*20)            ;READ FROM IBUS REGISTER 2
2933  017504  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2934  017506  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2935  017512  120504                          CMPB    R5,R4           ;DATA CORRECT?
2936  017514  001401                          BEQ     65$             ;BR IF YES
2937  017516  104005                          HLT     5               ;ERROR
2938  017520  104401                  65$:    SCOP1                   ;SW09=1?
2939  017522  000241                          CLC                     ;CLEAR CARRY
2940  017524  106100                          ROLB    R0              ;SHIFT BIT IN R0
2941  017526  001360                          BNE     64$             ;IF R0=0 THEN DONE
2942  017530  012737  017544  001220          MOV     #67$,LOCK       ;NEW SCOP1
2943  017536  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2944  017542  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2945  017544                          67$:
2946  017544  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2947  017550  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2948  017552  122102                          122100!2                ;MOV DATA TO IBUS REGISTER 2
2949  017554  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2950  017556  021045                          21005!(2*20)            ;READ FROM IBUS REGISTER 2
2951  017560  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2952  017562  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
```

```
2953  017566  120504                          CMPB    R5,R4           ;DATA CORRECT?
2954  017570  001401                          BEQ     68$             ;BR IF YES
2955  017572  104005                          HLT     5               ;ERROR
2956  017574  104401                  68$:    SCOP1                   ;SW09=1?
2957  017576  005100                          COM     R0              ;CHANGE TO FLOATING 1
2958  017600  000241                          CLC                     ;CLEAR CARRY
2959  017602  106100                          ROLB    R0              ;SHIFT BIT IN R0
2960  017604  001356                          BNE     69$             ;IF R0=0 THEN DONE
2961  017606  104400                          SCOPE                   ;SCOPE THIS TEST
2962
2963
2964                                          ;*************************** TEST 43 ****************************
2965                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
2966                                          ;*FLOAT A 1 THROUGH IBUS REGISTER 3
2967                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 3
2968                                          ;:************************************************************
2969
2970                                          ;  TEST 43
2971                                          ;-------------
2972  017610  012737  000043  001226  TST43:  MOV     #43,TSTNO
2973  017616  012737  017764  001216          MOV     #TST44,NEXT
2974  017624  012737  017644  001220          MOV     #64$,LOCK
2975                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2976  017632  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2977  017634  012702  000003                  MOV     #3,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
2978  017640  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2979  017644                          64$:
2980  017644  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2981  017650  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2982  017652  122103                          122100!3                ;MOV DATA TO IBUS REGISTER 3
2983  017654  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2984  017656  021065                          21005!(3*20)            ;READ FROM IBUS REGISTER 3
2985  017660  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
2986  017662  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
2987  017666  120504                          CMPB    R5,R4           ;DATA CORRECT?
2988  017670  001401                          BEQ     65$             ;BR IF YES
2989  017672  104005                          HLT     5               ;ERROR
2990  017674  104401                  65$:    SCOP1                   ;SW09=1?
2991  017676  000241                          CLC                     ;CLEAR CARRY
2992  017700  106100                          ROLB    R0              ;SHIFT BIT IN R0
2993  017702  001360                          BNE     64$             ;IF R0=0 THEN DONE
2994  017704  012737  017720  001220          MOV     #67$,LOCK       ;NEW SCOP1
2995  017712  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
2996  017716  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
2997  017720                          67$:
2998  017720  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
2999  017724  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3000  017726  122103                          122100!3                ;MOV DATA TO IBUS REGISTER 3
3001  017730  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3002  017732  021065                          21005!(3*20)            ;READ FROM IBUS REGISTER 3
3003  017734  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
3004  017736  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3005  017742  120504                          CMPB    R5,R4           ;DATA CORRECT?
3006  017744  001401                          BEQ     68$             ;BR IF YES
3007  017746  104005                          HLT     5               ;ERROR
3008  017750  104401                  68$:    SCOP1                   ;SW09=1?
```

```
3009  017752  005100                          COM     R0              ;CHANGE TO FLOATING 1
3010  017754  000241                          CLC                     ;CLEAR CARRY
3011  017756  106100                          ROLB    R0              ;SHIFT BIT IN R0
3012  017760  001356                          BNE     69$             ;IF R0=0 THEN DONE
3013  017762  104400                          SCOPE                   ;SCOPE THIS TEST
3014
3015
3016                                          ;*************************** TEST 44 ****************************
3017                                          ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3018                                          ;*FLOAT A 1 THROUGH IBUS REGISTER 4
3019                                          ;*FLOAT A 0 THROUGH IBUS REGISTER 4
3020                                          ;:************************************************************
3021
3022                                          ;  TEST 44
3023                                          ;--------------
3024  017764  012737  000044  001226  TST44:  MOV     #44,TSTNO
3025  017772  012737  020140  001216          MOV     #TST45,NEXT
3026  020000  012737  020020  001220          MOV     #64$,LOCK
3027                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3028  020006  104412                          MSTCLR                  ;MASTER CLEAR DMC11
3029  020010  012702  000004                  MOV     #4,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
3030  020014  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
3031  020020                          64$:
3032  020020  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3033  020024  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3034  020026  122104                          122100!4                ;MOV DATA TO IBUS REGISTER 4
3035  020030  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3036  020032  021105                          21005!(4*20)            ;READ FROM IBUS REGISTER 4
3037  020034  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
3038  020036  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3039  020042  120504                          CMPB    R5,R4           ;DATA CORRECT?
3040  020044  001401                          BEQ     65$             ;BR IF YES
3041  020046  104005                          HLT     5               ;ERROR
3042  020050  104401                  65$:    SCOP1                   ;SW09=1?
3043  020052  000241                          CLC                     ;CLEAR CARRY
3044  020054  106100                          ROLB    R0              ;SHIFT BIT IN R0
3045  020056  001360                          BNE     64$             ;IF R0=0 THEN DONE
3046  020060  012737  020074  001220          MOV     #67$,LOCK       ;NEW SCOP1
3047  020066  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
3048  020072  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
3049  020074                          67$:
3050  020074  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3051  020100  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3052  020102  122104                          122100!4                ;MOV DATA TO IBUS REGISTER 4
3053  020104  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3054  020106  021105                          21005!(4*20)            ;READ FROM IBUS REGISTER 4
3055  020110  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
3056  020112  116104  000005                  MOVB    5(R1),R4        ;PLT "FOUND" INTO R4
3057  020116  120504                          CMPB    R5,R4           ;DATA CORRECT?
3058  020120  001401                          BEQ     68$             ;BR IF YES
3059  020122  104005                          HLT     5               ;ERROR
3060  020124  104401                  68$:    SCOP1                   ;SW09=1?
3061  020126  005100                          COM     R0              ;CHANGE TO FLOATING 1
3062  020130  000241                          CLC                     ;CLEAR CARRY
3063  020132  106100                          ROLB    R0              ;SHIFT BIT IN R0
3064  020134  001356                          BNE     69$             ;IF R0=0 THEN DONE
```

```
3065  020136  104400                                  SCOPE                   ;SCOPE THIS TEST
3066
3067
3068                                                  ;************************** TEST 45 **************************
3069                                                  ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3070                                                  ;*FLOAT A 1 THROUGH IBUS REGISTER 5
3071                                                  ;*FLOAT A 0 THROUGH IBUS REGISTER 5
3072                                                  ;;*********************************************************************
3073
3074                                                  ;  TEST 45
3075                                                  ;--------------
3076  020140  012737  000045  001226          TST45:  MOV     #45,TSTNO
3077  020146  012737  020314  001216                  MOV     #TST46,NEXT
3078  020154  012737  020174  001220                  MOV     #64$,LOCK
3079                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
3080  020162  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
3081  020164  012702  000005                          MOV     #5,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
3082  020170  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3083  020174                                  64$:
3084  020174  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3085  020200  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3086  020202  122105                                  122100!5                ;MOV DATA TO IBUS REGISTER 5
3087  020204  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
3088  020206  021125                                  21005!(5*20)            ;READ FROM IBUS REGISTER 5
3089  020210  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
3090  020212  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3091  020216  120504                                  CMPB    R5,R4           ;DATA CORRECT?
3092  020220  001401                                  BEQ     65$             ;BR IF YES
3093  020222  104005                                  HLT     5               ;ERROR
3094  020224  104401                          65$:    SCOP1                   ;SW09=1?
3095  020226  000241                                  CLC                     ;CLEAR CARRY
3096  020230  106100                                  ROLB    R0              ;SHIFT BIT IN R0
3097  020232  001360                                  BNE     64$             ;IF R0=0 THEN DONE
3098  020234  012737  020250  001220                  MOV     #67$,LOCK       ;NEW SCOP1
3099  020242  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3100  020246  005100                          69$:    COM     R0              ;CHANGE TO FLOATING ZERO
3101  020250                                  67$:
3102  020250  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3103  020254  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3104  020256  122105                                  122100!5                ;MOV DATA TO IBUS REGISTER 5
3105  020260  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3106  020262  021125                                  21005!(5*20)            ;READ FROM IBUS REGISTER 5
3107  020264  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
3108  020266  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3109  020272  120504                                  CMPB    R5,R4           ;DATA CORRECT?
3110  020274  001401                                  BEQ     68$             ;BR IF YES
3111  020276  104005                                  HLT     5               ;ERROR
3112  020300  104401                          68$:    SCOP1                   ;SW09=1?
3113  020302  005100                                  COM     R0              ;CHANGE TO FLOATING 1
3114  020304  000241                                  CLC                     ;CLEAR CARRY
3115  020306  106100                                  ROLB    R0              ;SHIFT BIT IN R0
3116  020310  001356                                  BNE     69$             ;IF R0=0 THEN DONE
3117  020312  104400                                  SCOPE                   ;SCOPE THIS TEST
3118
3119
3120                                                  ;************************** TEST 46 **************************
```

```
3121                                                    ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3122                                                    ;*FLOAT A 1 THROUGH IBUS REGISTER 6
3123                                                    ;*FLOAT A 0 THROUGH IBUS REGISTER 6
3124                                                    ;;**************************************************************
3125
3126                                                    ;  TEST 46
3127                                                    ;--------------
3128  020314  012737  000046  001226          TST46:  MOV     #46,TSTNO
3129  020322  012737  020470  001216                  MOV     #TST47,NEXT
3130  020330  012737  020350  001220                  MOV     #64$,LOCK
3131                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
3132  020336  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
3133  020340  012702  000006                          MOV     #6,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
3134  020344  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3135  020350                                  64$:
3136  020350  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3137  020354  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3138  020356  122106                                   122100!6                ;MOV DATA TO IBUS REGISTER 6
3139  020360  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3140  020362  021145                                   21005!(6*20)            ;READ FROM IBUS REGISTER 6
3141  020364  010005                                   MOV     R0,R5           ;PUT EXPECTED IN R5
3142  020366  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3143  020372  120504                                   CMPB    R5,R4           ;DATA CORRECT?
3144  020374  001401                                   BEQ     65$             ;BR IF YES
3145  020376  104005                                   HLT     5               ;ERROR
3146  020400  104401                          65$:     SCOP1                   ;SW09=1?
3147  020402  000241                                   CLC                     ;CLEAR CARRY
3148  020404  106100                                   ROLB    R0              ;SHIFT BIT IN R0
3149  020406  001360                                   BNE     64$             ;IF R0=0 THEN DONE
3150  020410  012737  020424  001220                  MOV     #67$,LOCK       ;NEW SCOP1
3151  020416  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3152  020422  005100                          69$:     COM     R0              ;CHANGE TO FLOATING ZERO
3153  020424                                  67$:
3154  020424  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3155  020430  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3156  020432  122106                                   122100!6                ;MOV DATA TO IBUS REGISTER 6
3157  020434  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3158  020436  021145                                   21005!(6*20)            ;READ FROM IBUS REGISTER 6
3159  020440  010005                                   MOV     R0,R5           ;PUT EXPECTED IN R5
3160  020442  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3161  020446  120504                                   CMPB    R5,R4           ;DATA CORRECT?
3162  020450  001401                                   BEQ     68$             ;BR IF YES
3163  020452  104005                                   HLT     5               ;ERROR
3164  020454  104401                          68$:     SCOP1                   ;SW09=1?
3165  020456  005100                                   COM     R0              ;CHANGE TO FLOATING 1
3166  020460  000241                                   CLC                     ;CLEAR CARRY
3167  020462  106100                                   ROLB    R0              ;SHIFT BIT IN R0
3168  020464  001356                                   BNE     69$             ;IF R0=0 THEN DONE
3169  020466  104400                                   SCOPE                   ;SCOPE THIS TEST
3170
3171
3172                                                    ;************************** TEST 47 **************************
3173                                                    ;*MICRO PROCESSOR IBUS REGISTER WRITE/READ TEST
3174                                                    ;*FLOAT A 1 THROUGH IBUS REGISTER 7
3175                                                    ;*FLOAT A 0 THROUGH IBUS REGISTER 7
3176                                                    ;;**************************************************************
```

```
3177
3178
3179
3180  020470  012737  000047  001226          TST47:  MOV     #47,TSTNO
3181  020476  012737  020644  001216                  MOV     #TST50,NEXT
3182  020504  012737  020524  001220                  MOV     #64$,LOCK
3183
3184  020512  104412                                  MSTCLR
3185  020514  012702  000007                          MOV     #7,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
3186  020520  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3187  020524                                  64$:
3188  020524  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3189  020530  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3190  020532  122107                                  122100!7                ;MOV DATA TO IBUS REGISTER 7
3191  020534  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3192  020536  021165                                  21005!(7*20)            ;READ FROM IBUS REGISTER 7
3193  020540  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
3194  020542  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3195  020546  120504                                  CMPB    R5,R4           ;DATA CORRECT?
3196  020550  001401                                  BEQ     65$             ;BR IF YES
3197  020552  104005                                  HLT     5               ;ERROR
3198  020554  104401                          65$:    SCOP1                   ;SW09=1?
3199  020556  000241                                  CLC                     ;CLEAR CARRY
3200  020560  106100                                  ROLB    R0              ;SHIFT BIT IN R0
3201  020562  001360                                  BNE     64$             ;IF R0=0 THEN DONE
3202  020564  012737  020600  001220                  MOV     #67$,LOCK       ;NEW SCOP1
3203  020572  012700  000001                          MOV     #1,R0           ;START WITH BIT 0
3204  020576  005100                          69$:    COM     R0              ;CHANGE TO FLOATING ZERO
3205  020600                                  67$:
3206  020600  010061  000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
3207  020604  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3208  020606  122107                                  122100!7                ;MOV DATA TO IBUS REGISTER 7
3209  020610  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3210  020612  021165                                  21005!(7*20)            ;READ FROM IBUS REGISTER 7
3211  020614  010005                                  MOV     R0,R5           ;PUT EXPECTED IN R5
3212  020616  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
3213  020622  120504                                  CMPB    R5,R4           ;DATA CORRECT?
3214  020624  001401                                  BEQ     68$             ;BR IF YES
3215  020626  104005                                  HLT     5               ;ERROR
3216  020630  104401                          68$:    SCOP1                   ;SW09=1?
3217  020632  005100                                  COM     R0              ;CHANGE TO FLOATING 1
3218  020634  000241                                  CLC                     ;CLEAR CARRY
3219  020636  106100                                  ROLB    R0              ;SHIFT BIT IN R0
3220  020640  001356                                  BNE     69$             ;IF R0=0 THEN DONE
3221  020642  104400                                  SCOPE                   ;SCOPE THIS TEST
3222
3223
3224                                                  ;************************** TEST 50 **************************
3225                                                  ;*MICRO PROCESSOR IBUS DUAL ADDRESS TEST
3226                                                  ;*WRITE ALL IBUS REGISTERS WITH INCREMENTING PATTERN
3227                                                  ;*READ ALL IBUS REGISTERS TO VERIFY CORRECT ADDRESSING
3228                                                  ;*************************************************************
3229
3230                                                  ;  TEST 50
3231                                                  ;--------------
3232  020644  012737  000050  001226          TST50:  MOV     #50,TSTNO
```

```
3233  020652  012737  021072  001216                MOV     #TST51,NEXT
3234  020660  012737  020676  001220                MOV     #1$,LOCK
3235                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
3236  020666  104412                                MSTCLR                  ;MASTER CLEAR DMC11
3237  020670  012700  000001                        MOV     #1,R0           ;START WITH A ONE
3238  020674  005002                                CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
3239  020676  010203                        1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
3240  020700  010061  000004                        MOV     R0,4(R1)        ;WRITE DATA TO PORT4
3241  020704  042737  000017  020720                BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
3242  020712  050337  020720                        BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
3243  020716  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3244  020720  122100                        5$:     122100                  ;MOVE DATA TO IBUS REGISTER
3245  020722  006303                                ASL     R3              ;SHIFT ADDRESS
3246  020724  006303                                ASL     R3              ;4 TIMES TO GET
3247  020726  006303                                ASL     R3              ;IT TO BITS 4-7
3248  020730  006303                                ASL     R3              ;OF NEXT INSTRUCTION
3249  020732  042737  000360  020746                BIC     #360,6$         ;CLEAR ADDRESS FIELD
3250  020740  050337  020746                        BIS     R3,6$           ;ADD ADDRESS TO INSTRUCTION
3251  020744  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3252  020746  021005                        6$:     21005                   ;READ FROM IBUS REGISTER
3253  020750  010005                                MOV     R0,R5           ;PUT "EXPECTED" IN R5
3254  020752  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3255  020756  120504                                CMPB    R5,R4           ;IS DATA CORRECT?
3256  020760  001401                                BEQ     2$              ;BR IF YES
3257  020762  104005                                HLT     5               ;DATA ERROR
3258  020764  104401                        2$:     SCOP1                   ;SW09=1?
3259  020766  005200                                INC     R0              ;INCREMENT PATTERN
3260  020770  005202                                INC     R2              ;INCREMENT REGISTER ADDRESS
3261  020772  022702  000010                        CMP     #7+1,R2 ;LAST ADDRESS DONE?
3262  020776  001337                                BNE     1$              ;BR IF NO
3263  021000  012737  021016  001220                MOV     #3$,LOCK        ;NEW SCOP1
3264  021006  012700  000001                        MOV     #1,R0           ;RESTART PATTERN TO 1
3265  021012  005002                                CLR     R2              ;RESTART AT ADDRESS 0
3266  021014  005003                                CLR     R3              ;RESTART AT ADDRESS 0
3267  021016  042737  000360  021032        3$:     BIC     #360,7$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3268  021024  050337  021032                        BIS     R3,7$           ;ADD ADDRESS TO INSTRUCTION
3269  021030  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3270  021032  021005                        7$:     21005                   ;READ FROM IBUS REGISTER
3271  021034  010005                                MOV     R0,R5           ;PUT "EXPECTED" IN R5
3272  021036  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R5
3273  021042  120504                                CMPB    R5,R4           ;DATA CORRECT?
3274  021044  001401                                BEQ     4$              ;BR IF YES
3275  021046  104005                                HLT     5               ;DUAL ADDRESSING ERROR
3276  021050  104401                        4$:     SCOP1                   ;SW09=1?
3277  021052  005200                                INC     R0              ;INCREMENT PATTERN
3278  021054  005202                                INC     R2              ;NEXT ADDRESS
3279  021056  062703  000020                        ADD     #20,R3          ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
3280  021062  022702  000010                        CMP     #7+1,R2 ;LAST ADDRESS DONE?
3281  021066  001353                                BNE     3$              ;BR IF NO
3282  021070  104400                                SCOPE                   ;SCOPE THIS TEST
3283
3284
3285                                                ;*************************** TEST 51 ****************************
3286                                                ;*MICRO PROCESSOR BR REGISTER TEST
3287                                                ;*FLOAT A 1 THROUGH THE BR
3288                                                ;*FLOAT A 0 THROUGH THE BR
```

```
3289                                            ;:*************************************************************
3290
3291                                            ;  TEST 51
3292                                            ;-------------
3293  021072  012737  000051  001226  TST51:    MOV     #51,TSTNO
3294  021100  012737  021242  001216            MOV     #TST52,NEXT
3295  021106  012737  021122  001220            MOV     #64$,LOCK
3296                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
3297  021114  104412                           MSTCLR                  ;MASTER CLEAR DMC11
3298  021116  012700  000001                   MOV     #1,R0           ;START PATTERN WITH BIT0
3299  021122                          64$:
3300  021122  010061  000004                   MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
3301  021126  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3302  021130  120500                           120500                  ;MOVE DATA TO THE BR REGISTER
3303  021132  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3304  021134  061225                           061225                  ;MOVE BR TO PORT 5
3305  021136  010005                           MOV     R0,R5           ;PUT "EXPECTED" IN R5
3306  021140  116104  000005                   MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3307  021144  120504                           CMPB    R5,R4           ;DATA CORRECT?
3308  021146  001401                           BEQ     65$             ;BR IF YES
3309  021150  104006                           HLT     6               ;DATA ERROR
3310  021152  104401                  65$:     SCOP1
3311  021154  000241                           CLC                     ;CLEAR CARRY
3312  021156  106100                           ROLB    R0              ;SHIFT BIT IN R0
3313  021160  001360                           BNE     64$             ;DONE IF R0=0
3314  021162  012737  021176  001220           MOV     #67$,LOCK       ;NEW SCOP1
3315  021170  012700  000001                   MOV     #1,R0           ;START PATTERN WITH BIT0
3316  021174  005100                  69$:     COM     R0              ;CHANGE TO FLOATING ZERO
3317  021176                          67$:
3318  021176  010061  000004                   MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
3319  021202  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3320  021204  120500                           120500                  ;MOVE DATA TO THE BR REGISTER
3321  021206  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3322  021210  061225                           061225                  ;MOVE BR TO PORT 5
3323  021212  010005                           MOV     R0,R5           ;PUT "EXPECTED" IN R5
3324  021214  116104  000005                   MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3325  021220  120504                           CMPB    R5,R4           ;DATA CORRECT?
3326  021222  001401                           BEQ     68$             ;BR IF YES
3327  021224  104006                           HLT     6               ;DATA ERROR
3328  021226  104401                  68$:     SCOP1
3329  021230  005100                           COM     R0              ;CHANGE BACK TO ONE
3330  021232  000241                           CLC                     ;CLEAR CARRY
3331  021234  106100                           ROLB    R0              ;SHIFT BIT IN R0
3332  021236  001356                           BNE     69$             ;DONE IF R0=0
3333  021240  104400                           SCOPE                   ;SCOPE THIS TEST
3334
3335
3336                                            ;*************************** TEST 52 ****************************
3337                                            ;*SCRATCH PAD TEST
3338                                            ;*FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
3339                                            ;*FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION
3340                                            ;:*************************************************************
3341
3342                                            ;  TEST 52
3343                                            ;-------------
3344  021242  012737  000052  001226  TST52:    MOV     #52,TSTNO
```

```
3345  021250  012737  021510  001216                MOV     #TST53,NEXT
3346  021256  012737  021274  001220                MOV     #64$,LOCK
3347                                                ;R1 CONTAINS BASE DMC11 ADDRESS
3348  021264  104412                                MSTCLR                  ;MASTER CLEAR DMC11
3349  021266  005002                                CLR     R2              ;START AT ADDRESS ZERO
3350  021270  012700  000001                        MOV     #1,R0           ;START WITH BIT0
3351  021274  042737  000017  021314        64$:    BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3352  021302  050237  021314                        BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
3353  021306  010061  000004                        MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
3354  021312  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3355  021314  123100                        65$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
3356  021316  042737  000017  021332                BIC     #17,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3357  021324  050237  021332                        BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
3358  021330  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3359  021332  040600                        66$:    040600                  ;MOV SP TO BR
3360  021334  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3361  021336  061225                                061225                  ;MOVE BR TO PORT5
3362  021340  010005                                MOV     R0,R5           ;PUT "EXPECTED" IN R5
3363  021342  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3364  021346  120504                                CMPB    R5,R4           ;DATA CORRECT
3365  021350  001401                                BEQ     67$             ;BR IF YES
3366  021352  104007                                HLT     7               ;DATA ERROR
3367  021354  104401                        67$:    SCOP1                   ;SW09=1?
3368  021356  000241                                CLC                     ;CLEAR CARRY
3369  021360  106100                                ROLB    R0              ;SHIFT BIT IN R0
3370  021362  001344                                BNE     64$             ;DONE IF R0=0
3371  021364  012737  021400  001220                MOV     #69$,LOCK       ;NEW SCOP1
3372  021372  012700  000001                        MOV     #1,R0           ;START WITH BIT0
3373  021376  005100                        73$:    COM     R0              ;CHANGE TO FLOATING ZERO
3374  021400  042737  000017  021420        69$:    BIC     #17,70$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3375  021406  050237  021420                        BIS     R2,70$          ;ADD ADDRESS TO INSTRUCTION
3376  021412  010061  000004                        MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
3377  021416  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3378  021420  123100                        70$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
3379  021422  042737  000017  021436                BIC     #17,71$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3380  021430  050237  021436                        BIS     R2,71$          ;ADD ADDRESS TO INSTRUCTION
3381  021434  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3382  021436  040600                        71$:    040600                  ;MOV SP TO BR
3383  021440  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3384  021442  061225                                061225                  ;MOVE BR TO PORT5
3385  021444  010005                                MOV     R0,R5           ;PUT "EXPECTED" IN R5
3386  021446  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3387  021452  120504                                CMPB    R5,R4           ;DATA CORRECT
3388  021454  001401                                BEQ     72$             ;BR IF YES
3389  021456  104007                                HLT     7               ;DATA ERROR
3390  021460  104401                        72$:    SCOP1                   ;SW09=1?
3391  021462  005100                                COM     R0              ;CHANGE BACK TO A ONE
3392  021464  000241                                CLC                     ;CLEAR CARRY
3393  021466  106100                                ROLB    R0              ;SHIFT BIT IN R0
3394  021470  001342                                BNE     73$             ;DONE IF R0=0
3395  021472  012700  000001                        MOV     #1,R0           ;RESTART AT BIT 0
3396  021476  005202                                INC     R2              ;NEXT SP ADDRESS
3397  021500  022702  000020                        CMP     #20,R2  ;LAST ADDRESS?
3398  021504  001273                                BNE     64$             ;BR IF NO
3399  021506  104400                                SCOPE                   ;SCOPE THIS TEST
3400
```

```
3401
3402                                    ;*************************** TEST 53 ****************************
3403                                    ;*SCRATCH PAD DUAL ADDRESSING TEST
3404                                    ;*WRITE AN INCREMENTING PATTERN IN ALL SP LOCATIONS
3405                                    ;*READ ALL SP LOCATIONS TO VERIFY CORRECT ADDRESSING
3406                                    ;:************************************************************
3407
3408                                    ;  TEST 53
3409                                    ;--------------
3410  021510  012737  000053  001226  TST53:  MOV     #53,TSTNO
3411  021516  012737  021732  001216          MOV     #TST54,NEXT
3412  021524  012737  021542  001220          MOV     #1$,LOCK
3413                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3414  021532  104412                         MSTCLR                  ;MASTER CLEAR DMC11
3415  021534  012700  000001                 MOV     #1,R0           ;START WITH A 1
3416  021540  005003                         CLR     R3              ;ADDRESS 0
3417  021542  010302                  1$:    MOV     R3,R2           ;MOVE ADDRESS TO R2
3418  021544  042737  000017  021564         BIC     #17,2$          ;CLEAR ADDRESS FIELD
3419  021552  050237  021564                 BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
3420  021556  010061  000004                 MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
3421  021562  104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3422  021564  123100                  2$:    123100                  ;WRITE SP(ADDRESS IN R2)
3423  021566  042737  000017  021602         BIC     #17,3$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
3424  021574  050237  021602                 BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
3425  021600  104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3426  021602  060600                  3$:    60600                   ;MOV SP TO BR
3427  021604  104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3428  021606  061225                         61225                   ;MOV BR TO PORT5
3429  021610  010005                         MOV     R0,R5           ;PUT "EXPECTED" IN R5
3430  021612  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3431  021616  120504                         CMPB    R5,R4           ;DATA CORRECT?
3432  021620  001401                         BEQ     4$              ;BR IF YES
3433  021622  104007                         HLT     7               ;DATA ERROR
3434  021624  104401                  4$:    SCOP1                   ;SW09=0
3435  021626  005200                         INC     R0              ;INCREMENT PATTERN
3436  021630  005203                         INC     R3              ;NEXT ADDRESS
3437  021632  022703  000020                 CMP     #20,R3          ;LAST ADDRESS DONE?
3438  021636  001341                         BNE     1$              ;BR IF NO
3439  021640  012737  021654  001220         MOV     #5$,LOCK        ;NEW SCOP1
3440  021646  012700  000001                 MOV     #1,R0           ;RESTART PATTERN AT 1
3441  021652  005003                         CLR     R3              ;RESTART AT ADDRESS ZERO
3442  021654  010302                  5$:    MOV     R3,R2           ;PUT ADDRESS IN R2
3443  021656  042737  000017  021672         BIC     #17,6$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
3444  021664  050237  021672                 BIS     R2,6$           ;ADD ADDRESS TO INSTRUCTION
3445  021670  104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3446  021672  060600                  6$:    60600                   ;MOV SP TO BR
3447  021674  104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3448  021676  061225                         61225                   ;MOV BR TO PORT5
3449  021700  010005                         MOV     R0,R5           ;PUT "EXPECTED" IN R5
3450  021702  116104  000005                 MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3451  021706  120504                         CMPB    R5,R4           ;DATA CORRECT?
3452  021710  001401                         BEQ     7$              ;BR IF YES
3453  021712  104007                         HLT     7               ;SP ADDRESSING ERROR
3454  021714  104401                  7$:    SCOP1                   ;SW09=1?
3455  021716  005200                         INC     R0              ;INCREMENT PATTERN
3456  021720  005203                         INC     R3              ;NEXT ADDRESS
```

```
3457  021722  022703  000020                          CMP     #20,R3  ;LAST ADDRESS DONE?
3458  021726  001352                                  BNE     5$              ;BR IF NO
3459  021730  104400                                  SCOPE                   ;SCOPE THIS TEST
3460
3461                                                  ;************************** TEST 54 **************************
3462                                                  ;*INTERRUPT TEST
3463                                                  ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR A
3464                                                  ;:**********************************************************
3465
3466
3467                                                  ;  TEST 54
3468                                                  ;--------------
3469  021732  012737  000054  001226          TST54:  MOV     #54,TSTNO
3470  021740  012737  022026  001216                  MOV     #TST55,NEXT
3471                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3472  021746  000005                                  RESET                   ;BUS RESET
3473  021750  005011                                  CLR     (R1)            ;CLEAR RUN
3474  021752  004537  034600                          JSR     R5,SETVEC       ;SET UP VECTORS
3475  021756  022020                                  3$                      ;XX0
3476  021760  022016                                  2$                      ;XX4
3477  021762     340     340                          .BYTE   340,340         ;LEVEL 7
3478  021764  012737  000340  177776          1$:     MOV     #340,PS         ;PS = LEVEL 7
3479  021772  012761  000200  000004                  MOV     #200,4(R1)      ;WRITE PORT4
3480  022000  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3481  022002  121111                                  121111                  ;SET BR RQ IN IBUS* REG 11
3482  022004  005037  177776                          CLR     PS              ;ALLOW INTERRUPT
3483  022010  000240                                  NOP
3484  022012  104010                                  HLT     10              ;NO INTERRUPT
3485  022014  000403                                  BR      4$
3486  022016  104011                          2$:     HLT     11              ;WRONG VECTOR
3487  022020  012706  001200                  3$:     MOV     #STACK,SP       ;RESET STACK
3488  022024  104400                          4$:     SCOPE                   ;SCOPE THIS TEST
3489
3490
3491                                                  ;************************** TEST 55 **************************
3492                                                  ;*INTERRUPT TEST
3493                                                  ;*TEST THAT DEVICE CAN INTERRUPT TO VECTOR B
3494                                                  ;:**********************************************************
3495
3496                                                  ;  TEST 55
3497                                                  ;--------------
3498  022026  012737  000055  001226          TST55:  MOV     #55,TSTNO
3499  022034  012737  022120  001216                  MOV     #TST56,NEXT
3500                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3501  022042  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
3502  022044  004537  034600                          JSR     R5,SETVEC       ;SET UP VECTORS
3503  022050  022110                                  2$                      ;XX0
3504  022052  022112                                  3$                      ;XX4
3505  022054     340     340                          .BYTE   340,340         ;LEVEL 7
3506  022056  012737  000340  177776          1$:     MOV     #340,PS         ;PS = LEVEL 7
3507  022064  012761  000300  000004                  MOV     #300,4(R1)      ;WRITE PORT4
3508  022072  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3509  022074  121111                                  121111                  ;SET BR RQ IN IBUS* REG 11
3510  022076  005037  177776                          CLR     PS              ;ALLOW INTERRUPT
3511  022102  000240                                  NOP
3512  022104  104010                                  HLT     10              ;NO INTERRUPT
```

```
3513  022106  000403                          BR      4$
3514  022110  104011                  2$:     HLT     11                  ;WRONG VECTOR
3515  022112  012706  001200          3$:     MOV     #STACK,SP           ;RESET STACK
3516  022116  104400                  4$:     SCOPE                       ;SCOPE THIS TEST
3517
3518
3519                                          ;*************************** TEST 56 ****************************
3520                                          ;*PRIORITY INTERRUPT TESTS
3521                                          ;*SET PS TO ALL BR LEVELS EQUAL OR GREATER THAN
3522                                          ;*THE DMC11 LEVEL,VERIFY THAT DMC11 DOES NOT INTERRUPT
3523                                          ;;************************************************************
3524
3525                                          ;  TEST 56
3526                                          ;--------------
3527  022120  012737  000056  001226  TST56:  MOV     #56,TSTNO
3528  022126  012737  022240  001216          MOV     #TST57,NEXT
3529                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
3530  022134  104412                          MSTCLR                     ;MASTER CLEAR DMC11
3531  022136  012702  000340                  MOV     #340,R2            ;PUT LEVEL 7 IN R2
3532  022142  010237  177776                  MOV     R2,PS              ;SET PRIORITY TO 7
3533  022146  013700  001366                  MOV     STAT1,R0           ;GET BR LEVEL OF DMC11
3534  022152  006200                          ASR     R0                 ;SHIFT R0 4 TIMES
3535  022154  006200                          ASR     R0                 ;TO GET PROPER LEVEL
3536  022156  006200                          ASR     R0
3537  022160  006200                          ASR     R0
3538  022162  042700  177437                  BIC     #177437,R0         ;CLEAR UNWANTED BITS
3539  022166  004537  034600                  JSR     R5,SETVEC          ;SET UP VECTORS
3540  022172  022234                          2$                         ;A VECTOR
3541  022174  022234                          2$                         ;B VECTOR
3542  022176     340     340                  .BYTE   340,340            ;PRIORITY 7
3543  022200  012761  000200  000004  4$:     MOV     #200,4(R1)         ;LOAD PORT4
3544  022206  104414                          ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
3545  022210  121111                          121111                     ;SET BR REQUEST
3546  022212  010237  177776          5$:     MOV     R2,PS              ;PUT LEVEL IN R2 IN PS
3547  022216  000240                          NOP
3548  022220  020002                          CMP     R0,R2              ;IS PRESENT PS LEVEL = TO DMC LEVEL
3549  022222  001403                          BEQ     1$                 ;BR IF YES
3550  022224  162702  000040                  SUB     #40,R2             ;NO GET NEXT LOWER LEVEL IN R2
3551  022230  000770                          BR      5$                 ;AND CONTINUE WITH TEST
3552  022232  104400                  1$:     SCOPE                      ;SCOPE THIS TEST
3553  022234  104020                  2$:     HLT     20                 ;ERROR UNEXPECTED INTERRUPT
3554  022236  000002                          RTI
3555
3556
3557                                          ;*************************** TEST 57 ****************************
3558                                          ;*PRIORITY INTERRUPT TESTS
3559                                          ;*SET PS TO ALL BR LEVELS LESS THAN THE DMC11 LEVEL
3560                                          ;*VERIFY THAT THE DMC11 WILL INTERRUPT
3561                                          ;;************************************************************
3562
3563                                          ;  TEST 57
3564                                          ;--------------
3565  022240  012737  000057  001226  TST57:  MOV     #57,TSTNO
3566  022246  012737  022404  001216          MOV     #TST60,NEXT
3567                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
3568  022254  104412                          MSTCLR                     ;MASTER CLEAR DMC11
```

```
3569  022256  012702  000340                          MOV     #340,R2         ;PUT LEVEL 7 IN R2
3570  022262  010237  177776                          MOV     R2,PS           ;SET PRIORITY TO 7
3571  022266  013700  001366                          MOV     STAT1,R0        ;GET BR LEVEL OF DMC11
3572  022272  006200                                  ASR     R0              ;SHIFT R0 4 TIMES
3573  022274  006200                                  ASR     R0              ;TO GET PROPER LEVEL
3574  022276  006200                                  ASR     R0
3575  022300  006200                                  ASR     R0
3576  022302  042700  177437                          BIC     #177437,R0      ;CLEAR UNWANTED BITS
3577  022306  010002                                  MOV     R0,R2           ;PUT DMC LEVEL IN R2
3578  022310  162702  000040                          SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
3579  022314  004537  034600                          JSR     R5,SETVEC       ;SET UP VECTORS
3580  022320  022366                                  2$                      ;A VECTOR
3581  022322  022374                                  3$                      ;B VECTOR
3582  022324     340     340                          .BYTE   340,340         ;PRIORITY 7
3583  022326  012761  000200  000004          4$:     MOV     #200,4(R1)      ;LOAD PORT4
3584  022334  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3585  022336  121111                                  121111                  ;SET BR REQUEST
3586  022340  010237  177776                  5$:     MOV     R2,PS           ;PUT LEVEL IN R2 IN PS
3587  022344  000240                                  NOP
3588  022346  104010                                  HLT     10              ;ERROR, NO INTERRUPT
3589  022350  022702  000140                  6$:     CMP     #140,R2         ;IS IT DOWN TO LEVEL 3 YET?
3590  022354  001403                                  BEQ     1$              ;YES,DMC DID NOT INTERUPT, ERROR
3591  022356  162702  000040                          SUB     #40,R2          ;PUT NEXT LOWER LEVEL IN R2
3592  022362  000761                                  BR      4$              ;CONTINUE TEST
3593  022364  104400                          1$:     SCOPE                   ;SCOPE THIS TEST
3594  022366  012716  022350                  2$:     MOV     #6$,(SP)        ;SET UP FOR RTI
3595  022372  000002                                  RTI
3596  022374  104011                          3$:     HLT     11              ;ERROR, WRONG VECTOR
3597  022376  012716  022350                          MOV     #6$,(SP)        ;SET UP FOR RTI
3598  022402  000002                                  RTI
3599
3600
3601                                                  ;************************** TEST 60 **************************
3602                                                  ;*NPR TEST
3603                                                  ;*TEST OF DATO, 1 WORD FROM UPROC TO 11 MEMORY
3604                                                  ;:*********************************************************
3605
3606                                                  ;  TEST 60
3607                                                  ;-------------
3608  022404  012737  000060  001226          TST60:  MOV     #60,TSTNO
3609  022412  012737  022510  001216                  MOV     #TST61,NEXT
3610                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
3611  022420  000005                                  RESET                   ;BUS RESET
3612  022422  005011                                  CLR     (R1)            ;CLEAR RUN
3613  022424  005061  000004                          CLR     4(R1)           ;CLR PORT4
3614  022430  004537  034622                          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
3615  022434  000000                                  0                       ;IN DATA
3616  022436  177777                                  -1                      ;OUT DATA
3617  022440  022506                                  3$                      ;IN BA
3618  022442  022504                                  2$                      ;OUT BA
3619  022444  005037  022504                          CLR     2$              ;CLEAR 2$
3620  022450  012761  000021  000004                  MOV     #21,4(R1)       ;WRITE PORT4
3621  022456  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3622  022460  121110                                  121110                  ;SET NPR BITS IN IBUS* REG 11
3623  022462  000240                                  NOP
3624  022464  012705  177777                          MOV     #-1,R5          ;PUT "EXPECTED" IN R5
```

```
3625  022470  013704  022504                      MOV     2$,R4           ;PUT "FOUND" IN R4
3626  022474  020504                              CMP     R5,R4           ;DATA CORRECT?
3627  022476  001401                              BEQ     4$              ;BR IF YES
3628  022500  104012                              HLT     12              ;ERROR NPR FAILED
3629  022502  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
3630  022504  000000                      2$:     0                       ;OUT BA
3631  022506  000000                      3$:     0                       ;IN BA
3632
3633
3634                                              ;********************** TEST 61 ****************************
3635                                              ;*NPR TEST
3636                                              ;*TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
3637                                              ;:**********************************************************
3638
3639                                              ;  TEST 61
3640                                              ;--------------
3641  022510  012737  000061  001226      TST61:  MOV     #61,TSTNO
3642  022516  012737  022624  001216              MOV     #TST62,NEXT
3643                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
3644  022524  104412                              MSTCLR                  ;MASTER CLEAR DMC11
3645  022526  005061  000004                      CLR     4(R1)           ;CLR PORT4
3646  022532  004537  034622                      JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
3647  022536  000000                              0                       ;IN DATA
3648  022540  177777                              -1                      ;OUT DATA
3649  022542  022622                              3$                      ;IN BA
3650  022544  022620                              2$                      ;OUT BA
3651  022546  012737  177777  022622              MOV     #-1,3$          ;PUT DATA IN 3$
3652  022554  012761  000001  000004              MOV     #1,4(R1)        ;WRITE PORT4
3653  022562  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3654  022564  121110                              121110                  ;SET NPR BITS IN IBUS* REG 11
3655  022566  000240                              NOP
3656  022570  012705  177777                      MOV     #-1,R5          ;PUT "EXPECTED" IN R5
3657  022574  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3658  022576  021004                              021004                  ;MOVE IN DATA LOW BYTE TO PORT4
3659  022600  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3660  022602  021025                              021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
3661  022604  016104  000004                      MOV     4(R1),R4        ;PUT "FOUND" IN R4
3662  022610  020504                              CMP     R5,R4           ;DATA CORRECT?
3663  022612  001401                              BEQ     4$              ;BR IF YES
3664  022614  104012                              HLT     12              ;ERROR NPR FAILED
3665  022616  104400                      4$:     SCOPE                   ;SCOPE THIS TEST
3666  022620  000000                      2$:     0                       ;OUT BA
3667  022622  000000                      3$:     0                       ;IN BA
3668
3669
3670                                              ;********************** TEST 62 ****************************
3671                                              ;*NPR TEST
3672                                              ;*TEST OF DATOB, 1 BYTE FROM UPROC TO 11 MEMORY
3673                                              ;:**********************************************************
3674
3675                                              ;  TEST 62
3676                                              ;--------------
3677  022624  012737  000062  001226      TST62:  MOV     #62,TSTNO
3678  022632  012737  022726  001216              MOV     #TST63,NEXT
3679                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
3680  022640  104412                              MSTCLR                  ;MASTER CLEAR DMC11
```

```
3681  022642  005061  000004                          CLR     4(R1)           ;CLR PORT4
3682  022646  004537  034622                          JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
3683  022652  000000                                  0                       ;IN DATA
3684  022654  177777                                  -1                      ;OUT DATA
3685  022656  022724                                  3$                      ;IN BA
3686  022660  022723                                  2$+1            ;OUT BA
3687  022662  005037  022722                          CLR     2$              ;CLEAR 2$
3688  022666  012761  000221  000004                  MOV     #221,4(R1)      ;WRITE PORT4
3689  022674  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3690  022676  121110                                  121110                  ;SET NPR BITS IN IBUS* REG 11
3691  022700  000240                                  NOP
3692  022702  012705  177400                          MOV     #177400,R5              ;PUT "EXPECTED" IN R5
3693  022706  013704  022722                          MOV     2$,R4           ;PUT "FOUND" IN R4
3694  022712  020504                                  CMP     R5,R4           ;DATA CORRECT?
3695  022714  001401                                  BEQ     4$              ;BR IF YES
3696  022716  104012                                  HLT     12              ;ERROR NPR FAILED
3697  022720  104400                          4$:     SCOPE                   ;SCOPE THIS TEST
3698  022722  000000                          2$:     0                       ;OUT BA
3699  022724  000000                          3$:     0                       ;IN BA
3700
3701                                                  ;*************************** TEST 63 ****************************
3702                                                  ;*TEST OF EA BITS 16 AND 17
3703                                                  ;*DO A DATO TO AN ADDRESS USING OUT BA BITS 16 AND 17
3704                                                  ;*VERIFY CORRECT RESULTS
3705                                                  ;;************************************************************
3706
3707                                                  ;  TEST 63
3708                                                  ;--------------
3709
3710  022726  012737  000063  001226          TST63:  MOV     #63,TSTNO
3711  022734  012737  023064  001216                  MOV     #TST64,NEXT
3712                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3713  022742  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
3714  022744  013737  001412  022772                  MOV     DMP04,1$        ;USE SEL4 FOR ADDRESS
3715  022752  013737  001412  022770                  MOV     DMP04,2$        ;USE SEL4 FOR ADDRESS
3716  022760  004537  034622                          JSR     R5,NPRSET       ;LOAD BA AND DATA
3717  022764  000000                                  0                       ;IN DATA
3718  022766  125252                                  125252                  ;OUT DATA
3719  022770  000000                          2$:     0                       ;IN BA
3720  022772  000000                          1$:     0                       ;OUT BA
3721  022774  012761  000014  000004                  MOV     #14,4(R1)       ;LOAD SEL 4 WITH OUT BA16 AND 17
3722  023002  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3723  023004  121111                                  121111                  ;SET OUTBA 16 AND 17
3724  023006  012761  000021  000004                  MOV     #21,4(R1)       ;LOAD SEL4
3725  023014  012761  121110  000006                  MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
3726  023022  012711  003000                          MOV     #BIT9!BIT10,(R1);SET CROMI AND CROMO!!
3727  023026  052711  000400                          BIS     #BIT8,(R1)      ;CLOCK IT!
3728  023032  000240                                  NOP                     ;WAIT FOR NPR
3729  023034  012705  121110                          MOV     #121110,R5      ;PUT "EXPECTED" IN R5
3730  023040  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3731  023042  021044                                  021044                  ;MOVE OUT DATA LB TO SEL4
3732  023044  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3733  023046  021065                                  021065                  ;MOVE OUT DATA HB TO SEL5
3734  023050  016104  000004                          MOV     4(R1),R4        ;PUT "FOUND" IN R4
3735  023054  020504                                  CMP     R5,R4           ;CORRECT RESULTS ?
3736  023056  001401                                  BEQ     3$              ;BR IF YES
```

```
3737  023060  104012                          HLT     12              ;ERROR BA 16 AND 17 FAILED
3738  023062  104400                  3$:     SCOPE                   ;SCOPE THIS TEST
3739
3740
3741                                          ;*************************** TEST 64 ****************************
3742                                          ;*TEST OF EA BITS 16 AND 17
3743                                          ;*DO A DATI USING IN BA BITS 16 AND 17
3744                                          ;*VERIFY CORRECT RESULTS
3745                                          ;;************************************************************
3746
3747                                          ;  TEST 64
3748                                          ;--------------
3749  023064  012737  000064  001226  TST64:  MOV     #64,TSTNO
3750  023072  012737  023210  001216          MOV     #TST65,NEXT
3751                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3752  023100  104412                          MSTCLR                  ;MASTER CLEAR DMC11
3753  023102  013737  001412  023130          MOV     DMP04,1$        ;USE SEL4 FOR ADDRESS
3754  023110  013737  001412  023126          MOV     DMP04,2$        ;USE SEL4 FOR ADDRESS
3755  023116  004537  034622                  JSR     R5,NPRSET       ;LOAD BA AND DATA
3756  023122  000000                          0                       ;IN DATA
3757  023124  125252                          125252                  ;OUT DATA
3758  023126  000000                  2$:     0                       ;IN BA
3759  023130  000000                  1$:     0                       ;OUT BA
3760  023132  012761  000015  000004          MOV     #15,4(R1)       ;LOAD SEL4
3761  023140  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
3762  023146  012711  003000                  MOV     #BIT9!BIT10,(R1);SET CROMI AND CROMO!!
3763  023152  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT!
3764  023156  000240                          NOP                     ;WAIT FOR NPR
3765  023160  012705  121110                  MOV     #121110,R5      ;PUT "EXPECTED" IN R5
3766  023164  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3767  023166  021004                          021004                  ;MOVE IN DATA LB TO SEL4
3768  023170  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3769  023172  021025                          021025                  ;MOVE IN DATA HB TO SEL5
3770  023174  016104  000004                  MOV     4(R1),R4        ;PUT "FOUND" IN R4
3771  023200  020504                          CMP     R5,R4           ;CORRECT RESULTS ?
3772  023202  001401                          BEQ     3$              ;BR IF YES
3773  023204  104012                          HLT     12              ;ERROR BA 16 AND 17 FAILED
3774  023206  104400                  3$:     SCOPE                   ;SCOPE THIS TEST
3775
3776
3777                                          ;*************************** TEST 65 ****************************
3778                                          ;*NPR NON-EXISTENT MEMORY TEST
3779                                          ;*DO A DATO TO A NON-EXISTENT ADDRESS
3780                                          ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
3781                                          ;;************************************************************
3782
3783                                          ;  TEST 65
3784                                          ;--------------
3785  023210  012737  000065  001226  TST65:  MOV     #65,TSTNO
3786  023216  012737  023320  001216          MOV     #TST66,NEXT
3787                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3788  023224  104412                          MSTCLR                  ;MASTER CLEAR DMC11
3789  023226  004537  034622                  JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
3790  023232  000000                          0                       ;IN DATA
3791  023234  000000                          0                       ;OUT DATA
3792  023236  177320                          177320                  ;IN BA
```

```
3793  023240  177320                                  177320                  ;OUT BA
3794  023242  012761  000014  000004                  MOV     #14,4(R1)       ;SET OUT BA BITS 16+17 IN PORT4
3795  023250  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3796  023252  121111                                  121111                  ;SET OUTBA 16 AND 17
3797  023254  012761  000021  000004                  MOV     #21,4(R1)       ;SET NPR REQUEST BITS IN PORT4
3798  023262  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3799  023264  121110                                  121110                  ;MOV IBUS* 4 TO IBUS* 10
3800  023266  000240                                  NOP
3801  023270  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3802  023272  121225                                  121225                  ;MOV IBUS*11 TO IBUS*5
3803  023274  012705  000001                          MOV     #1,R5           ;PUT "EXPECTED" IN R5
3804  023300  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3805  023304  042704  177776                          BIC     #177776,R4      ;CLEAR UNWANTED BITS
3806  023310  020504                                  CMP     R5,R4           ;DATA CORRECT?
3807  023312  001401                                  BEQ     1$              ;BR IF YES
3808  023314  104012                                  HLT     12              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
3809  023316  104400                          1$:     SCOPE                   ;SCOPE THIS TEST
3810
3811
3812                                                  ;*************************** TEST 66 ****************************
3813                                                  ;*NPR NON-EXISTENT MEMORY TEST
3814                                                  ;*DO A DATI FROM A NON-EXISTENT ADDRESS
3815                                                  ;*VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
3816                                                  ;;************************************************************
3817
3818                                                  ;  TEST 66
3819                                                  ;--------------
3820  023320  012737  000066  001226          TST66:  MOV     #66,TSTNO
3821  023326  012737  023426  001216                  MOV     #TST67,NEXT
3822                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
3823  023334  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
3824  023336  004537  034622                          JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
3825  023342  000000                                  0                       ;IN DATA
3826  023344  000000                                  0                       ;OUT DATA
3827  023346  177320                                  177320                  ;IN BA
3828  023350  177320                                  177320                  ;OUT BA
3829  023352  005061  000004                          CLR     4(R1)
3830  023356  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3831  023360  121111                                  121111                  ;CLEAR NON-EXISTENT BIT
3832  023362  012761  000015  000004                  MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
3833  023370  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3834  023372  121110                                  121110                  ;MOV IBUS* 4 TO IBUS* 10
3835  023374  000240                                  NOP
3836  023376  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3837  023400  121225                                  121225                  ;MOV IBUS*11 TO IBJS*5
3838  023402  012705  000001                          MOV     #1,R5           ;PUT "EXPECTED" IN R5
3839  023406  116104  000005                          MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3840  023412  042704  177776                          BIC     #177776,R4      ;CLEAR UNWANTED BITS
3841  023416  020504                                  CMP     R5,R4           ;DATA CORRECT?
3842  023420  001401                                  BEQ     1$              ;BR IF YES
3843  023422  104012                                  HLT     12              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
3844  023424  104400                          1$:     SCOPE                   ;SCOPE THIS TEST
3845
3846
3847                                                  ;*************************** TEST 67 ****************************
3848                                                  ;*NPR TEST
```

```
3849                                                    ;*USING DATO, NPR A BINARY COUNT (0-377 )
3850                                                    ;*FROM MICRO-PROCESSOR TO ALL AVAILABLE MEMORY
3851                                                    ;;*****************************************************************
3852
3853                                                    ;  TEST 67
3854                                                    ;--------------
3855  023426  012737  000067  001226        TST67:    MOV     #67,TSTNO
3856  023434  012737  000003  001222                  MOV     #3,ICOUNT
3857  023442  012737  023624  001216                  MOV     #TST70,NEXT
3858                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
3859  023450  104412                                 MSTCLR                  ;MASTER CLEAR DMC11
3860  023452  005037  023622                         CLR     5$              ;START FLAG AT 0
3861  023456  005000                                 CLR     R0              ;DATA
3862  023460  012702  036522                         MOV     #CORMAX,R2      ;ADDRESS
3863  023464                                  1$:
3864  023464  010037  023514                         MOV     R0,2$           ;LOAD DATA
3865  023470  010237  023520                         MOV     R2,4$           ;LOAD BA
3866  023474  032702  000001                         BIT     #BIT0,R2        ;IS BA ODD?
3867  023500  001402                                 BEQ     .+6             ;BR IF NO
3868  023502  000337  023514                         SWAB    2$              ;IF ODD PUT DATA IN HI-BYTE
3869  023506  004537  034622                         JSR     R5,NPRSET       ;LOAD NPR REGISTERS
3870  023512  000000                                 0                       ;IN DATA
3871  023514  000000                          2$:    0                       ;OUT DATA
3872  023516  000000                                 0                       ;IN BA
3873  023520  000000                          4$:    0                       ;OUT BA
3874  023522  105012                                 CLRB    (R2)            ;CLEAR MEMORY LOCATION
3875  023524  012761  000221  000004                 MOV     #221,4(R1)      ;LOAD PORT4
3876  023532  104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3877  023534  121110                                 121110                  ;DO THE NPR
3878  023536  000240                                 NOP
3879  023540  010005                                 MOV     R0,R5           ;PUT "EXPECTED" IN R5
3880  023542  111204                                 MOVB    (R2),R4         ;PUT "FOUND" IN R4
3881  023544  120504                                 CMPB    R5,R4           ;IS DATA CORRECT?
3882  023546  001401                                 BEQ     3$              ;BR IF YES
3883  023550  104021                                 HLT     21              ;ERROR, DATA INCORRECT
3884  023552  104401                          3$:    SCOP1
3885  023554  005200                                 INC     R0              ;NEXT CHARACTER
3886  023556  042700  177400                         BIC     #177400,R0      ;USE ONLY LOW BYTE
3887  023562  005737  023622                         TST     5$              ;HAS MAX MEMORY BEEN REACHED YET?
3888  023566  001402                                 BEQ     6$              ;BR IF NO
3889  023570  005700                                 TST     R0              ;DONE PATTERN?
3890  023572  001412                                 BEQ     7$              ;BR IF YES
3891  023574  005202                          6$:    INC     R2              ;INC BA
3892  023576  023702  001304                         CMP     MEMLIM,R2       ;REACHED MEMORY LIMIT YET?
3893  023602  001330                                 BNE     1$              ;BR IF NOT
3894  023604  012702  036522                         MOV     #CORMAX,R2      ;RESTART BA AT FIRST ADDRESS
3895  023610  012737  177777  023622                 MOV     #-1,5$          ;SET FLAG TO END TEST AT END OF DATA PATTERN
3896  023616  000722                                 BR 1$                   ;CONTINUE
3897  023620  104400                          7$:    SCOPE                   ;SCOPE THIS TEST
3898  023622  000000                          5$:    0                       ;THIS LOCATION IS A FLAG,  IT STARTS AT 0,
3899                                                                          ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
3900                                                                          ;IS USED, TEST IS THEN ENDED WHEN PATTERN IS FIN
3901
3902
3903                                                    ;***************************** TEST 70 *****************************
3904                                                    ;*MAIN MEMORY TEST
```

```
                                                        ;*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
                                                        ;:*************************************************************

                                                        ;  TEST 70
                                                        ;-------------
023624  012737  000070  001226        TST70:  MOV     #70,TSTNO
023632  012737  023752  001216                MOV     #TST71,NEXT
023640  012737  023656  001220                MOV     #65$,LOCK

                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023646  104412                                MSTCLR                  ;MASTER CLEAR DMC11
023650  005002                                CLR     R2              ;START WITH ADDRESS 0
023652  012700  000001        1$:     MOV     #1,R0           ;START WITH BIT 0
023656  042737  000377  023672  65$:     BIC     #377,66$                ;CLEAR ADDRESS FIELD OF INSTRUCTION
023664  050237  023672                BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
023670  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023672  010000                        66$:    010000                  ;LOAD MAR WITH ADDRESS IN R2
023674  010061  000004                        MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
023700  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023702  122500                                122500                  ;MOVE PORT4 TO MEMORY
023704  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023706  040620                                040620                  ;MOVE MEMORY TO BR
023710  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023712  061225                                61225                   ;MOVE BR TO PORT5
023714  010005                                MOV     R0,R5           ;PUT "EXPECTED" IN R5
023716  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
023722  120504                                CMPB    R5,R4           ;DATA CORRECT?
023724  001401                                BEQ     67$             ;BR IF YES
023726  104013                                HLT     13              ;DATA ERROR
023730  104401                        67$:    SCOP1                   ;SW09=1?
023732  000241                                CLC                     ;CLEAR CARRY
023734  106100                                ROLB    R0              ;SHIFT BIT IN RC
023736  001347                                BNE     65$             ;DONE IF R0=0
023740  005202                                INC     R2              ;NEXT ADDRESS
023742  022702  000400                        CMP     #400,R2         ;LAST ADDRESS
023746  001341                                BNE     1$              ;BR IF NO
023750  104400                                SCOPE                   ;SCOPE THIS TEST

                                                        ;*************************** TEST 71 ****************************
                                                        ;*MAIN MEMORY TEST
                                                        ;*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
                                                        ;:*************************************************************

                                                        ;  TEST 71
                                                        ;-------------
023752  012737  000071  001226        TST71:  MOV     #71,TSTNO
023760  012737  024104  001216                MOV     #TST72,NEXT
023766  012737  024006  001220                MOV     #65$,LOCK

                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023774  104412                                MSTCLR                  ;MASTER CLEAR DMC11
024000  005002                                CLR     R2              ;START WITH ADDRESS 0
024002  012700  000001        1$:     MOV     #1,R0           ;START WITH BIT 0
024006  005100                        64$:    COM     R0              ;CHANGE TO FLOATING 0
024010  042737  000377  024022        65$:    BIC     #377,66$                ;CLEAR ADDRESS FIELD OF INSTRUCTION
024014  050237  024022                BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
024020  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
3961  024022  010000                    66$:  010000          ;LOAD MAR WITH ADDRESS IN R2
3962  024024  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
3963  024030  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3964  024032  122500                          122500                  ;MOVE PORT4 TO MEMORY
3965  024034  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3966  024036  040620                          040620                  ;MOVE MEMORY TO BR
3967  024040  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3968  024042  061225                          61225                   ;MOVE BR TO PORTS
3969  024044  010005                          MOV     R0,R5           ;PUT "EXPECTED" IN R5
3970  024046  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3971  024052  120504                          CMPB    R5,R4           ;DATA CORRECT?
3972  024054  001401                          BEQ     67$             ;BR IF YES
3973  024056  104013                          HLT     13              ;DATA ERROR
3974  024060  104401                    67$:  SCOP1                   ;SW09=1?
3975  024062  005100                          COM     R0              ;CHANGE TO FLOATING 1
3976  024064  000241                          CLC                     ;CLEAR CARRY
3977  024066  106100                          ROLB    R0              ;SHIFT BIT IN R0
3978  024070  001345                          BNE     64$             ;DONE IF R0=0
3979  024072  005202                          INC     R2              ;NEXT ADDRESS
3980  024074  022702  000400                  CMP     #400,R2         ;LAST ADDRESS
3981  024100  001337                          BNE     1$              ;BR IF NO
3982  024102  104400                          SCOPE                   ;SCOPE THIS TEST
3983
3984
3985                                          ;*************************** TEST 72 ****************************
3986                                          ;*MAIN MEMORY DUAL ADDRESSING TEST
3987                                          ;*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
3988                                          ;*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
3989                                          ;:************************************************************
3990
3991                                          ;  TEST 72
2992                                          ;--------------
3993  024104  012737  000072  001226  TST72:  MOV     #72,TSTNO
3994  024112  012737  024304  001216          MOV     #TST73,NEXT
3995  024120  012737  024132  001220          MOV     #1$,LOCK
3996                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
3997  024126  104412                          MSTCLR                  ;MASTER CLEAR DMC11
3998  024130  005002                          CLR     R2              ;START AT ADDRESS 0
3999  024132  042737  000377  024146    1$:   BIC     #377,2$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
4000  024140  050237  024146                  BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
4001  024144  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4002  024146  010000                    2$:   010000                  ;LOAD MAR
4003  024150  010261  000004                  MOV     R2,4(R1)
4004  024154  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4005  024156  122500                          122500                  ;MOVE PORT4 TO MEMORY
4006  024160  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4007  024162  040620                          040620                  ;MOVE MEMORY TO THE BR
4008  024164  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4009  024166  061225                          61225                   ;MOV BR TO PORTS
4010  024170  010205                          MOV     R2,R5           ;PUT "EXPECTED" IN R5
4011  024172  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
4012  024176  120504                          CMPB    R5,R4           ;DATA CORRECT?
4013  024200  001401                          BEQ     3$              ;BR IF YES
4014  024202  104013                          HLT     13              ;DATA ERROR
4015  024204  104401                    3$:   SCOP1                   ;SW09=1?
4016  024206  005202                          INC     R2              ;NEXT ADDRESS
```

```
4017  024210  022702  000400                        CMP     #400,R2         ;LAST ADDRESS
4018  024214  001346                                BNE     1$              ;BR IF NO
4019  024216  012737  024226  001220                MOV     #4$,LOCK        ;NEW SCOPE 1
4020  024224  005002                                CLR     R2              ;RESTART AT ADDRESS 0
4021  024226  042737  000377  024242        4$:     BIC     #377,5$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
4022  024234  050237  024242                        BIS     R2,5$           ;ADD ADDRESS TO INSTRUCTION
4023  024240  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4024  024242  010000                        5$:     010000                  ;LOAD THE MAR
4025  024244  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4026  024246  040620                                040620                  ;MOVE MEMORY TO THE BR
4027  024250  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4028  024252  061225                                61225                   ;MOV BR TO PORT5
4029  024254  010205                                MOV     R2,R5           ;PUT "EXPECTED" IN R5
4030  024256  116104  000005                        MOVB    5(R1),R4        ;PUT "FOUND" IN R4
4031  024262  120504                                CMPB    R5,R4           ;DATA CORRECT?
4032  024264  001401                                BEQ     6$              ;BR IF YES
4033  024266  104013                                HLT     13              ;ADDRESSING ERROR
4034  024270  104401                        6$:     SCOP1                   ;SW09=1?
4035  024272  005202                                INC     R2              ;NEXT ADDRESS
4036  024274  022702  000400                        CMP     #400,R2         ;IS IT THE LAST
4037  024300  001352                                BNE     4$              ;BR IF NO
4038  024302  104400                                SCOPE                   ;SCOPE THIS TEST
4039
4040
4041                                                ;************************** TEST 73 **************************
4042                                                ;*MAR TEST
4043                                                ;*PERFORM DUAL ADDRESSING TEST
4044                                                ;*USING MAR AUTO-INC FEATURE
4045                                                ;;*********************************************************
4046
4047                                                ;  TEST 73
4048                                                ;--------------
4049  024304  012737  000073  001226        TST73:  MOV     #73,TSTNO
4050  024312  012737  024440  001216                MOV     #TST74,NEXT
4051                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
4052  024320  104412                                MSTCLR                  ;MASTER CLEAR DMC11
4053  024322  005002                                CLR     R2              ;START WITH A ZERO
4054  024324  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4055  024326  010000                                010000                  ;LOAD MAR
4056  024330  032737  100000  001366                BIT     #BIT15,STAT1    ;DMC?
4057  024336  001402                                BEQ     .+6             ;BR IF YES
4058  024340  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4059  024342  004000                                4000                    ;MAR HI = 0 (KMC ONLY)
4060  024344  010261  000004                1$:     MOV     R2,4(R1)        ;WRITE DATA TO PORT4
4061  024350  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4062  024352  136500                                136500                  ;LOAD MEM AUTO-INC MAR
4063  024354  005202                                INC     R2              ;INCREMENT DATA
4064  024356  022702  000400                        CMP     #400,R2         ;DONE YET?
4065  024362  001370                                BNE     1$              ;BR IF NO
4066  024364  005002                                CLR     R2              ;RESTART WITH A ZERO
4067  024366  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4068  024370  010000                                010000                  ;LOAD MAR
4069  024372  032737  100000  001366                BIT     #BIT15,STAT1    ;DMC?
4070  024400  001402                                BEQ     .+6             ;BR IF YES
4071  024402  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4072  024404  004000                                4000                    ;MAR HI = 0 (KMC ONLY)
```

```
4073  024406                                  2$:
4074  024406  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4075  024410  055224                                055224                  ;MOVE MEM TO PORT4
4076  024412  010205                                MOV     R2,R5           ;PUT "EXPECTED" IN R5
4077  024414  016104  000004                       MOV     4(R1),R4        ;PUT "FOUND" IN R4
4078  024420  120504                                CMPB    R5,R4           ;DATA CORRECT?
4079  024422  001401                                BEQ     3$              ;BR IF YES
4080  024424  104014                                HLT     14              ;MAR ERROR
4081  024426  005202                          3$:   INC     R2              ;NEXT ADDRESS
4082  024430  022702  000400                       CMP     #400.R2         ;DONE YET?
4083  024434  001364                                BNE     2$              ;BR IF NO
4084  024436  104400                                SCOPE                   ;SCOPE THIS TEST
4085
4086                                                ;************************** TEST 74 **************************
4087                                                ;*ALU C BIT TEST
4088                                                ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
4089                                                ;:**********************************************************
4090
4091                                                ;  TEST 74
4092                                                ;-------------
4093
4094  024440  012737  000074  001226        TST74:  MOV     #74,TSTNO
4095  024446  012737  024552  001216                MOV     #TST75,NEXT
4096  024454  012737  024500  001220                MOV     #1$,LOCK
4097                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
4098  024462  104412                                MSTCLR                  ;MASTER CLEAR DMC11
4099  024464  004737  034664                       JSR     PC,MEMLD        ;LOAD MAINMEM DATA
4100  024470  024542                                TDATA                   ;POINTER TO DATA
4101  024472  004737  034720                       JSR     PC,SPLD         ;LOAD SP DATA
4102  024476  024542                                TDATA                   ;POINTER TO DATA
4103  024500                                  1$:
4104  024500  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4105  024502  010000                                010000                  ;MAR←0
4106  024504  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4107  024506  054400                                054400!<0*20>           ;ADD 377 AND 377, TO SET C BIT
4108  024510  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4109  024512  040421                                040401!<1*20>           ;ADD 0 AND 0 AND THE C BIT
4110  024514  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4111  024516  061224                                61224                   ;PUT RESULTS IN PORT4
4112  024520  012705  000001                       MOV     #1,R5           ;PUT "EXPECTED" IN R5
4113  024524  016104  000004                       MOV     4(R1),R4        ;PUT "FOUND" IN R4
4114  024530  120504                                CMPB    R5,R4           ;DATA CORRECT?
4115  024532  001401                                BEQ     2$              ;BR IF YES
4116  024534  104015                                HLT     15              ;ERROR C BIT NOT SET
4117  024536  104401                          2$:   SCOP1                   ;SW09=1?
4118  024540  104400                                SCOPE                   ;SCOPE THIS TEST
4119  024542     377     000     000        TDATA:  .BYTE   -1,0,0,0,0,0,0,0
4120  024545     000     000     000
4121  024550     000     000
4122                                          .EVEN
4123
4124
4125                                                ;************************** TEST 75 **************************
4126                                                ;*ALU TEST
4127                                                ;*TEST OF ALU FUNCTION SEL B WITH C BIT CLEARED
4128                                                ;*ALU FUNCTION (B)    CODE=11
```

```
4129                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4130                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4131                                                    ;;**********************************************************************
4132
4133                                                    ;  TEST 75
4134                                                    ;-------------
4135  024552  012737  000075  001226        TST75:    MOV     #75,TSTNO
4136  024560  012737  024726  001216                  MOV     #TST76,NEXT
4137  024566  012737  024620  001220                  MOV     #1$,LOCK
4138                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
4139  024574  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
4140  024576  005000                                   CLR     R0              ;MEM + SP ADDRESS
4141  024600  012702  024716                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
4142  024604  004737  034664                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
4143  024610  035010                                   MEMDAT                  ;POINTER TO DATA
4144  024612  004737  034720                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
4145  024616  035020                                   SPDAT                   ;POINTER TO DATA
4146  024620  004737  034764        1$:       JSR     PC,CLRC         ;CLEAR C BIT!
4147  024624  042737  000017  024640                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4148  024632  050037  024640                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
4149  024636  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4150  024640  010000                        2$:       010000                  ;LOAD MAR
4151  024642  042737  000017  024656                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4152  024650  050037  024656                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4153  024654  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4154  024656  040620                        3$:       040400!<11*20>          ;BR + SEL B
4155  024660  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4156  024662  061224                                   61224                   ;MOVE BR TO PORT4
4157  024664  111205                                   MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
4158  024666  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
4159  024672  120504                                   CMPB    R5,R4           ;DATA CORRECT?
4160  024674  001401                                   BEQ     4$              ;BR IF YES
4161  024676  104015                                   HLT     15              ;ALU ERROR
4162  024700  104401                        4$:       SCOP1                   ;SW09=1?
4163  024702  005202                                   INC     R2              ;NEXT DATA
4164  024704  005200                                   INC     R0              ;NEXT ADDRESS
4165  024706  022700  000010                          CMP     #10,R0          ;DONE YET?
4166  024712  001342                                   BNE     1$              ;BR IF NO
4167  024714  104400                                   SCOPE                   ;SCOPE THIS TEST
4168  024716     000     377     000        5$:       .BYTE   0,-1,0,-1,125.252,125,252
4169  024721     377     125     252
4170  024724     125     252
4171                                          .EVEN
4172
4173
4174                                                    ;*************************** TEST 76 ****************************
4175                                                    ;*ALU TEST
4176                                                    ;*TEST OF ALU FUNCTION SEL A WITH C BIT CLEARED
4177                                                    ;*ALU FUNCTION (A)    CODE=10
4178                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4179                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4180                                                    ;;**********************************************************************
4181
4182                                                    ;  TEST 76
4183                                                    ;-------------
4184  024726  012737  000076  001226        TST76:    MOV     #76,TSTNO
```

```
 4185  024734  012737  025102  001216                    MOV     #TST77,NEXT
 4186  024742  012737  024774  001220                    MOV     #1$,LOCK
 4187
 4188  024750  104412                                    MSTCLR
 4189  024752  005000                                    CLR     R0              ;R1 CONTAINS BASE DMC11 ADDRESS
 4190  024754  012702  025072                            MOV     #5$,R2          ;MASTER CLEAR DMC11
 4191  024760  004737  034664                            JSR     PC,MEMLD        ;MEM + SP ADDRESS
 4192  024764  035010                                    MEMDAT                  ;POINTER TO CORRECT DATA
 4193  024766  004737  034720                            JSR     PC,SPLD         ;LOAD 8 WORDS OF MAIN MEMORY
 4194  024772  035020                                    SPDAT                   ;POINTER TO DATA
 4195  024774  004737  034764                1$:         JSR     PC,CLRC         ;LOAD 8 WORDS OF SP
 4196  025000  042737  000017  025014                    BIC     #17,2$          ;POINTER TO DATA
 4197  025006  050037  025014                            BIS     R0,2$           ;CLEAR C BIT!
 4198  025012  104414                                    ROMCLK                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4199  025014  010000                        2$:         010000                  ;ADD ADDRESS TO INSTRUCTION
 4200  025016  042737  000017  025032                    BIC     #17,3$          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4201  025024  050037  025032                            BIS     R0,3$           ;LOAD MAR
 4202  025030  104414                                    ROMCLK                  ;CLEAR ADDRESS OF INSTRUCTION
 4203  025032  040600                        3$:         040400!<10*20>          ;ADD ADDRESS TO INSTRUCTION
 4204  025034  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4205  025036  061224                                    61224                   ;BR ← SEL A
 4206  025040  111205                                    MOVB    (R2),R5         ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4207  025042  116104  000004                            MOVB    4(R1),R4        ;MOVE BR TO PORT4
 4208  025046  120504                                    CMPB    R5,R4           ;PUT "EXPECTED" IN R5
 4209  025050  001401                                    BEQ     4$              ;PUT "FOUND" IN R4
 4210  025052  104015                                    HLT     15              ;DATA CORRECT?
 4211  025054  104401                        4$:         SCOP1                   ;BR IF YES
 4212  025056  005202                                    INC     R2              ;ALU ERROR
 4213  025060  005200                                    INC     R0              ;SW09=1?
 4214  025062  022700  000010                            CMP     #10,R0          ;NEXT DATA
 4215  025066  001342                                    BNE     1$              ;NEXT ADDRESS
 4216  025070  104400                                    SCOPE                   ;DONE YET?
 4217  025072     000     000     377        5$:         .BYTE   0,0,-1,-1,125,125,252,252  ;BR IF NO
 4218  025075     377     125     125                                            ;SCOPE THIS TEST
 4219  025100     252     252
 4220                                           .EVEN
 4221
 4222                                           ;*************************** TEST 77 ****************************
 4223                                           ;*ALU TEST
 4224                                           ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED
 4225                                           ;*ALU FUNCTION (A OR NOTB)    CODE=12
 4226                                           ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 4227                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 4228                                           ;:****************************************************************
 4229
 4230
 4231                                           ;  TEST 77
 4232                                           ;-------------
 4233  025102  012737  000077  001226  TST77:    MOV     #77,TSTNO
 4234  025110  012737  025256  001216            MOV     #TST100,NEXT
 4235  025116  012737  025150  001220            MOV     #1$,LOCK
 4236                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
 4237  025124  104412                           MSTCLR                  ;MASTER CLEAR DMC11
 4238  025126  005000                           CLR     R0              ;MEM + SP ADDRESS
 4239  025130  012702  025246                   MOV     #5$,R2          ;POINTER TO CORRECT DATA
 4240  025134  004737  034664                   JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
4241  025140  035010                                    MEMDAT                  ;POINTER TO DATA
4242  025142  004737  034720                            JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
4243  025146  035020                                    SPDAT                   ;POINTER TO DATA
4244  025150  004737  034764                    1$:     JSR     PC,CLRC         ;CLEAR C BIT!
4245  025154  042737  000017  025170                    BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4246  025162  050037  025170                            BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
4247  025166  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4248  025170  010000                                    010000                  ;LOAD MAR
4249  025172  042737  000017  025206                    BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4250  025200  050037  025206                            BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4251  025204  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4252  025206  040640                            3$:     040400!<12*20>          ;BR = A OR NOTB
4253  025210  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4254  025212  061224                                    61224                   ;MOVE BR TO PORT4
4255  025214  111205                                    MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
4256  025216  116104  000004                            MOVB    4(R1),R4        ;PUT "FOUND" IN R4
4257  025222  120504                                    CMPB    R5,R4           ;DATA CORRECT?
4258  025224  001401                                    BEQ     4$              ;BR IF YES
4259  025226  104015                                    HLT     15              ;ALU ERROR
4260  025230  104401                            4$:     SCOP1                   ;SW09=1?
4261  025232  005202                                    INC     R2              ;NEXT DATA
4262  025234  005200                                    INC     R0              ;NEXT ADDRESS
4263  025236  022700  000010                            CMP     #10,R0          ;DONE YET?
4264  025242  001342                                    BNE     1$              ;BR IF NO
4265  025244  104400                                    SCOPE                   ;SCOPE THIS TEST
4266  025246     377     000     377            5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
4267  025251     377     377     125
4268  025254     252     377
4269                                            .EVEN
4270
4271
4272                                            ;************************** TEST 100 ****************************
4273                                            ;*ALU TEST
4274                                            ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED
4275                                            ;*ALU FUNCTION (A AND B)    CODE=13
4276                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4277                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4278                                            ;;************************************************************
4279
4280                                            ;  TEST 100
4281                                            ;- ------------
4282  025256  012737  000100  001226    TST100: MOV     #100,TSTNO
4283  025264  012737  025432  001216            MOV     #TST101,NEXT
4284  025272  012737  025324  001220            MOV     #1$,LOCK
4285                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
4286  025300  104412                            MSTCLR                          ;MASTER CLEAR DMC11
4287  025302  005000                            CLR     R0                      ;MEM + SP ADDRESS
4288  025304  012702  025422                    MOV     #5$,R2                  ;POINTER TO CORRECT DATA
4289  025310  004737  034664                    JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
4290  025314  035010                            MEMDAT                          ;POINTER TO DATA
4291  025316  004737  034720                    JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
4292  025322  035020                            SPDAT                           ;POINTER TO DATA
4293  025324  004737  034764            1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
4294  025330  042737  000017  025344            BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
4295  025336  050037  025344                    BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
4296  025342  104414                            ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
4297  025344  010000                          2$:   010000                  ;LOAD MAR
4298  025346  042737  000017  025362                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4299  025354  050037  025362                        BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4300  025360  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4301  025362  040660                          3$:   040400!<13*20>          ;BR = A AND B
4302  025364  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4303  025366  061224                                61224                   ;MOVE BR TO PORT4
4304  025370  111205                                MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
4305  025372  116104  000004                        MOVB    4(R1),R4        ;PUT "FOUND" IN R4
4306  025376  120504                                CMPB    R5,R4           ;DATA CORRECT?
4307  025400  001401                                BEQ     4$              ;BR IF YES
4308  025402  104015                                HLT     15              ;ALU ERROR
4309  025404  104401                          4$:   SCOP1                   ;SW09=1?
4310  025406  005202                                INC     R2              ;NEXT DATA
4311  025410  005200                                INC     R0              ;NEXT ADDRESS
4312  025412  022700  000010                        CMP     #10,R0          ;DONE YET?
4313  025416  001342                                BNE     1$              ;BR IF NO
4314  025420  104400                                SCOPE                   ;SCOPE THIS TEST
4315  025422     000     000     000          5$:   .BYTE   0,0,0,-1,125,0,0,252
4316  025425     377     125     000
4317  025430     000     252
4318                                          .EVEN
4319
4320
4321                                                ;************************** TEST 101 **************************
4322                                                ;*ALU TEST
4323                                                ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED
4324                                                ;*ALU FUNCTION (A OR B)    CODE=14
4325                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4326                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4327                                                ;;**********************************************************
4328
4329                                                ;  TEST 101
4330                                                ;--------------
4331  025432  012737  000101  001226        TST101: MOV     #101,TSTNO
4332  025440  012737  025606  001216                MOV     #TST102,NEXT
4333  025446  012737  025500  001220                MOV     #1$,LOCK
4334                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
4335  025454  104412                                MSTCLR                  ;MASTER CLEAR DMC11
4336  025456  005000                                CLR     R0              ;MEM + SP ADDRESS
4337  025460  012702  025576                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
4338  025464  004737  034664                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
4339  025470  035010                                MEMDAT                  ;POINTER TO DATA
4340  025472  004737  034720                        JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
4341  025476  035020                                SPDAT                   ;POINTER TO DATA
4342  025500  004737  034764                  1$:   JSR     PC,CLRC         ;CLEAR C BIT!
4343  025504  042737  000017  025520                BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4344  025512  050037  025520                        BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
4345  025516  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4346  025520  010000                          2$:   010000                  ;LOAD MAR
4347  025522  042737  000017  025536                BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4348  025530  050037  025536                        BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4349  025534  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4350  025536  040700                          3$:   040400!<14*20>          ;BR = A OR B
4351  025540  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4352  025542  061224                                61224                   ;MOVE BR TO PORT4
```

```
 4353  025544  111205                                MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4354  025546  116104  000004                       MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4355  025552  120504                                CMPB    R5,R4           ;DATA CORRECT?
 4356  025554  001401                                BEQ     4$              ;BR IF YES
 4357  025556  104015                                HLT     15              ;ALU ERROR
 4358  025560  104401                        4$:     SCOP1                   ;SW09=1?
 4359  025562  005202                                INC     R2              ;NEXT DATA
 4360  025564  005200                                INC     R0              ;NEXT ADDRESS
 4361  025566  022700  000010                       CMP     #10,R0          ;DONE YET?
 4362  025572  001342                                BNE     1$              ;BR IF NO
 4363  025574  104400                                SCOPE                   ;SCOPE THIS TEST
 4364  025576     000     377     377        5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
 4365  025601     377     125     377
 4366  025604     377     252
 4367                                        .EVEN
 4368
 4369
 4370                                                ;*************************** TEST 102 ****************************
 4371                                                ;*ALU TEST
 4372                                                ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED
 4373                                                ;*ALU FUNCTION (A XOR B)    CODE=15
 4374                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 4375                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 4376                                                ;;************************************************************************
 4377
 4378                                                ;  TEST 102
 4379                                                ;--------------
 4380  025606  012737  000102  001226        TST102: MOV     #102,TSTNO
 4381  025614  012737  025762  001216                MOV     #TST103,NEXT
 4382  025622  012737  025654  001220                MOV     #1$,LOCK
 4383                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
 4384  025630  104412                                MSTCLR                  ;MASTER CLEAR DMC11
 4385  025632  005000                                CLR     R0              ;MEM + SP ADDRESS
 4386  025634  012702  025752                       MOV     #5$,R2          ;POINTER TO CORRECT DATA
 4387  025640  004737  034664                       JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 4388  025644  035010                                MEMDAT                  ;POINTER TO DATA
 4389  025646  004737  034720                       JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 4390  025652  035020                                SPDAT                   ;POINTER TO DATA
 4391  025654  004737  034764                1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 4392  025660  042737  000017  025674               BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4393  025666  050037  025674                       BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
 4394  025672  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4395  025674  010000                        2$:     010000                  ;LOAD MAR
 4396  025676  042737  000017  025712               BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 4397  025704  050037  025712                       BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
 4398  025710  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4399  025712  040720                        3$:     040400!<15*20>          ;BR ← A XOR B
 4400  025714  104414                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4401  025716  061224                                61224                   ;MOVE BR TO PORT4
 4402  025720  111205                                MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4403  025722  116104  000004                       MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4404  025726  120504                                CMPB    R5,R4           ;DATA CORRECT?
 4405  025730  001401                                BEQ     4$              ;BR IF YES
 4406  025732  104015                                HLT     15              ;ALU ERROR
 4407  025734  104401                        4$:     SCOP1                   ;SW09=1?
 4408  025736  005202                                INC     R2              ;NEXT DATA
```

```
4409  025740  005200                                    INC     R0              ;NEXT ADDRESS
4410  025742  022700  000010                            CMP     #10,R0          ;DONE YET?
4411  025746  001342                                    BNE     1$              ;BR IF NO
4412  025750  104400                                    SCOPE                   ;SCOPE THIS TEST
4413  025752     000     377     377          5$:       .BYTE   0,-1,-1,0,0,-1,-1,0
4414  025755     000     000     377
4415  025760     377     000
4416                                          .EVEN
4417
4418                                          ;*************************** TEST 103 ****************************
4419                                          ;*ALU TEST
4420                                          ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED
4421                                          ;*ALU FUNCTION (A PLUS B)    CODE=00
4422                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4423                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4424                                          ;*****************************************************************
4425
4426
4427                                          ;  TEST 103
4428                                          ;--------------
4429  025762  012737  000103  001226  TST103: MOV     #103,TSTNO
4430  025770  012737  026136  001216          MOV     #TST104,NEXT
4431  025776  012737  026030  001220          MOV     #1$,LOCK
4432                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
4433  026004  .04412                         MSTCLR                  ;MASTER CLEAR DMC11
4434  026006  005000                          CLR     R0              ;MEM + SP ADDRESS
4435  026010  012702  026126                 MOV     #5$,R2          ;POINTER TO CORRECT DATA
4436  026014  004737  034664                 JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
4437  026020  035010                          MEMDAT                  ;POINTER TO DATA
4438  026022  004737  034720                 JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
4439  026026  035020                          SPDAT                   ;POINTER TO DATA
4440  026030  004737  034764          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
4441  026034  042737  000017  026050         BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4442  026042  050037  026050                 BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
4443  026046  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4444  026050  010000                  2$:     010000                  ;LOAD MAR
4445  026052  042737  000017  026066         BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4446  026060  050037  026066                 BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4447  026064  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4448  026066  040400                  3$:     040400!<00*20>          ;BR + ADD
4449  026070  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4450  026072  061224                          61224                   ;MOVE BR TO PORT4
4451  026074  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
4452  026076  116104  000004                 MOVB    4(R1),R4        ;PUT "FOUND" IN R4
4453  026102  120504                          CMPB    R5,R4           ;DATA CORRECT?
4454  026104  001401                          BEQ     4$              ;BR IF YES
4455  026106  104015                          HLT     15              ;ALU ERROR
4456  026110  104401                  4$:     SCOPI                   ;SW09=1?
4457  026112  005202                          INC     R2              ;NEXT DATA
4458  026114  005200                          INC     R0              ;NEXT ADDRESS
4459  026116  022700  000010                 CMP     #10,R0          ;DONE YET?
4460  026122  001342                          BNE     1$              ;BR IF NO
4461  026124  104400                          SCOPE                   ;SCOPE THIS TEST
4462  026126     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
4463  026131     376     252     377
4464  026134     377     124
```

```
4465                                                    .EVEN
4466
4467
4468                                                    ;*************************** TEST 104 ****************************
4469                                                    ;*ALU TEST
4470                                                    ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED
4471                                                    ;*ALU FUNCTION (A PLUS A PLUS C)    CODE=6
4472                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4473                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4474                                                    ;;*************************************************************
4475
4476                                                    ;  TEST 104
4477                                                    ;--------------
4478  026136  012737  000104  001226          TST104:  MOV     #104,TSTNO
4479  026144  012737  026312  001216                   MOV     #TST105,NEXT
4480  026152  012737  026204  001220                   MOV     #1$,LOCK
4481                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
4482  026160  104412                                   MSTCLR                ;MASTER CLEAR DMC11
4483  026162  005000                                   CLR     R0            ;MEM + SP ADDRESS
4484  026164  012702  026302                          MOV     #5$,R2        ;POINTER TO CORRECT DATA
4485  026170  004737  034664                          JSR     PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
4486  026174  035010                                   MEMDAT                ;POINTER TO DATA
4487  026176  004737  034720                          JSR     PC,SPLD       ;LOAD 8 WORDS OF SP
4488  026202  035020                                   SPDAT                 ;POINTER TO DATA
4489  026204  004737  034764          1$:      JSR     PC,CLRC       ;CLEAR C BIT!
4490  026210  042737  000017  026224                   BIC     #17,2$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
4491  026216  050037  026224                          BIS     R0,2$         ;ADD ADDRESS TO INSTRUCTION
4492  026222  104414                                   ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4493  026224  010000                          2$:      010000                ;LOAD MAR
4494  026226  042737  000017  026242                   BIC     #17,3$        ;CLEAR ADDRESS OF INSTRUCTION
4495  026234  050037  026242                          BIS     R0,3$         ;ADD ADDRESS TO INSTRUCTION
4496  026240  104414                                   ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4497  026242  040540                          3$:      040400!<6*20>         ;BR + 2A W/C
4498  026244  104414                                   ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4499  026246  061224                                   61224                 ;MOVE BR TO PORT4
4500  026250  111205                                   MOVB    (R2),R5       ;PUT "EXPECTED" IN R5
4501  026252  116104  000004                          MOVB    4(R1),R4      ;PUT "FOUND" IN R4
4502  026256  120504                                   CMPB    R5,R4         ;DATA CORRECT?
4503  026260  001401                                   BEQ     4$            ;BR IF YES
4504  026262  104015                                   HLT     15            ;ALU ERROR
4505  026264  104401                          4$:      SCOP1                 ;SW09=1?
4506  026266  005202                                   INC     R2            ;NEXT DATA
4507  026270  005200                                   INC     R0            ;NEXT ADDRESS
4508  026272  022700  000010                          CMP     #10,R0        ;DONE YET?
4509  026276  001342                                   BNE     1$            ;BR IF NO
4510  026300  104400                                   SCOPE                 ;SCOPE THIS TEST
4511  026302     000     000     376          5$:      .BYTE   0,0,376,376,252,252,124,124
4512  026305     376     252     252
4513  026310     124     124
4514                                                    .EVEN
4515
4516
4517                                                    ;*************************** TEST 105 ****************************
4518                                                    ;*ALU TEST
4519                                                    ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
4520                                                    ;*ALU FUNCTION (A-B)    CODE=16
```

```
4521                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4522                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4523                                                    ;;**********************************************************************
4524
4525                                                    ;  TEST 105
4526                                                    ;-------------
4527  026312  012737  000105  001226        TST105:   MOV     #105,TSTNO
4528  026320  012737  026466  001216                  MOV     #TST106,NEXT
4529  026326  012737  026360  001220                  MOV     #1$,LOCK
4530                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
4531  026334  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
4532  026336  005000                                   CLR     R0              ;MEM + SP ADDRESS
4533  026340  012702  026456                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
4534  026344  004737  034664                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
4535  026350  035010                                   MEMDAT                  ;POINTER TO DATA
4536  026352  004737  034720                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
4537  026356  035020                                   SPDAT                   ;POINTER TO DATA
4538  026360  004737  034764              1$:         JSR     PC,CLRC         ;CLEAR C BIT!
4539  026364  042737  000017  026400                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
4540  026372  050037  026400                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
4541  026376  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4542  026400  010000                      2$:         010000                  ;LOAD MAR
4543  026402  042737  000017  026416                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
4544  026410  050037  026416                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
4545  026414  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4546  026416  040740                      3$:         040400!<16*20>          ;BR + SUB
4547  026420  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4548  026422  061224                                   61224                   ;MOVE BR TO PORT4
4549  026424  111205                                   MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
4550  026426  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
4551  026432  120504                                   CMPB    R5,R4           ;DATA CORRECT?
4552  026434  001401                                   BEQ     4$              ;BR IF YES
4553  026436  104015                                   HLT     15              ;ALU ERROR
4554  026440  104401                      4$:         SCOP1                   ;SW09=1?
4555  026442  005202                                   INC     R2              ;NEXT DATA
4556  026444  005200                                   INC     R0              ;NEXT ADDRESS
4557  026446  022700  000010                          CMP     #10,R0          ;DONE YET?
4558  026452  001342                                   BNE     1$              ;BR IF NO
4559  026454  104400                                   SCOPE                   ;SCOPE THIS TEST
4560  026456     000     001     377        5$:         .BYTE   0,1,-1,0,0,253,125,0
4561  026461     000     000     253
4562  026464     125     000
4563                                          .EVEN
4564
4565
4566                                                    ;*************************** TEST 106 ****************************
4567                                                    ;*ALU TEST
4568                                                    ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED
4569                                                    ;*ALU FUNCTION (A PLUS B PLUS C)    CODE=01
4570                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4571                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4572                                                    ;;**********************************************************************
4573
4574                                                    ;  TEST 106
4575                                                    ;-------------
4576  026466  012737  000106  001226        TST106:   MOV     #106,TSTNO
```

```
4577  026474  012737  026642  001216                    MOV     #TST107,NEXT
4578  026502  012737  026534  001220                    MOV     #1$,LOCK
4579
4580  026510  104412                            MSTCLR
4581  026512  005000                            CLR     R0              ;R1 CONTAINS BASE DMC11 ADDRESS
4582  026514  012702  026632                    MOV     #5$,R2          ;MASTER CLEAR DMC11
4583  026520  004737  034664                    JSR     PC,MEMLD        ;MEM + SP ADDRESS
4584  026524  035010                            MEMDAT                  ;POINTER TO CORRECT DATA
4585  026526  004737  034720                    JSR     PC,SPLD         ;LOAD 8 WORDS OF MAIN MEMORY
4586  026532  035020                            SPDAT                   ;POINTER TO DATA
4587  026534  004737  034764          1$:       JSR     PC,CLRC         ;LOAD 8 WORDS OF SP
4588  026540  042737  000017  026554            BIC     #17,2$          ;POINTER TO DATA
4589  026546  050037  026554                    BIS     R0,2$           ;CLEAR C BIT!
4590  026552  104414                            ROMCLK                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
4591  026554  010000                    2$:     010000                  ;ADD ADDRESS TO INSTRUCTION
4592  026556  042737  000017  026572            BIC     #17,3$          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4593  026564  050037  026572                    BIS     R0,3$           ;LOAD MAR
4594  026570  104414                            ROMCLK                  ;CLEAR ADDRESS OF INSTRUCTION
4595  026572  040420                    3$:     040400!<01*20>          ;ADD ADDRESS TO INSTRUCTION
4596  026574  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4597  026576  061224                            61224                   ;BR + ADD W/C
4598  026600  111205                            MOVB    (R2),R5         ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4599  026602  116104  000004                    MOVB    4(R1),R4        ;MOVE BR TO PORT4
4600  026606  120504                            CMPB    R5,R4           ;PUT "EXPECTED" IN R5
4601  026610  001401                            BEQ     4$              ;PUT "FOUND" IN R4
4602  026612  104015                            HLT     15              ;DATA CORRECT?
4603  026614  104401                    4$:     SCOP1                   ;BR IF YES
4604  026616  005202                            INC     R2              ;ALU ERROR
4605  026620  005200                            INC     R0              ;SW09=1?
4606  026622  022700  000010                    CMP     #10,R0          ;NEXT DATA
4607  026626  001342                            BNE     1$              ;NEXT ADDRESS
4608  026630  104400                            SCOPE                   ;DONE YET?
4609  026632     000     377     377    5$:     .BYTE   0,-1,-1,376,252,-1,-1,124  ;BR IF NO
4610  026635     376     252     377            ;SCOPE THIS TEST
4611  026640     377     124
4612                                    .EVEN
4613
4614
4615                                            ;************************** TEST 107 **************************
4616                                            ;*ALU TEST
4617                                            ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED
4618                                            ;*ALU FUNCTION (A-B-C)    CODE=2
4619                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4620                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4621                                            ;:************************************************************
4622
4623                                            ;  TEST 107
4624                                            ;--------------
4625  026642  012737  000107  001226    TST107: MOV     #107,TSTNO
4626  026650  012737  027016  001216            MOV     #TST110,NEXT
4627  026656  012737  026710  001220            MOV     #1$,LOCK
4628                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
4629  026664  104412                            MSTCLR                  ;MASTER CLEAR DMC11
4630  026666  005000                            CLR     R0              ;MEM + SP ADDRESS
4631  026670  012702  027006                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
4632  026674  004737  034664                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
 4633  026700  035010                                    MEMDAT                  ;POINTER TO DATA
 4634  026702  004737  034720                            JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 4635  026706  035020                                    SPDAT                   ;POINTER TO DATA
 4636  026710  004737  034764                    1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 4637  026714  042737  000017  026730                    BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4638  026722  050037  026730                            BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
 4639  026726  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4640  026730  010000                    2$:     010000                          ;LOAD MAR
 4641  026732  042737  000017  026746                    BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 4642  026740  050037  026746                            BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
 4643  026744  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4644  026746  040440                    3$:     040400!<2*20>                   ;BR + SUB W/C
 4645  026750  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4646  026752  061224                                    61224                   ;MOVE BR TO PORT4
 4647  026754  111205                                    MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4648  026756  116104  000004                            MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4649  026762  120504                                    CMPB    R5,R4           ;DATA CORRECT?
 4650  026764  001401                                    BEQ     4$              ;BR IF YES
 4651  026766  104015                                    HLT     15              ;ALU ERROR
 4652  026770  104401                    4$:     SCOP1                           ;SW09=1?
 4653  026772  005202                                    INC     R2              ;NEXT DATA
 4654  026774  005200                                    INC     R0              ;NEXT ADDRESS
 4655  026776  022700  000010                            CMP     #10,R0          ;DONE YET?
 4656  027002  001342                                    BNE     1$              ;BR IF NO
 4657  027004  104400                                    SCOPE                   ;SCOPE THIS TEST
 4658  027006     377     000     376            5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
 4659  027011     377     377     252
 4660  027014     124     377
 4661                                            .EVEN
 4662
 4663
 4664                                            ;************************** TEST 110 **************************
 4665                                            ;*ALU TEST
 4666                                            ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED
 4667                                            ;*ALU FUNCTION (A PLUS 1)    CODE=3
 4668                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 4669                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 4670                                            ;:************************************************************
 4671
 4672                                            ;  TEST 110
 4673                                            ;-------------
 4674  027016  012737  000110  001226    TST110: MOV     #110,TSTNO
 4675  027024  012737  027172  001216            MOV     #TST111,NEXT
 4676  027032  012737  027064  001220            MOV     #1$,LOCK
 4677                                                                            ;R1 CONTAINS BASE DMC11 ADDRESS
 4678  027040  104412                                    MSTCLR                  ;MASTER CLEAR DMC11
 4679  027042  005000                                    CLR     R0              ;MEM + SP ADDRESS
 4680  027044  012702  027162                            MOV     #5$,R2          ;POINTER TO CORRECT DATA
 4681  027050  004737  034664                            JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 4682  027054  035010                                    MEMDAT                  ;POINTER TO DATA
 4683  027056  004737  034720                            JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 4684  027062  035020                                    SPDAT                   ;POINTER TO DATA
 4685  027064  004737  034764                    1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 4686  027070  042737  000017  027104                    BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4687  027076  050037  027104                            BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
 4688  027102  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
4689  027104  010000                      2$:   010000                ;LOAD MAR
4690  027106  042737  000017  027122            BIC     #17,3$        ;CLEAR ADDRESS OF INSTRUCTION
4691  027114  050037  027122                    BIS     R0,3$         ;ADD ADDRESS TO INSTRUCTION
4692  027120  104414                            ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4693  027122  040460                      3$:   040400!<3*20>         ;BR ← INC A
4694  027124  104414                            ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4695  027126  061224                            61224                 ;MOVE BR TO PORT4
4696  027130  111205                            MOVB    (R2),R5       ;PUT "EXPECTED" IN R5
4697  027132  116104  000004                    MOVB    4(R1),R4      ;PUT "FOUND" IN R4
4698  027136  120504                            CMPB    R5,R4         ;DATA CORRECT?
4699  027140  001401                            BEQ     4$            ;BR IF YES
4700  027142  104015                            HLT     15            ;ALU ERROR
4701  027144  104401                      4$:   SCOP1                 ;SW09=1?
4702  027146  005202                            INC     R2            ;NEXT DATA
4703  027150  005200                            INC     R0            ;NEXT ADDRESS
4704  027152  022700  000010                    CMP     #10,R0        ;DONE YET?
4705  027156  001342                            BNE     1$            ;BR IF NO
4706  027160  104400                            SCOPE                 ;SCOPE THIS TEST
4707  027162     001     001     000      5$:   .BYTE   1,1,0,0,126,126,253,253
4708  027165     000     126     126
4709  027170     253     253
4710                                      .EVEN
4711
4712
4713                                            ;************************** TEST 111 **************************
4714                                            ;*ALU TEST
4715                                            ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED
4716                                            ;*ALU FUNCTION (A PLUS A)    CODE=5
4717                                            ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4718                                            ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4719                                            ;:************************************************************
4720
4721                                            ;  TEST 111
4722                                            ;--------------
4723  027172  012737  000111  001226    TST111: MOV     #111,TSTNO
4724  027200  012737  027346  001216            MOV     #TST112,NEXT
4725  027206  012737  027240  001220            MOV     #1$,LOCK
4726
4727  027214  104412                            MSTCLR                ;MASTER CLEAR DMC11
4728  027216  005000                            CLR     R0            ;MEM ← SP ADDRESS
4729  027220  012702  027336                    MOV     #5$,R2        ;POINTER TO CORRECT DATA
4730  027224  004737  034664                    JSR     PC,MEMLD      ;LOAD 8 WORDS OF MAIN MEMORY
4731  027230  035010                            MEMDAT                ;POINTER TO DATA
4732  027232  004737  034720                    JSR     PC,SPLD       ;LOAD 8 WORDS OF SP
4733  027236  035020                            SPDAT                 ;POINTER TO DATA
4734  027240  004737  034764            1$:     JSR     PC,CLRC       ;CLEAR C BIT!
4735  027244  042737  000017  027260            BIC     #17,2$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
4736  027252  050037  027260                    BIS     R0,2$         ;ADD ADDRESS TO INSTRUCTION
4737  027256  104414                            ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4738  027260  010000                      2$:   010000                ;LOAD MAR
4739  027262  042737  000017  027276            BIC     #17,3$        ;CLEAR ADDRESS OF INSTRUCTION
4740  027270  050037  027276                    BIS     R0,3$         ;ADD ADDRESS TO INSTRUCTION
4741  027274  104414                            ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4742  027276  040520                      3$:   040400!<5*20>         ;BR ← 2A
4743  027300  104414                            ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4744  027302  061224                            61224                 ;MOVE BR TO PORT4
```

```
 4745  027304  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4746  027306  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4747  027312  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4748  027314  001401                          BEQ     4$              ;BR IF YES
 4749  027316  104015                          HLT     15              ;ALU ERROR
 4750  027320  104401                  4$:     SCOP1                   ;SW09=1?
 4751  027322  005202                          INC     R2              ;NEXT DATA
 4752  027324  005200                          INC     R0              ;NEXT ADDRESS
 4753  027326  022700  000010                  CMP     #10,R0          ;DONE YET?
 4754  027332  001342                          BNE     1$              ;BR IF NO
 4755  027334  104400                          SCOPE                   ;SCOPE THIS TEST
 4756  027336     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
 4757  027341     376     252     252
 4758  027344     124     124
 4759                                  .EVEN
 4760
 4761
 4762                                          ;*************************** TEST 112 ****************************
 4763                                          ;*ALU TEST
 4764                                          ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED
 4765                                          ;*ALU FUNCTION (A PLUS C)    CODE=4
 4766                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 4767                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 4768                                          ;:*************************************************************
 4769
 4770                                          ;  TEST 112
 4771                                          ;-------------
 4772  027346  012737  000112  001226  TST112: MOV     #112,TSTNO
 4773  027354  012737  027522  001216          MOV     #TST113,NEXT
 4774  027362  012737  027414  001220          MOV     #1$,LOCK
 4775                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
 4776  027370  104412                          MSTCLR                  ;MASTER CLEAR DMC11
 4777  027372  005000                          CLR     R0              ;MEM + SP ADDRESS
 4778  027374  012702  027512                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 4779  027400  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 4780  027404  035010                          MEMDAT                  ;POINTER TO DATA
 4781  027406  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 4782  027412  035020                          SPDAT                   ;POINTER TO DATA
 4783  027414  004737  034764          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 4784  027420  042737  000017  027434          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4785  027426  050037  027434                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
 4786  027432  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4787  027434  010000                  2$:     010000                  ;LOAD MAR
 4788  027436  042737  000017  027452          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 4789  027444  050037  027452                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
 4790  027450  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4791  027452  040500                  3$:     040400!<4*20>           ;BR ← A PLUS C
 4792  027454  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4793  027456  061224                          61224                   ;MOVE BR TO PORT4
 4794  027460  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4795  027462  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4796  027466  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4797  027470  001401                          BEQ     4$              ;BR IF YES
 4798  027472  104015                          HLT     15              ;ALU ERROR
 4799  027474  104401                  4$:     SCOP1                   ;SW09=1?
 4800  027476  005202                          INC     R2              ;NEXT DATA
```

```
 4801  027500  005200                          INC     R0              ;NEXT ADDRESS
 4802  027502  022700  000010                  CMP     #10,R0          ;DONE YET?
 4803  027506  001342                          BNE     1$              ;BR IF NO
 4804  027510  104400                          SCOPE                   ;SCOPE THIS TEST
 4805  027512     000     000     377    5$:   .BYTE   0,0,-1,-1,125,125,252,252
 4806  027515     377     125     125
 4807  027520     252     252
 4808                                  .EVEN
 4809
 4810
 4811                                          ;************************** TEST 113 **************************
 4812                                          ;*ALU TEST
 4813                                          ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED
 4814                                          ;*ALU FUNCTION (A-B-1)    CODE=17
 4815                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 4816                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 4817                                          ;;**********************************************************
 4818
 4819                                          ;  TEST 113
 4820                                          ;--------------
 4821  027522  012737  000113  001226  TST113: MOV     #113,TSTNO
 4822  027530  012737  027676  001216          MOV     #TST114,NEXT
 4823  027536  012737  027570  001220          MOV     #1$,LOCK
 4824                                                                  ;P1 CONTAINS BASE DMC11 ADDRESS
 4825  027544  104412                          MSTCLR                  ;MASTER CLEAR DMC11
 4826  027546  005000                          CLR     R0              ;MEM + SP ADDRESS
 4827  027550  012702  027666                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
 4828  027554  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
 4829  027560  035010                          MEMDAT                  ;POINTER TO DATA
 4830  027562  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
 4831  027566  035020                          SPDAT                   ;POINTER TO DATA
 4832  027570  004737  034764          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
 4833  027574  042737  000017  027610          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4834  027602  050037  027610                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
 4835  027606  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4836  027610  010000                  2$:     010000                  ;LOAD MAR
 4837  027612  042737  000017  027626          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
 4838  027620  050037  027626                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
 4839  027624  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4840  027626  040760                  3$:     040400!<17*20>          ;BR + 2'S COMP SUB
 4841  027630  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4842  027632  061224                          61224                   ;MOVE BR TO PORT4
 4843  027634  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
 4844  027636  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
 4845  027642  120504                          CMPB    R5,R4           ;DATA CORRECT?
 4846  027644  001401                          BEQ     4$              ;BR IF YES
 4847  027646  104015                          HLT     15              ;ALU ERROR
 4848  027650  104401                  4$:     SCOP1                   ;SW09=1?
 4849  027652  005202                          INC     R2              ;NEXT DATA
 4850  027654  005200                          INC     R0              ;NEXT ADDRESS
 4851  027656  022700  000010                  CMP     #10,R0          ;DONE YET?
 4852  027662  001342                          BNE     1$              ;BR IF NO
 4853  027664  104400                          SCOPE                   ;SCOPE THIS TEST
 4854  027666     377     000     376    5$:   .BYTE   -1,0,376,-1,-1,252,124,-1
 4855  027671     377     377     252
 4856  027674     124     377
```

```
                                        .EVEN


                                        ;*************************** TEST 114 ****************************
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED
                                        ;*ALU FUNCTION (A-1)    CODE=7
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                        ;:*************************************************************

                                        ;  TEST 114
                                        ;-------------
027676  012737  000114  001226  TST114: MOV     #114,TSTNO
027704  012737  030052  001216          MOV     #TST115,NEXT
027712  012737  027744  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
027720  104412                          MSTCLR                  ;MASTER CLEAR DMC11
027722  005000                          CLR     R0              ;MEM + SP ADDRESS
027724  012702  030042                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
027730  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
027734  035010                          MEMDAT                  ;POINTER TO DATA
027736  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
027742  035020                          SPDAT                   ;POINTER TO DATA
027744  004737  034764          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
027750  042737  000017  027764          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
027756  050037  027764                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
027762  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027764  010000                  2$:     010000                  ;LOAD MAR
027766  042737  000017  030002          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
027774  050037  030002                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
030000  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030002  040560                  3$:     040400!<7*20>           ;BR = DEC A
030004  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030006  061224                          61224                   ;MOVE BR TO PORT4
030010  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
030012  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
030016  120504                          CMPB    R5,R4           ;DATA CORRECT?
030020  001401                          BEQ     4$              ;BR IF YES
030022  104015                          HLT     15              ;ALU ERROR
030024  104401                  4$:     SCOP1                   ;SW09=1?
030026  005202                          INC     R2              ;NEXT DATA
030030  005200                          INC     R0              ;NEXT ADDRESS
030032  022700  000010                  CMP     #10,R0          ;DONE YET?
030036  001342                          BNE     1$              ;BR IF NO
030040  104400                          SCOPE                   ;SCOPE THIS TEST
030042     377     377     376  5$:     .BYTE   -1,-1,376,376,124,124,251,251
030045     376     124     124
030050     251     251
                                        .EVEN


                                        ;*************************** TEST 115 ****************************
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION SEL B WITH C BIT SET
                                        ;*ALU FUNCTION (B)    CODE=11
```

```
4913                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4914                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4915                                                    ;:**********************************************************************
4916
4917                                                    ;  TEST 115
4918                                                    ;--------------
4919  030052  012737  000115  001226        TST115:   MOV     #115,TSTNO                    ;
4920  030060  012737  030226  001216                  MOV     #TST116,NEXT                  ;
4921  030066  012737  030120  001220                  MOV     #1$,LOCK                      ;
4922                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
4923  030074  104412                                 MSTCLR                                ;MASTER CLEAR DMC11
4924  030076  005000                                 CLR     R0                            ;MEM + SP ADDRESS
4925  030100  012702  030216                        MOV     #5$,R2                        ;POINTER TO CORRECT DATA
4926  030104  004737  034664                        JSR     PC,MEMLD                      ;LOAD 8 WORDS OF MAIN MEMORY
4927  030110  035010                                 MEMDAT                                ;POINTER TO DATA
4928  030112  004737  034720                        JSR     PC,SPLD                       ;LOAD 8 WORDS OF SP
4929  030116  035020                                 SPDAT                                 ;POINTER TO DATA
4930  030120  004737  034776        1$:             JSR     PC,SETC                       ;SET C BIT!
4931  030124  042737  000017  030140                BIC     #17,2$                        ;CLEAR ADDRESS FIELD OF INSTRUCTION
4932  030132  050037  030140                        BIS     R0,2$                         ;ADD ADDRESS TO INSTRUCTION
4933  030136  104414                                 ROMCLK                                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4934  030140  010000                        2$:     010000                                ;LOAD MAR
4935  030142  042737  000017  030156                BIC     #17,3$                        ;CLEAR ADDRESS OF INSTRUCTION
4936  030150  050037  030156                        BIS     R0,3$                         ;ADD ADDRESS TO INSTRUCTION
4937  030154  104414                                 ROMCLK                                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4938  030156  040620                        3$:     040400!<11*20>                        ;BR + SEL B
4939  030160  104414                                 ROMCLK                                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4940  030162  061224                                 61224                                 ;MOVE BR TO PORT4
4941  030164  111205                                 MOVB    (R2),R5                       ;PUT "EXPECTED" IN R5
4942  030166  116104  000004                        MOVB    4(R1),R4                      ;PUT "FOUND" IN R4
4943  030172  120504                                 CMPB    R5,R4                         ;DATA CORRECT?
4944  030174  001401                                 BEQ     4$                            ;BR IF YES
4945  030176  104015                                 HLT     15                            ;ALU ERROR
4946  030200  104401                        4$:     SCOP1                                 ;SW09=1?
4947  030202  005202                                 INC     R2                            ;NEXT DATA
4948  030204  005200                                 INC     R0                            ;NEXT ADDRESS
4949  030206  022700  000010                        CMP     #10,R0                        ;DONE YET?
4950  030212  001342                                 BNE     1$                            ;BR IF NO
4951  030214  104400                                 SCOPE                                 ;SCOPE THIS TEST
4952  030216     000     377     000        5$:     .BYTE   0,-1,0,-1,125,252,125,252
4953  030221     377     125     252
4954  030224     125     252
4955                                            .EVEN
4956
4957                                                    ;************************** TEST 116 **************************
4958                                                    ;*ALU TEST
4959                                                    ;*TEST OF ALU FUNCTION SEL A WITH C BIT SET
4960                                                    ;*ALU FUNCTION (A)    CODE=10
4961                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
4962                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
4963                                                    ;:**********************************************************************
4964
4965                                                    ;  TEST 116
4966                                                    ;--------------
4967
4968  030226  012737  000116  001226        TST116:   MOV     #116,TSTNO
```

```
 4969  030234  012737  030402  001216                MOV     #TST117,NEXT
 4970  030242  012737  030274  001220                MOV     #1$,LOCK
 4971
 4972  030250  104412                                MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
 4973  030252  005000                                CLR     R0              ;MASTER CLEAR DMC11
 4974  030254  012702  030372                        MOV     #5$,R2          ;MEM + SP ADDRESS
 4975  030260  004737  034664                        JSR     PC,MEMLD        ;POINTER TO CORRECT DATA
 4976  030264  035010                                MEMDAT                  ;LOAD 8 WORDS OF MAIN MEMORY
 4977  030266  004737  034720                        JSR     PC,SPLD         ;POINTER TO DATA
 4978  030272  035020                                SPDAT                   ;LOAD 8 WORDS OF SP
 4979  030274  004737  034776              1$:       JSR     PC,SETC         ;POINTER TO DATA
 4980  030300  042737  000017  030314                BIC     #17,2$          ;SET C BIT!
 4981  030306  050037  030314                        BIS     R0,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
 4982  030312  104414                                ROMCLK                  ;ADD ADDRESS TO INSTRUCTION
 4983  030314  010000                        2$:     010000                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4984  030316  042737  000017  030332                BIC     #17,3$          ;LOAD MAR
 4985  030324  050037  030332                        BIS     R0,3$           ;CLEAR ADDRESS OF INSTRUCTION
 4986  030330  104414                                ROMCLK                  ;ADD ADDRESS TO INSTRUCTION
 4987  030332  040600                        3$:     040400!<10*20>          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4988  030334  104414                                ROMCLK                  ;BR + SEL A
 4989  030336  061224                                61224                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 4990  030340  111205                                MOVB    (R2),R5         ;MOVE BR TO PORT4
 4991  030342  116104  000004                        MOVB    4(R1),R4        ;PUT "EXPECTED" IN R5
 4992  030346  120504                                CMPB    R5,R4           ;PUT "FOUND" IN R4
 4993  030350  001401                                BEQ     4$              ;DATA CORRECT?
 4994  030352  104015                                HLT     15              ;BR IF YES
 4995  030354  104401                        4$:     SCOP1                   ;ALU ERROR
 4996  030356  005202                                INC     R2              ;SW09=1?
 4997  030360  005200                                INC     R0              ;NEXT DATA
 4998  030362  022700  000010                        CMP     #10,R0          ;NEXT ADDRESS
 4999  030366  001342                                BNE     1$              ;DONE YET?
 5000  030370  104400                                SCOPE                   ;BR IF NO
 5001  030372     000     000     377        5$:     .BYTE   0,0,-1,-1,125,125,252,252  ;SCOPE THIS TEST
 5002  030375     377     125     125
 5003  030400     252     252
 5004                                       .EVEN
 5005
 5006
 5007                                               ;*************************** TEST 117 ****************************
 5008                                               ;*ALU TEST
 5009                                               ;*TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET
 5010                                               ;*ALU FUNCTION (A OR NOTB)    CODE=12
 5011                                               ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
 5012                                               ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
 5013                                               ;;*************************************************************
 5014
 5015                                               ;  TEST 117
 5016                                               ;--------------
 5017  030402  012737  000117  001226        TST117: MOV     #117,TSTNO
 5018  030410  012737  030556  001216                MOV     #TST120,NEXT
 5019  030416  012737  030450  001220                MOV     #1$,LOCK
 5020                                                                       ;R1 CONTAINS BASE DMC11 ADDRESS
 5021  030424  104412                                MSTCLR                  ;MASTER CLEAR DMC11
 5022  030426  005000                                CLR     R0              ;MEM + SP ADDRESS
 5023  030430  012702  030546                        MOV     #5$,R2          ;POINTER TO CORRECT DATA
 5024  030434  004737  034664                        JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
5025  030440  035010                                  MEMDAT                  ;POINTER TO DATA
5026  030442  004737  034720                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5027  030446  035020                                  SPDAT                   ;POINTER TO DATA
5028  030450  004737  034776                  1$:     JSR     PC,SETC         ;SET C BIT!
5029  030454  042737  000017  030470                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5030  030462  050037  030470                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5031  030466  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5032  030470  010000                          2$:     010000                  ;LOAD MAR
5033  030472  042737  000017  030506                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5034  030500  050037  030506                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5035  030504  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5036  030506  040640                          3$:     040400!<12*20>          ;BR <- A OR NOTB
5037  030510  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5038  030512  061224                                  61224                   ;MOVE BR TO PORT4
5039  030514  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5040  030516  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5041  030522  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5042  030524  001401                                  BEQ     4$              ;BR IF YES
5043  030526  104015                                  HLT     15              ;ALU ERROR
5044  030530  104401                          4$:     SCOP1                   ;SW09=1?
5045  030532  005202                                  INC     R2              ;NEXT DATA
5046  030534  005200                                  INC     R0              ;NEXT ADDRESS
5047  030536  022700  000010                          CMP     #10,R0          ;DONE YET?
5048  030542  001342                                  BNE     1$              ;BR IF NO
5049  030544  104400                                  SCOPE                   ;SCOPE THIS TEST
5050  030546     377     000     377          5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
5051  030551     377     377     125
5052  030554     252     377
5053                                          .EVEN
5054
5055
5056                                          ;************************** TEST 120 **************************
5057                                          ;*ALU TEST
5058                                          ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET
5059                                          ;*ALU FUNCTION (A AND B)    CODE=13
5060                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5061                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5062                                          ;:************************************************************
5063
5064                                          ;  TEST 120
5065                                          ;--------------
5066  030556  012737  000120  001226  TST120: MOV     #120,TSTNO
5067  030564  012737  030732  001216          MOV     #TST121,NEXT
5068  030572  012737  030624  001220          MOV     #1$,LOCK
5069                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5070  030600  104412                          MSTCLR                  ;MASTER CLEAR DMC11
5071  030602  005000                          CLR     R0              ;MEM + SP ADDRESS
5072  030604  012702  030722                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5073  030610  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5074  030614  035010                          MEMDAT                  ;POINTER TO DATA
5075  030616  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5076  030622  035020                          SPDAT                   ;POINTER TO DATA
5077  030624  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5078  030630  042737  000017  030644          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5079  030636  050037  030644                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5080  030642  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
5081  030644  010000                          2$:     010000                  ;LOAD MAR
5082  030646  042737  000017  030662                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5083  030654  050037  030662                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5084  030660  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5085  030662  040660                          3$:     040400!<13*20>          ;BR = A AND B
5086  030664  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5087  030666  061224                                  61224                   ;MOVE BR TO PORT4
5088  030670  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5089  030672  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5090  030676  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5091  030700  001401                                  BEQ     4$              ;BR IF YES
5092  030702  104015                                  HLT     15              ;ALU ERROR
5093  030704  104401                          4$:     SCOP1                   ;SW09=1?
5094  030706  005202                                  INC     R2              ;NEXT DATA
5095  030710  005200                                  INC     R0              ;NEXT ADDRESS
5096  030712  022700  000010                          CMP     #10,R0          ;DONE YET?
5097  030716  001342                                  BNE     1$              ;BR IF NO
5098  030720  104400                                  SCOPE                   ;SCOPE THIS TEST
5099  030722     000     000     000          5$:     .BYTE   0,0,0,-1,125,0,0,252
5100  030725     377     125     000
5101  030730     000     252
5102                                          .EVEN
5103
5104
5105                                          ;*************************** TEST 121 ****************************
5106                                          ;*ALU TEST
5107                                          ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET
5108                                          ;*ALU FUNCTION (A OR B)    CODE=14
5109                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5110                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5111                                          ;;*************************************************************
5112
5113                                          ;  TEST 121
5114                                          ;--------------
5115  030732  012737  000121  001226  TST121: MOV     #121,TSTNO
5116  030740  012737  031106  001216          MOV     #TST122,NEXT
5117  030746  012737  031000  001220          MOV     #1$,LOCK
5118                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
5119  030754  104412                          MSTCLR                         ;MASTER CLEAR DMC11
5120  030756  005000                          CLR     R0                     ;MEM + SP ADDRESS
5121  030760  012702  031076                  MOV     #5$,R2                 ;POINTER TO CORRECT DATA
5122  030764  004737  034664                  JSR     PC,MEMLD               ;LOAD 8 WORDS OF MAIN MEMORY
5123  030770  035010                          MEMDAT                         ;POINTER TO DATA
5124  030772  004737  034720                  JSR     PC,SPLD                ;LOAD 8 WORDS OF SP
5125  030776  035020                          SPDAT                          ;POINTER TO DATA
5126  031000  004737  034776          1$:     JSR     PC,SETC                ;SET C BIT!
5127  031004  042737  000017  031020          BIC     #17,2$                 ;CLEAR ADDRESS FIELD OF INSTRUCTION
5128  031012  050037  031020                  BIS     R0,2$                  ;ADD ADDRESS TO INSTRUCTION
5129  031016  104414                          ROMCLK                         ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5130  031020  010000                  2$:     010000                         ;LOAD MAR
5131  031022  042737  000017  031036          BIC     #17,3$                 ;CLEAR ADDRESS OF INSTRUCTION
5132  031030  050037  031036                  BIS     R0,3$                  ;ADD ADDRESS TO INSTRUCTION
5133  031034  104414                          ROMCLK                         ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5134  031036  040700                  3$:     040400!<14*20>                 ;BR = A OR B
5135  031040  104414                          ROMCLK                         ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5136  031042  061224                          61224                          ;MOVE BR TO PORT4
```

```
5137  031044  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5138  031046  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5139  031052  120504                          CMPB    R5,R4           ;DATA CORRECT?
5140  031054  001401                          BEQ     4$              ;BR IF YES
5141  031056  104015                          HLT     15              ;ALU ERROR
5142  031060  104401                  4$:     SCOP1                   ;SW09=1?
5143  031062  005202                          INC     R2              ;NEXT DATA
5144  031064  005200                          INC     R0              ;NEXT ADDRESS
5145  031066  022700  000010                  CMP     #10,R0          ;DONE YET?
5146  031072  001342                          BNE     1$              ;BR IF NO
5147  031074  104400                          SCOPE                   ;SCOPE THIS TEST
5148  031076     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
5149  031101     377     125     377
5150  031104     377     252
5151                                  .EVEN
5152
5153
5154                                          ;************************** TEST 122 **************************
5155                                          ;*ALU TEST
5156                                          ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET
5157                                          ;*ALU FUNCTION (A XOR B)    CODE=15
5158                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5159                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5160                                          ;:**********************************************************************
5161
5162                                          ;  TEST 122
5163                                          ;--------------
5164  031106  012737  000122  001226  TST122: MOV     #122,TSTNO
5165  031114  012737  031262  001216          MOV     #TST123,NEXT
5166  031122  012737  031154  001220          MOV     #1$,LOCK
5167                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5168  031130  104412                          MSTC_R                  ;MASTER CLEAR DMC11
5169  031132  005000                          CLR     R0              ;MEM + SP ADDRESS
5170  031134  012702  031252                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5171  031140  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5172  031144  035010                          MEMDAT                  ;POINTER TO DATA
5173  031146  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5174  031152  035020                          SPDAT                   ;POINTER TO DATA
5175  031154  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5176  031160  042737  000017  031174          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5177  031166  050037  031174                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5178  031172  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5179  031174  010000                  2$:     010000                  ;LOAD MAR
5180  031176  042737  000017  031212          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5181  031204  050037  031212                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5182  031210  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5183  031212  040720                  3$:     040400!<15*20>          ;BR + A XOR B
5184  031214  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5185  031216  061224                          61224                   ;MOVE BR TO PORT4
5186  031220  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5187  031222  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5188  031226  120504                          CMPB    R5,R4           ;DATA CORRECT?
5189  031230  001401                          BEQ     4$              ;BR IF YES
5190  031232  104015                          HLT     15              ;ALU ERROR
5191  031234  104401                  4$:     SCOP1                   ;SW09=1?
5192  031236  005202                          INC     R2              ;NEXT DATA
```

```
5193  031240  005200                                  INC     R0              ;NEXT ADDRESS
5194  031242  022700  000010                          CMP     #10,R0          ;DONE YET?
5195  031246  001342                                  BNE     1$              ;BR IF NO
5196  031250  104400                                  SCOPE                   ;SCOPE THIS TEST
5197  031252     000     377     377          5$:     .BYTE   0,-1,-1,0,0,-1,-1,0
5198  031255     000     000     377
5199  031260     377     000
5200                                          .EVEN
5201
5202
5203                                          ;*************************** TEST 123 ****************************
5204                                          ;*ALU TEST
5205                                          ;*TEST OF ALU FUNCTION ADD WITH C BIT SET
5206                                          ;*ALU FUNCTION (A PLUS B)    CODE=00
5207                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5208                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5209                                          ;:*************************************************************
5210
5211                                          ;  TEST 123
5212                                          ;-------------
5213  031262  012737  000123  001226  TST123: MOV     #123,TSTNO
5214  031270  012737  031436  001216          MOV     #TST124,NEXT
5215  031276  012737  031330  001220          MOV     #1$,LOCK
5216                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5217  031304  104412                          MSTCLR                  ;MASTER CLEAR DMC11
5218  031306  005000                          CLR     R0              ;MEM + SP ADDRESS
5219  031310  012702  031426                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5220  031314  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5221  031320  035010                          MEMDAT                  ;POINTER TO DATA
5222  031322  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5223  031326  035020                          SPDAT                   ;POINTER TO DATA
5224  031330  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5225  031334  042737  000017  031350          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5226  031342  050037  031350                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5227  031346  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5228  031350  010000                  2$:     010000                  ;LOAD MAR
5229  031352  042737  000017  031366          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5230  031360  050037  031366                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5231  031364  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5232  031366  040400                  3$:     040400!<00*20>          ;BR + ADD
5233  031370  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5234  031372  061224                          61224                   ;MOVE BR TO PORT4
5235  031374  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5236  031376  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5237  031402  120504                          CMPB    R5,R4           ;DATA CORRECT?
5238  031404  001401                          BEQ     4$              ;BR IF YES
5239  031406  104015                          HLT     15              ;ALU ERROR
5240  031410  104401                  4$:     SCOPI                   ;SW09=1?
5241  031412  005202                          INC     R2              ;NEXT DATA
5242  031414  005200                          INC     R0              ;NEXT ADDRESS
5243  031416  022700  000010                  CMP     #10,R0          ;DONE YET?
5244  031422  001342                          BNE     1$              ;BR IF NO
5245  031424  104400                          SCOPE                   ;SCOPE THIS TEST
5246  031426     000     377     377          5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
5247  031431     376     252     377
5248  031434     377     124
```

```
                                        .EVEN


                                        ;*************************** TEST 124 ****************************
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET
                                        ;*ALU FUNCTION (A PLUS A PLUS C)    CODE=6
                                        ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                        ;:*************************************************************

                                        ;  TEST 124
                                        ;--------------
031436  012737  000124  001226  TST124: MOV     #124,TSTNO
031444  012737  031612  001216         MOV     #TST125,NEXT
031452  012737  031504  001220         MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
031460  104412                        MSTCLR                  ;MASTER CLEAR DMC11
031462  005000                        CLR     R0              ;MEM + SP ADDRESS
031464  012702  031602                MOV     #5$,R2          ;POINTER TO CORRECT DATA
031470  004737  034664                JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031474  035010                        MEMDAT                  ;POINTER TO DATA
031476  004737  034720                JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031502  035020                        SPDAT                   ;POINTER TO DATA
031504  004737  034776         1$:    JSR     PC,SETC         ;SET C BIT!
031510  042737  000017  031524        BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031516  050037  031524                BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031522  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031524  010000                 2$:    010000                  ;LOAD MAR
031526  042737  000017  031542        BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031534  050037  031542                BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031540  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031542  040540                 3$:    040400!(6*20)           ;BR + 2A W/C
031544  104414                        ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031546  061224                        61224                   ;MOVE BR TO PORT4
031550  111205                        MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
031552  116104  000004                MOVB    4(R1),R4        ;PUT "FOUND" IN R4
031556  120504                        CMPB    R5,R4           ;DATA CORRECT?
031560  001401                        BEQ     4$              ;BR IF YES
031562  104015                        HLT     15              ;ALU ERROR
031564  104401                 4$:    SCOP1                   ;SW09=1?
031566  005202                        INC     R2              ;NEXT DATA
031570  005200                        INC     R0              ;NEXT ADDRESS
031572  022700  000010                CMP     #10,R0          ;DONE YET?
031576  001342                        BNE     1$              ;BR IF NO
031600  104400                        SCOPE                   ;SCOPE THIS TEST
031602     001     001     377  5$:    .BYTE   1,1,-1,-1,253,253,125,125
031605     377     253     253
031610     125     125
                                        .EVEN


                                        ;*************************** TEST 125 ****************************
                                        ;*ALU TEST
                                        ;*TEST OF ALU FUNCTION SUB WITH C BIT SET
                                        ;*ALU FUNCTION (A-B)    CODE=16
```

```
5305                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5306                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5307                                                    ;;*****************************************************************
5308
5309                                                    ;  TEST 125
5310                                                    ;-------------
5311  031612  012737  000125  001226        TST125:   MOV     #125,TSTNO
5312  031620  012737  031766  001216                  MOV     #TST126,NEXT
5313  031626  012737  031660  001220                  MOV     #1$,LOCK
5314                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
5315  031634  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
5316  031636  005000                                  CLR     R0              ;MEM + SP ADDRESS
5317  031640  012702  031756                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
5318  031644  004737  034664                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5319  031650  035010                                  MEMDAT                  ;POINTER TO DATA
5320  031652  004737  034720                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5321  031656  035020                                  SPDAT                   ;POINTER TO DATA
5322  031660  004737  034776        1$:       JSR     PC,SETC         ;SET C BIT!
5323  031664  042737  000017  031700                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5324  031672  050037  031700                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5325  031676  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5326  031700  010000                        2$:       010000                  ;LOAD MAR
5327  031702  042737  000017  031716                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5328  031710  050037  031716                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5329  031714  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5330  031716  040740                        3$:       040400!<16*20>          ;BR + SUB
5331  031720  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5332  031722  061224                                  61224                   ;MOVE BR TO PORT4
5333  031724  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5334  031726  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5335  031732  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5336  031734  001401                                  BEQ     4$              ;BR IF YES
5337  031736  104015                                  HLT     15              ;ALU ERROR
5338  031740  104401                        4$:       SCOP1                   ;SW09=1?
5339  031742  005202                                  INC     R2              ;NEXT DATA
5340  031744  005200                                  INC     R0              ;NEXT ADDRESS
5341  031746  022700  000010                          CMP     #10,R0          ;DONE YET?
5342  031752  001342                                  BNE     1$              ;BR IF NO
5343  031754  104400                                  SCOPE                   ;SCOPE THIS TEST
5344  031756     000     001     377        5$:       .BYTE   0,1,-1,0,0,253,125,0
5345  031761     000     000     253
5346  031764     125     000
5347                                          .EVEN
5348
5349
5350                                                    ;*********************** TEST 126 ****************************
5351                                                    ;*ALU TEST
5352                                                    ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET
5353                                                    ;*ALU FUNCTION (A PLUS B PLUS C)    CODE=01
5354                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5355                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5356                                                    ;;*****************************************************************
5357
5358                                                    ;  TEST 126
5359                                                    ;-------------
5360  031766  012737  000126  001226        TST126:   MOV     #126,TSTNO
```

```
5361  031774  012737  032142  001216                  MOV     #TST127,NEXT
5362  032002  012737  032034  001220                  MOV     #1$,LOCK
5363                                                  
5364  032010  104412                                  MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
5365  032012  005000                                  CLR     R0              ;MASTER CLEAR DMC11
5366  032014  012702  032132                          MOV     #5$,R2          ;MEM + SP ADDRESS
5367  032020  004737  034664                          JSR     PC,MEMLD        ;POINTER TO CORRECT DATA
5368  032024  035010                                  MEMDAT                  ;LOAD 8 WORDS OF MAIN MEMORY
5369  032026  004737  034720                          JSR     PC,SPLD         ;POINTER TO DATA
5370  032032  035020                                  SPDAT                   ;LOAD 8 WORDS OF SP
5371  032034  004737  034776          1$:             JSR     PC,SETC         ;SET C BIT!
5372  032040  042737  000017  032054                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5373  032046  050037  032054                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5374  032052  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5375  032054  010000                          2$:     010000                  ;LOAD MAR
5376  032056  042737  000017  032072                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5377  032064  050037  032072                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5378  032070  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5379  032072  040420                          3$:     040400!<01*20>          ;BR + ADD W/C
5380  032074  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5381  032076  061224                                  61224                   ;MOVE BR TO PORT4
5382  032100  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5383  032102  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5384  032106  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5385  032110  001401                                  BEQ     4$              ;BR IF YES
5386  032112  104015                                  HLT     15              ;ALU ERROR
5387  032114  104401                          4$:     SCOP1                   ;SW09=1?
5388  032116  005202                                  INC     R2              ;NEXT DATA
5389  032120  005200                                  INC     R0              ;NEXT ADDRESS
5390  032122  022700  000010                          CMP     #10,R0          ;DONE YET?
5391  032126  001342                                  BNE     1$              ;BR IF NO
5392  032130  104400                                  SCOPE                   ;SCOPE THIS TEST
5393  032132     001     000     000          5$:     .BYTE   1,0,0,-1,253,0,0,125
5394  032135     377     253     000
5395  032140     000     125
5396                                                  .EVEN
5397
5398                                                  ;*************************** TEST 127 ****************************
5399                                                  ;*ALU TEST
5400                                                  ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET
5401                                                  ;*ALU FUNCTION (A-B-C)    CODE=2
5402                                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5403                                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5404                                                  ;*****************************************************************
5405
5406
5407                                                  ;  TEST 127
5408                                                  ;-------------
5409  032142  012737  000127  001226          TST127: MOV     #127,TSTNO
5410  032150  012737  032316  001216                  MOV     #TST130,NEXT
5411  032156  012737  032210  001220                  MOV     #1$,LOCK
5412                                                  
5413  032164  104412                                  MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
5414  032166  005000                                  CLR     R0              ;MASTER CLEAR DMC11
5415  032170  012702  032306                          MOV     #5$,R2          ;MEM + SP ADDRESS
5416  032174  004737  034664                          JSR     PC,MEMLD        ;POINTER TO CORRECT DATA
                                                                              ;LOAD 8 WORDS OF MAIN MEMORY
```

```
5417  032200  035010                          MEMDAT                  ;POINTER TO DATA
5418  032202  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5419  032206  035020                          SPDAT                   ;POINTER TO DATA
5420  032210  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5421  032214  042737  000017  032230          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5422  032222  050037  032230                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5423  032226  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5424  032230  010000                  2$:     010000                  ;LOAD MAR
5425  032232  042737  000017  032246          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5426  032240  050037  032246                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5427  032244  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5428  032246  040440                  3$:     040400!(2*20)           ;BR + SUB W/C
5429  032250  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5430  032252  061224                          61224                   ;MOVE BR TO PORT4
5431  032254  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5432  032256  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5433  032262  120504                          CMPB    R5,R4           ;DATA CORRECT?
5434  032264  001401                          BEQ     4$              ;BR IF YES
5435  032266  104015                          HLT     15              ;ALU ERROR
5436  032270  104401                  4$:     SCOP1                   ;SW09=1?
5437  032272  005202                          INC     R2              ;NEXT DATA
5438  032274  005200                          INC     R0              ;NEXT ADDRESS
5439  032276  022700  000010                  CMP     #10,R0          ;DONE YET?
5440  032302  001342                          BNE     1$              ;BR IF NO
5441  032304  104400                          SCOPE                   ;SCOPE THIS TEST
5442  032306     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
5443  032311     000     000     253
5444  032314     125     000
5445                                  .EVEN
5446
5447
5448                                          ;*************************** TEST 130 ****************************
5449                                          ;*ALU TEST
5450                                          ;*TEST OF ALU FUNCTION INC A WITH C BIT SET
5451                                          ;*ALU FUNCTION (A PLUS 1)    CODE=3
5452                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5453                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5454                                          ;*****************************************************************
5455
5456                                          ;  TEST 130
5457                                          ;--------------
5458  032316  012737  000130  001226  TST130: MOV     #130,TSTNO
5459  032324  012737  032472  001216          MOV     #TST131,NEXT
5460  032332  012737  032364  001220          MOV     #1$,LOCK
5461                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5462  032340  104412                          MSTCLR                  ;MASTER CLEAR DMC11
5463  032342  005000                          CLR     R0              ;MEM + SP ADDRESS
5464  032344  012702  032462                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5465  032350  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5466  032354  035010                          MEMDAT                  ;POINTER TO DATA
5467  032356  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5468  032362  035020                          SPDAT                   ;POINTER TO DATA
5469  032364  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5470  032370  042737  000017  032404          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5471  032376  050037  032404                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5472  032402  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
5473  032404  010000                          2$:     010000                  ;LOAD MAR
5474  032406  042737  000017  032422                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5475  032414  050037  032422                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5476  032420  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5477  032422  040460                          3$:     040400!<3*20>           ;BR + INC A
5478  032424  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5479  032426  061224                                  61224                   ;MOVE BR TO PORT4
5480  032430  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5481  032432  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5482  032436  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5483  032440  001401                                  BEQ     4$              ;BR IF YES
5484  032442  104015                                  HLT     15              ;ALU ERROR
5485  032444  104401                          4$:     SCOP1                   ;SW09=1?
5486  032446  005202                                  INC     R2              ;NEXT DATA
5487  032450  005200                                  INC     R0              ;NEXT ADDRESS
5488  032452  022700  000010                          CMP     #10,R0          ;DONE YET?
5489  032456  001342                                  BNE     1$              ;BR IF NO
5490  032460  104400                                  SCOPE                   ;SCOPE THIS TEST
5491  032462     001     001     000          5$:     .BYTE   1,1,0,0,126,126,253,253
5492  032465     000     126     126
5493  032470     253     253
5494                                          .EVEN
5495
5496
5497                                                  ;*************************** TEST 131 ****************************
5498                                                  ;*ALU TEST
5499                                                  ;*TEST OF ALU FUNCTION 2A WITH C BIT SET
5500                                                  ;*ALU FUNCTION (A PLUS A)    CODE=5
5501                                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5502                                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5503                                                  ;;*****************************************************************
5504
5505                                                  ;  TEST 131
5506                                                  ;--------------
5507  032472  012737  000131  001226          TST131: MOV     #131,TSTNO
5508  032500  012737  032646  001216                  MOV     #TST132,NEXT
5509  032506  012737  032540  001220                  MOV     #1$,LOCK
5510                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
5511  032514  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
5512  032516  005000                                  CLR     R0              ;MEM + SP ADDRESS
5513  032520  012702  032636                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
5514  032524  004737  034664                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5515  032530  035010                                  MEMDAT                  ;POINTER TO DATA
5516  032532  004737  034720                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5517  032536  035020                                  SPDAT                   ;POINTER TO DATA
5518  032540  004737  034776                  1$:     JSR     PC,SETC         ;SET C BIT!
5519  032544  042737  000017  032560                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5520  032552  050037  032560                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5521  032556  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5522  032560  010000                          2$:     010000                  ;LOAD MAR
5523  032562  042737  000017  032576                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5524  032570  050037  032576                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5525  032574  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5526  032576  040520                          3$:     040400!<5*20>           ;BR + 2A
5527  032600  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5528  032602  061224                                  61224                   ;MOVE BR TO PORT4
```

```
5529  032604  111205                                  MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5530  032606  116104  000004                          MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5531  032612  120504                                  CMPB    R5,R4           ;DATA CORRECT?
5532  032614  001401                                  BEQ     4$              ;BR IF YES
5533  032616  104015                                  HLT     15              ;ALU ERROR
5534  032620  104401                          4$:     SCOP1                   ;SW09=1?
5535  032622  005202                                  INC     R2              ;NEXT DATA
5536  032624  005200                                  INC     R0              ;NEXT ADDRESS
5537  032626  022700  000010                          CMP     #10,R0          ;DONE YET?
5538  032632  001342                                  BNE     1$              ;BR IF NO
5539  032634  104400                                  SCOPE                   ;SCOPE THIS TEST
5540  032636     000     000     376          5$:     .BYTE   0,0,376,376,252,252,124,124
5541  032641     376     252     252
5542  032644     124     124
5543                                          .EVEN
5544
5545
5546                                          ;*************************** TEST 132 ****************************
5547                                          ;*ALU TEST
5548                                          ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET
5549                                          ;*ALU FUNCTION (A PLUS C)    CODE=4
5550                                          ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5551                                          ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5552                                          ;*****************************************************************
5553
5554                                          ;  TEST 132
5555                                          ;--------------
5556  032646  012737  000132  001226  TST132: MOV     #132,TSTNO
5557  032654  012737  033022  001216          MOV     #TST133,NEXT
5558  032662  012737  032714  001220          MOV     #1$,LOCK
5559                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5560  032670  104412                          MSTCLR                  ;MASTER CLEAR DMC11
5561  032672  005000                          CLR     R0              ;MEM + SP ADDRESS
5562  032674  012702  033012                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5563  032700  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5564  032704  035010                          MEMDAT                  ;POINTER TO DATA
5565  032706  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5566  032712  035020                          SPDAT                   ;POINTER TO DATA
5567  032714  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5568  032720  042737  000017  032734          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5569  032726  050037  032734                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5570  032732  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5571  032734  010000                  2$:     010000                  ;LOAD MAR
5572  032736  042737  000017  032752          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5573  032744  050037  032752                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5574  032750  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5575  032752  040500                  3$:     040400!<4*20>           ;BR ← A PLUS C
5576  032754  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5577  032756  061224                          61224                   ;MOVE BR TO PORT4
5578  032760  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5579  032762  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5580  032766  120504                          CMPB    R5,R4           ;DATA CORRECT?
5581  032770  001401                          BEQ     4$              ;BR IF YES
5582  032772  104015                          HLT     15              ;ALU ERROR
5583  032774  104401                  4$:     SCOP1                   ;SW09=1?
5584  032776  005202                          INC     R2              ;NEXT DATA
```

```
5585  033000  005200                          INC     R0              ;NEXT ADDRESS
5586  033002  022700  000010                  CMP     #10,R0          ;DONE YET?
5587  033006  001342                          BNE     1$              ;BR IF NO
5588  033010  104400                          SCOPE                   ;SCOPE THIS TEST
5589  033012     001     001     000  5$:     .BYTE   1,1,0,0,126,126,253,253
5590  033015     000     126     126
5591  033020     253     253
5592                                  .EVEN
5593
5594
5595                                  ;************************** TEST 133 **************************
5596                                  ;*ALU TEST
5597                                  ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET
5598                                  ;*ALU FUNCTION (A-B-1)    CODE=17
5599                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5600                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5601                                  ;:************************************************************
5602
5603                                  ;  TEST 133
5604                                  ;-------------
5605  033022  012737  000133  001226  TST133: MOV     #133,TSTNO
5606  033030  012737  033176  001216          MOV     #TST134,NEXT
5607  033036  012737  033070  001220          MOV     #1$,LOCK
5608                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
5609  033044  104412                          MSTCLR                  ;MASTER CLEAR DMC11
5610  033046  005000                          CLR     R0              ;MEM + SP ADDRESS
5611  033050  012702  033166                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
5612  033054  004737  034664                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
5613  033060  035010                          MEMDAT                  ;POINTER TO DATA
5614  033062  004737  034720                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
5615  033066  035020                          SPDAT                   ;POINTER TO DATA
5616  033070  004737  034776          1$:     JSR     PC,SETC         ;SET C BIT!
5617  033074  042737  000017  033110          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5618  033102  050037  033110                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
5619  033106  106414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5620  033110  010000                  2$:     010000                  ;LOAD MAR
5621  033112  042737  000017  033126          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
5622  033120  050037  033126                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
5623  033124  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5624  033126  040760                  3$:     040400!<17*20>          ;BR + 2'S COMP SUB
5625  033130  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5626  033132  061224                          61224                   ;MOVE BR TO PORT4
5627  033134  111205                          MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
5628  033136  116104  000004                  MOVB    4(R1),R4        ;PUT "FOUND" IN R4
5629  033142  120504                          CMPB    R5,R4           ;DATA CORRECT?
5630  033144  001401                          BEQ     4$              ;BR IF YES
5631  033146  104015                          HLT     15              ;ALU ERROR
5632  033150  104401                  4$:     SCOP1                   ;SW09=1?
5633  033152  005202                          INC     R2              ;NEXT DATA
5634  033154  005200                          INC     R0              ;NEXT ADDRESS
5635  033156  022700  000010                  CMP     #10,R0          ;DONE YET?
5636  033162  001342                          BNE     1$              ;BR IF NO
5637  033164  104400                          SCOPE                   ;SCOPE THIS TEST
5638  033166     377     000     376  5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
5639  033171     377     377     252
5640  033174     124     377
```

```
5641                                                    .EVEN
5642
5643
5644                                                    ;*************************** TEST 134 ****************************
5645                                                    ;*ALU TEST
5646                                                    ;*TEST OF ALU FUNCTION DEC A WITH C BIT SET
5647                                                    ;*ALU FUNCTION (A-1)    CODE=7
5648                                                    ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
5649                                                    ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
5650                                                    ;;*************************************************************
5651
5652                                                    ;  TEST 134
5653                                                    ;--------------
5654  033176  012737  000134  001226          TST134: MOV     #134,TSTNO
5655  033204  012737  033352  001216                  MOV     #TST135,NEXT
5656  033212  012737  033244  001220                  MOV     #1$,LOCK
5657                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
5658  033220  104412                                  MSTCLR              ;MASTER CLEAR DMC11
5659  033222  005000                                  CLR     R0          ;MEM + SP ADDRESS
5660  033224  012702  033342                          MOV     #5$,R2      ;POINTER TO CORRECT DATA
5661  033230  004737  034664                          JSR     PC,MEMLD    ;LOAD 8 WORDS OF MAIN MEMORY
5662  033234  035010                                  MEMDAT              ;POINTER TO DATA
5663  033236  004737  034720                          JSR     PC,SPLD     ;LOAD 8 WORDS OF SP
5664  033242  035020                                  SPDAT               ;POINTER TO DATA
5665  033244  004737  034776          1$:     JSR     PC,SETC     ;SET C BIT!
5666  033250  042737  000017  033264                  BIC     #17,2$      ;CLEAR ADDRESS FIELD OF INSTRUCTION
5667  033256  050037  033264                          BIS     R0,2$       ;ADD ADDRESS TO INSTRUCTION
5668  033262  104414                                  ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5669  033264  010000                          2$:     010000              ;LOAD MAR
5670  033266  042737  000017  033302                  BIC     #17,3$      ;CLEAR ADDRESS OF INSTRUCTION
5671  033274  050037  033302                          BIS     R0,3$       ;ADD ADDRESS TO INSTRUCTION
5672  033300  104414                                  ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5673  033302  040560                          3$:     040400!<7*20>       ;BR ← DEC A
5674  033304  104414                                  ROMCLK              ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5675  033306  061224                                  61224               ;MOVE BR TO PORT4
5676  033310  111205                                  MOVB    (R2),R5     ;PUT "EXPECTED" IN R5
5677  033312  116104  000004                          MOVB    4(R1),R4    ;PUT "FOUND" IN R4
5678  033316  120504                                  CMPB    R5,R4       ;DATA CORRECT?
5679  033320  001401                                  BEQ     4$          ;BR IF YES
5680  033322  104015                                  HLT     15          ;ALU ERROR
5681  033324  104401                          4$:     SCOP1               ;SW09=1?
5682  033326  005202                                  INC     R2          ;NEXT DATA
5683  033330  005200                                  INC     R0          ;NEXT ADDRESS
5684  033332  022700  000010                          CMP     #10,R0      ;DONE YET?
5685  033336  001342                                  BNE     1$          ;BR IF NO
5686  033340  104400                                  SCOPE               ;SCOPE THIS TEST
5687  033342     377     377     376          5$:     .BYTE   -1,-1,376,376,124,124,251,251
5688  033345     376     124     124
5689  033350     251     251
5690                                                    .EVEN
5691
5692
5693                                                    ;*************************** TEST 135 ****************************
5694                                                    ;*TEST OF PROGRAM CLOCK BIT
5695                                                    ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
5696                                                    ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
```

```
5697                                                    ;*AND THEN SETS SOME TIME LATER
5698                                                    ;;*****************************************************************
5699
5700                                                    ;  TEST 135
5701                                                    ;-------------
5702  033352  012737  000135  001226          TST135: MOV     #135,TSTNO
5703  033360  012737  033532  001216                  MOV     #TST136,NEXT
5704                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
5705  033366  104412                                   MSTCLR                        ;MASTER CLEAR DMC11
5706  033370  005037  001416                           CLR     TEMP                  ;PREPARE FOR
5707  033374  005037  001246                           CLR     TEMP1                 ;DELAY
5708  033400  012702  000011                           MOV     #11,R2                ;SAVE FOR TYPEOUT
5709  033404  012761  000020  000004          1$:     MOV     #20,4(R1)             ;LOAD PORT 4
5710  033412  152761  000002  000001                   BISB    #BIT1,1(R1)           ;SET ROMI
5711  033420  012761  121111  000006                   MOV     #121111,6(R1)         ;SEL6 = INSTRUCTION
5712  033426  152761  000003  000001                   BISB    #BIT1!BIT0,1(R1)      ;SET CLOCK BIT
5713  033434  012761  121224  000006                   MOV     #121224,6(R1)         ;LOAD NEXT INSTRUCTION
5714  033442  152761  000003  000001                   BISB    #BIT1!BIT0,1(R1)      ;READ CLOCK BIT
5715  033450  142761  000003  000001                   BICB    #BIT1!BIT0,1(R1)      ;CLEAR MAINT BITS
5716  033456  016104  000004                           MOV     4(R1),R4              ;PUT "FOUND" IN R4
5717  033462  005005                                   CLR     R5                    ;PUT "EXPECTED" IN R5
5718  033464  120504                                   CMPB    R5,R4                 ;IS PGM CLOCK CLEAR?
5719  033466  001401                                   BEQ     2$
5720  033470  104016                                   HLT     16                    ;ERROR, PGM CLOCK IS NOT CLEAR
5721  033472                                    2$:
5722  033472  104414                                   ROMCLK                        ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5723  033474  121224                                   121224                        ;PORT4=LUII
5724  033476  122761  000020  000004                   CMPB    #20,4(R1)             ;IS PGM CLOCK SET?
5725  033504  001411                                   BEQ     3$                    ;BR IF YES
5726  033506  005237  001416                           INC     TEMP                  ;INCREMENT DELAY
5727  033512  005537  001246                           ADC     TEMP1                 ;INCREMENT DELAY
5728  033516  022737  000006  001246                   CMP     #6,TEMP1              ;IS DELAY DONE
5729  033524  001362                                   BNE     2$                    ;BR IF NO
5730  033526  104016                                   HLT     16                    ;ERROR PGM CLOCK NOT SET
5731  033530  104400                            3$:     SCOPE                         ;SCOPE THIS TEST
5732
5733
5734                                                    ;*************************** TEST 136 ****************************
5735                                                    ;*FORCE POWER FAIL TEST
5736                                                    ;*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
5737                                                    ;*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
5738                                                    ;*BLOCKED FROM GETTING TO THE DMC DURING THE POWER FAIL
5739                                                    ;*THIS TEST MAY HANG ON SOME PROCESSORS IF AN M9301 IS PRESENT.
5740                                                    ;*TO AVOID HANGING SW02 (POWER ON REBOOT ENABLE) ON THE M9301
5741                                                    ;*MUST BE IN THE OFF POSITION. THIS TEST WILL ALSO FAIL IF THE
5742                                                    ;*CPU POWER FAIL VECTOR IS SET TO ANY LOCATION OTHER THAN 24.
5743                                                    ;*IF THIS TEST HANGS OR FAILS DUE TO EITHER REASON ABOVE THE
5744                                                    ;*FOLLOWING PATCH MAY BE INSTALLED TO SKIP THIS TEST:
5745                                                    ;*
5746                                                    ;*      LOC 33362       WAS 33532       SB 33724
5747                                                    ;;*****************************************************************
5748
5749                                                    ;  TEST 136
5750                                                    ;-------------
5751  033532  012737  000136  001226          TST136: MOV     #136,TSTNO
5752  033540  012737  033724  001216                  MOV     #TST137,NEXT
```

```
5753
5754  033546  104412                                  MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
5755  033550  005037  001416                          CLR     TEMP            ;MASTER CLEAR DMC11
5756  033554  013746  000024                          MOV     @#24,-(SP)      ;PREPARE FOR DELAY
5757  033560  012737  033624  000024                  MOV     #1$,@#24        ;STORE POWER FAIL ADDRESS
5758  033566  012761  000002  000004                  MOV     #2,4(R1)        ;SET UP FOR FORCE POWER FAIL
5759  033574  012711  001000                          MOV     #BIT9,(R1)      ;LOAD PORT4
5760  033600  012761  121111  000006                  MOV     #121111,6(R1)   ;SET ROMI
5761  033606  012711  001400                          MOV     #BIT9!BIT8,(R1) ;LOAD INSTRUCTION
5762  033612  005237  001416              5$:         INC     TEMP            ;CLOCK INSTRUCTION
5763  033616  001375                                  BNE     5$              ;WAIT FOR POWER FAIL
5764  033620  104017                                  HLT     17              ;BR IF DELAY NOT DONE
5765  033622  000426                                  BR      4$              ;ERROR, NO POWER FAIL
5766  033624  012737  033642  000024          1$:     MOV     #3$,@#24        ;POWER UP ADDRESS
5767  033632  010637  033640                          MOV     SP,2$           ;STORE STACK
5768  033636  000000                                  HALT                    ;WAIT FOR POWER UP SEQUENCE
5769  033640  000000                              2$: 0
5770  033642  013706  033640              3$:     MOV     2$,SP           ;RESTORE STACK
5771  033646  022626                                  POP2SP                  ;POP STACK TWICE
5772  033650  012637  000024                          MOV     (SP)+,@#24      ;RESTORE TRUE POWER FAIL ADDRESS
5773  033654  022737  005336  000024                  CMP     #.PFAIL,@#24    ;IS IT CORRECT?
5774  033662  001406                                  BEQ     4$              ;BR IF YES
5775  033664  104017                                  HLT     17              ;ERROR, STACK IS INCORRECT
5776  033666  012737  005336  000024                  MOV     #.PFAIL,@#24    ;RESTORE TRUE POWER FAIL ADDRESS
5777  033674  012706  001200                          MOV     #STACK,SP       ;RESTORE STACK
5778  033700  012711  003000              4$:         MOV     #BIT9!BIT10,(R1);SEL6 = MAINT IR
5779  033704  012705  121111                          MOV     #121111,R5      ;R5 = EXPECTED
5780  033710  016104  000004                          MOV     4(R1),R4        ;R4 = FOUND
5781  033714  020504                                  CMP     R5,R4           ;MAINT IR SHOULD = 12111
5782  033716  001401                                  BEQ     .+4             ;BR IF OK
5783  033720  104025                                  HLT     25              ;IF = 0 THEN BUS INIT WAS
5784                                                                          ;NOT BLOCKED FROM CLEARING
5785                                                                          ;THE DMC-11
5786  033722  104400                                  SCOPE                   ;SCOPE THIS TEST
5787
5788
5789                                                  ;*************************** TEST 137 ****************************
5790                                                  ;*MICRO-PROCESSOR NOISE TEST
5791                                                  ;*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
5792                                                  ;*TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
5793                                                  ;*THEN GO BACK AND READ THE DATA PATERNS  TO VERIFY THAT
5794                                                  ;*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
5795                                                  ;*DID NOT CHANGE THE DATA.
5796                                                  ;:*************************************************************
5797
5798                                                  ;  TEST 137
5799                                                  ;--------------
5800  033724  012737  000137  001226      TST137:     MOV     #137,TSTNO
5801  033732  012737  003364  001216                  MOV     #.EOP,NEXT
5802                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
5803  033740  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
5804  033742  005002                                  CLR     R2              ;R2 IS INDEX REGISTER
5805  033744  042737  000017  033770          1$:     BIC     #17,2$          ;CLEAR ADDRESS FIELD
5806  033752  156237  034564  033770                  BISB    30$(R2),2$      ;ADD IBUS* REG ADDRESS TO INSTRUCTION
5807  033760  116261  034572  000004                  MOVB    31$(R2),4(R1)   ;LOAD PORT4
5808  033766  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
5809  033770  121100                    2$:   121100                ;WRITE IBUS* REGISTER
5810  033772  005202                          INC     R2            ;INC INDEX REGISTER
5811  033774  022702  000005                CMP     #5,R2         ;DONE YET?
5812  034000  001361                        BNE     1$            ;BR IF NO
5813  034002  005002                        CLR     R2            ;R2 IS IBUS REGISTER ADDRESS
5814  034004  042737  000017  034050  3$:   BIC     #17,4$        ;CLEAR ADDRESS FIELD OF INSTRUCTIONS
5815  034012  042737  000017  034062        BIC     #17,5$
5816  034020  042737  000017  034072        BIC     #17,6$
5817  034026  050237  034050                BIS     R2,4$         ;ADD IBUS REG ADDRESS TO INSTRUCTION
5818  034032  050237  034062                BIS     R2,5$
5819  034036  050237  034072                BIS     R2,6$
5820  034042  105061  000004                CLRB    4(R1)         ;CLEAR PORT4
5821  034046  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5822  034050  122100                    4$:   122100                ;WRITE 0 TO IBUS REG
5823  034052  112761  000377  000004        MOVB    #377,4(R1)    ;LOAD PORT4
5824  034060  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5825  034062  122100                    5$:   122100                ;WRITE ALL ONES TO IBUS REG
5826  034064  110261  000004                MOVB    R2,4(R1)      ;LOAD PORT4
5827  034070  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5828  034072  122100                    6$:   122100                ;WRITE ITS OWN ADDRESS TO IBUS REG
5829  034074  005202                        INC     R2            ;NEXT ADDRESS
5830  034076  022702  000010                CMP     #10,R2        ;DONE YET?
5831  034102  001340                        BNE     3$            ;BR IF NO
5832  034104  005002                        CLR     R2            ;START AT SP ADDRESS 0
5833  034106  042737  000017  034152  7$:   BIC     #17,8$        ;CLEAR ADDRESS FIELD
5834  034114  042737  000017  034164        BIC     #17,9$
5835  034122  042737  000017  034174        BIC     #17,10$
5836  034130  050237  034152                BIS     R2,8$         ;ADD ADDRESS TO INSTRUCTION
5837  034134  050237  034164                BIS     R2,9$
5838  034140  050237  034174                BIS     R2,10$
5839  034144  105061  000004                CLRB    4(R1)         ;CLEAR PORT4
5840  034150  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5841  034152  123100                    8$:   123100                ;WRITE ZERO TO SP
5842  034154  112761  000377  000004        MOVB    #377,4(R1)    ;LOAD PORT4
5843  034162  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5844  034164  123100                    9$:   123100                ;WRITE ALL ONES TO SP
5845  034166  110261  000004                MOVB    R2,4(R1)      ;LOAD PORT4
5846  034172  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5847  034174  123100                   10$:   123100                ;WRITE SP ADDRESS TO ITSELF
5848  034176  005202                        INC     R2            ;NEXT SP ADDRESS
5849  034200  022702  000020                CMP     #20,R2        ;DONE YET?
5850  034204  001340                        BNE     7$            ;BR IF NO
5851  034206  005002                        CLR     R2            ;R2 = MAIN MEM ADDRESS
5852  034210  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5853  034212  010000                        010000                ;MAR ← 0
5854  034214  105061  000004               11$:   CLRB    4(R1)         ;CLEAR PORT4
5855  034220  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5856  034222  122500                        122500                ;WRITE ZEROS TO MEM
5857  034224  112761  000377  000004        MOVB    #377,4(R1)    ;LOAD PORT4
5858  034232  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5859  034234  122500                        122500                ;WRITE ONES TO MEM
5860  034236  110261  000004                MOVB    R2,4(R1)      ;LOAD PORT4
5861  034242  104414                        ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5862  034244  136500                        136500                ;WRITE TO MEM IT OWN ADDRESS
5863  034246  005202                        INC     R2            ;NEXT MEM ADDRESS
5864  034250  022702  000400                CMP     #400,R2       ;DONE YET?
```

```
5865  034254  001357                                    BNE     11$             ;BR IF NO
5866
5867                                                    ;NOW GO BACK AND READ EVERYTHING
5868
5869  034256  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5870  034260  010000                                    010000                  ;MAR←0
5871  034262  032737  100000  001366                    BIT     #BIT15,STAT1    ;DMC?
5872  034270  001402                                    BEQ     .+6             ;BR IF YES
5873  034272  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5874  034274  004000                                    4000                    ;MAR HI ← 0 (KMC ONLY)
5875  034276  005000                                    CLR     R0              ;R0 IS INDEX REGISTER
5876  034300  042737  000360  034336          12$:     BIC     #360,13$        ;CLEAR ADDRESS FIELD
5877  034306  116002  034564                          MOVB    30$(R0),R2      ;R2 = IBUS# ADDRESS
5878  034312  010203                                    MOV     R2,R3           ;PUT IBUS# ADDRESS IN R3
5879  034314  006303                                    ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
5880  034316  006303                                    ASL     R3
5881  034320  006303                                    ASL     R3
5882  034322  006303                                    ASL     R3
5883  034324  050337  034336                          BIS     R3,13$          ;ADD ADDRESS TO INSTRUCTION
5884  034330  116005  034572                          MOVB    31$(R0),R5      ;R5 = "EXPECTED"
5885  034334  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5886  034336  121004                          13$:     121004                  ;PORT4 ← IBUS# REGISTER
5887  034340  016104  000004                          MOV     4(R1),R4        ;R4 = "FOUND"
5888  034344  120504                                    CMPB    R5,R4           ;IBUS# CONTENTS OK?
5889  034346  001401                                    BEQ     .+4             ;BR IF YES
5890  034350  104004                                    HLT     4               ;IBUS# DATA ERROR
5891  034352  005200                                    INC     R0              ;INC COUNTER
5892  034354  022700  000005                          CMP     #5,R0           ;DONE YET?
5893  034360  001347                                    BNE     12$             ;BR IF NO
5894  034362  005002                                    CLR     R2              ;R2 = IBUS REG ADDRESS
5895  034364  042737  000360  034414          14$:     BIC     #360,15$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
5896  034372  010203                                    MOV     R2,R3           ;R3 = IBUS ADDRESS
5897  034374  006303                                    ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
5898  034376  006303                                    ASL     R3
5899  034400  006303                                    ASL     R3
5900  034402  006303                                    ASL     R3
5901  034404  050337  034414                          BIS     R3,15$          ;ADD ADDRESS TO INSTUCTION
5902  034410  010205                                    MOV     R2,R5           ;R5 = "EXPECTED"
5903  034412  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5904  034414  021004                          15$:     021004                  ;PORT4 ← IBUS REG
5905  034416  016104  000004                          MOV     4(R1),R4        ;R4 = "FOUND"
5906  034422  120504                                    CMPB    R5,R4           ;IBUS CONTENTS OK?
5907  034424  001401                                    BEQ     .+4             ;BR IF YES
5908  034426  104005                                    HLT     5               ;IBUS DATA ERROR
5909  034430  005202                                    INC     R2              ;NEXT IBUS REGISTER
5910  034432  022702  000010                          CMP     #10,R2          ;DONE YET?
5911  034436  001352                                    BNE     14$             ;BR IF NO
5912  034440  005002                                    CLR     R2              ;R2 = SP ADDRESS
5913  034442  042737  000017  034456          16$:     BIC     #17,17$ ;CLEAR ADRESS FIELD OF INSTRUCTION
5914  034450  050237  034456                          BIS     R2,17$          ;ADD ADDRESS TO INSTRUCTION
5915  034454  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5916  034456  040600                          17$:     040600                  ;BR ← SP
5917  034460  010205                                    MOV     R2,R5           ;R5 = "EXPECTED"
5918  034462  104414                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5919  034464  061224                                    061224                  ;PORT4 ← BR
5920  034466  016104  000004                          MOV     4(R1),R4        ;R4 = "FOUND"
```

```
5921  034472  120504                                  CMPB    R5,R4           ;SP CONTENTS OK?
5922  034474  001401                                  BEQ     .+4             ;BR IF YES
5923  034476  104007                                  HLT     7               ;SP DATA ERROR
5924  034500  005202                                  INC     R2              ;NEXT SP LOCATION
5925  034502  022702  000020                          CMP     #20,R2          ;DONE YET?
5926  034506  001355                                  BNE     16$             ;BR IF NO
5927  034510  005002                                  CLR     R2              ;R2 = MEMORY ADDRESS
5928  034512  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5929  034514  010000                                  010000                  ;MAR ← 0
5930  034516  032737  100000  001366                  BIT     #BIT15,STAT1    ;DMC?
5931  034524  001402                                  BEQ     .+6             ;BR IF YES
5932  034526  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5933  034530  004000                                  4000                    ;MAR HI ← 0 (KMC ONLY)
5934  034532  010205                          18$:    MOV     R2,R5           ;R5 = "EXPECTED"
5935  034534  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5936  034536  055224                                  055224                  ;PORT4 ← MAIN MEM
5937  034540  016104  000004                          MOV     4(R1),R4        ;R4 = "FOUND"
5938  034544  120504                                  CMPB    R5,R4           ;MAIN MEM CONTENTS OK?
5939  034546  001401                                  BEQ     .+4             ;BR IF YES
5940  034550  104013                                  HLT     13              ;MAIN MEM DATA ERROR
5941  034552  005202                                  INC     R2              ;NEXT MEM ADDRESS
5942  034554  022702  000400                          CMP     #400,R2         ;DONE YET?
5943  034560  001364                                  BNE     18$             ;BR IF NO
5944  034562  104400                                  SCOPE                   ;SCOPE THIS TEST
5945  034564     000     002     003          30$:    .BYTE   0,2,3,5,10
5946  034567     005     010
5947          034572                                  .EVEN
5948  034572     001     003     004          31$:    .BYTE   1,3,4,6,10
5949  034575     006     010
5950          034600                                  .EVEN
5951                                          00200
5952                                          00300
5953                                          00400   ;SUBROUTINES
5954                                          00500   ;-----------
5955                                          00600
5956  034600                                  00700   SETVEC:
5957                                          00800           ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS
5958                                          00900
5959  034600  012577  144570                  01000           MOV     (R5)+,@DMRVEC   ;LOAD BASE VECTOR
5960  034604  012577  144570                  01100           MOV     (R5)+,@DMTVEC   ;LOAD VECTOR + 2
5961  034610  112577  144562                  01200           MOVB    (R5)+,@DMRLVL   ;LOAD VECTOR + 4
5962  034614  112577  144562                  01300           MOVB    (R5)+,@DMTLVL   ;LOAD VECTOR + 6
5963  034620  000205                          01400           RTS     R5              ;RETURN
5964                                          01500
5965                                          01600
5966  034622                                  01700   NPRSET:
5967                                          01800           ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
5968                                          01900           ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)
5969                                          02000
5970  034622  010246                          02100           MOV     R2,-(SP)        ;SAVE R2
5971  034624  005002                          02200           CLR     R2              ;START AT IBUS REG 0
5972  034626  112561  000004                  02300   1$:     MOVB    (R5)+,4(R1)     ;LOAD PORT4
5973  034632  042737  000017  034646          02400           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
5974  034640  050237  034646                  02500           BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
5975  034644  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5976  034646  122100                          02700   2$:     122100                  ;MOVE PORT4 TO IBUS REG
```

```
  5977  034650  005202                  02800            INC     R2              ;NEXT ADDRESS
  5978  034652  022702  000010          02900            CMP     #10,R2          ;ALL DONE?
  5979  034656  001363                  03000            BNE     1$              ;BR IF NO
  5980  034660  012602                  03100            MOV     (SP)+,R2        ;RESTORE R2
  5981  034662  000205                  03200            RTS     R5              ;RETURN
  5982                                  03300
  5983                                  03400
  5984  034664                          03500    ME%LD:
  5985                                  03600            ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
  5986                                  03700            ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252
  5987                                  03800
  5988  034664  013605                  03900            MOV     @(SP)+,R5       ;PUT POINTER TO DATA IN R5
  5989  034666  062746  000002          04000            ADD     #2,-(SP)        ;ADJUST STACK
  5990  034672  012704  000010          04100            MOV     #10,R4          ;DO 8 LOADS
  5991  034676  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  5992  034700  010000                  04300            010000                  ;MAR < 0
  5993  034702  112577  144504          04400    1$:     MOVB    (R5)+,@DMP04    ;LOAD PORT4
  5994  034706  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  5995  034710  136500                  04600            136500                  ;MOV DATA TO MEM, AUTO INC MAR
  5996  034712  005304                  04700            DEC     R4              ;DECREMENT COUNT
  5997  034714  001372                  04800            BNE     1$              ;BR IF NOT DONE
  5998  034716  000207                  04900            RTS     PC              ;RETURN
  5999                                  05000
  6000                                  05100
  6001  034720                          00200    SPLD:
  6002                                  00300            ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
  6003                                  00400            ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252
  6004                                  00500
  6005  034720  013605                  00600            MOV     @(SP)+,R5       ;PUT POINTER TO DATA IN R5
  6006  034722  062746  000002          00700            ADD     #2,-(SP)        ;ADJUST STACK
  6007  034726  005004                  00800            CLR     R4              ;START AT SP ADDRESS 0
  6008  034730  112577  144456          00900    1$:     MOVB    (R5)+,@DMP04    ;LOAD PORT4 WITH DATA
  6009  034734  042737  000017  034750  01000            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
  6010  034742  050437  034750          01100            BIS     R4,2$           ;ADD ADDRESS TO INSTRUCTION
  6011  034746  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  6012  034750  123100                  01300    2$:     123100                  ;MOVE DATA TO SP
  6013  034752  005204                  01400            INC     R4              ;INCREMENT COUNT
  6014  034754  022704  000010          01500            CMP     #10,R4          ;DONE YET?
  6015  034760  001363                  01600            BNE     1$              ;BR IF NO
  6016  034762  000207                  01700            RTS     PC              ;RETURN
  6017                                  01800
  6018                                  01900
  6019  034764                          02000    CLRC:
  6020                                  02100            ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT
  6021                                  02200
  6022  034764  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  6023  034766  010000                  02400            010000                  ;MAR←0
  6024  034770  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  6025  034772  040400                  02600            040400!<0*20>           ;CLEAR C BIT
  6026  034774  000207                  02700            RTS     PC              ;RETURN
  6027                                  02800
  6028                                  02900
  6029  034776                          03000    SETC:
  6030                                  03100            ;THIS SUBROUTINE SETS THE MICRO PROCESSOR C BIT
  6031                                  03200
  6032  034776  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
 6033  035000  010003                    03400          010003                  ;MAR←3
 6034  035002  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=53C4
 6035  035004  040403                    03600          040403!<0*20>           ;SET C BIT  ~
 6036  035006  000207                    03700          RTS     PC              ;RETURN
 6037                                     03800
 6038                                     03900
 6039  035010     000     377     000    04000  MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
 6040  035013     377     125     252
 6041  035016     125     252
 6042  035020     000     000     377    04100  SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
 6043  035023     377     125     125
 6044  035026     252     252
 6045                                     04200  .EVEN
 6046  035030  052777  044516  052502    00300  EM1:    .ASCIZ  <377>/UNIBUS REGISTER ADDRESSING TIME-OUT/
       035075     377  047125  041111    00400  EM2:    .ASCIZ  <377>↑UNIBUS REGISTER WRITE/READ TEST↑
       035136  046777  041511  047522    00500  EM3:    .ASCIZ  <377>/MICRO PROCESSOR TEST/
       035164  046777  041511  047522    00600  EM4:    .ASCIZ  <377>↑MICRO PROCESSOR WRITE/READ TEST↑
       035225     377  051102  051040    00700  EM5:    .ASCIZ  <377>/BR REGISTER TEST/
       035247     377  041523  040522    00800  EM6:    .ASCIZ  <377>/SCRATCH PAD TEST/
       035271     377  042504  044526    00900  EM7:    .ASCIZ  <377>/DEVICE FAILED TO INTERRUPT/
       035325     377  042504  044526    01000  EM10:   .ASCIZ  <377>/DEVICE INTERRUPTED TO WRONG VECTOR/
       035371     377  050116  020122    01100  EM11:   .ASCIZ  <377>/NPR TEST/
       035403     377  040515  047111    01200  EM12:   .ASCIZ  <377>/MAIN MEMORY TEST/
       035425     377  040515  020122    01300  EM13:   .ASCIZ  <377>/MAR TEST/
       035437     377  046101  020125    01400  EM14:   .ASCIZ  <377>/ALU TEST/
       035451     377  051120  043517    01500  EM15:   .ASCIZ  <377>/PROGRAM CLOCK TEST/
       035475     377  047506  041522    01600  EM16:   .ASCIZ  <377>/FORCE POWER FAIL ERROR/
       035525     377  047125  054105    01700  EM17:   .ASCIZ  <377>/UNEXPECTED INTERRUPT/
       035553     377  046504  030503    01800  EM20:   .ASCIZ  <377>/DMC11 CONFIGURATION ERROR/
       035606  046777  044501  052116    01900  EM21:   .ASCIZ  <377>/MAINTENANCE INSTRUCTION REGISTER TEST
       035655     377  047520  042527    02000  EM22:   .ASCIZ  <377>/POWER FAIL INITIALIZE FAILURE
                                         02100
       035714  051377  043505  051511    02200  DH1:    .ASCIZ  <377>/REGISTER          TRAPPED FROM
       035755     377  054105  042520    02300  DH2:    .ASCIZ  <377>/EXPECTED  FOUND    REGISTER/
       036013     377  054105  042520    02400  DH3:    .ASCIZ  <377>/EXPECTED  FOUND
       036034  042777  050130  041505    02500  DH4:    .ASCIZ  <377>/EXPECTED  FOUND   IBUS* REGISTER
       036076  042777  050130  041505    02600  DH5:    .ASCIZ  <377>/EXPECTED  FOUND   IBUS REGISTER
       036137     377  054105  042520    02700  DH6:    .ASCIZ  <377>/EXPECTED  FOUND    ADDRESS
                                         02800  .EVEN
                                         02900
       036174  000002                    03000  DT1:    2
       036176     006     015            03100          .BYTE   6,15
       036200  001262                    03200          SAVR1
       036202     006     002            03300          .BYTE   6,2
       036204  001264                    03400          SAVR2
       036206  000003                    03500  DT2:    3
       036210     006     004            03600          .BYTE   6,4
       036212  001272                    03700          SAVR5
       036214     006     004            03800          .BYTE   6,4
       036216  001270                    03900          SAVR4
       036220     006     002            04000          .BYTE   6,2
       036222  001262                    04100          SAVR1
       036224  000002                    04200  DT3:    2
       036226     006     004            04300          .BYTE   6,4
       036230  001272                    04400          SAVR5
       036232     006     002            04500          .BYTE   6,2
```

```
036234  001270                  04600           SAVR4
036236  000003                  04700   DT4:    3
036240      003     007         04800           .BYTE   3,7
036242  001272                  04900           SAVR5
036244      003     011         05000           .BYTE   3,11
036246  001270                  05100           SAVR4
036250      002     002         05200           .BYTE   2,2
036252  001264                  05300           SAVR2
036254  000003                  05400   DT5:    3
036256      003     007         05500           .BYTE   3,7
036260  001272                  05600           SAVR5
036262      003     007         05700           .BYTE   3,7
036264  001270                  05800           SAVR4
036266      006     002         05900           .BYTE   6,2
036270  001264                  06000           SAVR2
036272  000002                  06100   DT6:    2
036274      003     007         06200           .BYTE   3,7
036276  001272                  06300           SAVR5
036300      003     002         06400           .BYTE   3,2
036302  001270                  06500           SAVR4
036304  000002                  06600   DT7:    2
036306      006     004         06700           .BYTE   6,4
036310  001262                  06800           SAVR1
036312      006     002         06900           .BYTE   6,2
036314  001404                  07000           DMCSR
036316                          07100   .ERRTAB:
036316  000000                  07200           0
036320  000000                  07300           0
036322  000000                  07400           0
036324  035030                  07500           EM1
036326  035714                  07600           DH1     ;HLT    1
036330  036174                  07700           DT1
036332  035075                  07800           EM2
036334  035755                  07900           DH2     ;HLT    2
036336  036206                  08000           DT2
036340  035136                  08100           EM3
036342  036013                  08200           DH3     ;HLT    3
036344  036224                  08300           DT3
036346  035164                  08400           EM4
036350  036034                  08500           DH4     ;HLT    4
036352  036236                  08600           DT4
036354  035164                  08700           EM4
036356  036076                  08800           DH5     ;HLT    5
036360  036236                  08900           DT4
036362  035225                  09000           EM5
036364  036013                  09100           DH3     ;HLT    6
036366  036272                  09200           DT6
036370  035247                  09300           EM6
036372  036137                  09400           DH6     ;HLT    7
036374  036254                  09500           DT5
036376  035271                  09600           EM7
036400  000000                  09700           0       ;HLT    10
036402  000000                  09800           0
036404  035325                  09900           EM10
036406  000000                  10000           0       ;HLT    11
036410  000000                  10100           0
```

```
036412  035371                  10200           EM11
036414  036013                  10300           DH3     ;HLT    12
036416  036224                  10400           DT3
036420  035403                  10500           EM12
036422  036137                  10600           DH6     ;HLT    13
036424  036254                  10700           DT5
036426  035425                  10800           EM13
036430  036137                  10900           DH6     ;HLT    14
036432  036254                  11000           DT5
036434  035437                  11100           EM14
036436  036013                  11200           DH3     ;HLT    15
036440  036272                  11300           DT6
036442  035451                  11400           EM15
036444  036034                  11500           DH4     ;HLT    16
036446  036236                  11600           DT4
036450  035475                  11700           EM16
036452  000000                  11800           0       ;HLT    17
036454  000000                  11900           0
036456  035525                  12000           EM17
036460  000000                  12100           0       ;HLT    20
036462  000000                  12200           0
036464  035371                  12300           EM11
036466  036137                  12400           DH6     ;HLT    21
036470  036254                  12500           DT5
036472  035553                  12600           EM20
036474  036013                  12700           DH3     ;HLT    22
036476  036304                  12800           DT7
036500  035606                  12900           EM21
036502  036013                  13000           DH3     ;HLT    23
036504  036224                  13100           DT3
036506  035553                  13200           EM20
036510  000000                  13300           0       ;HLT    24
036512  000000                  13400           0
036514  035655                  13500           EM22
036516  036013                  13600           DH3     ;HLT    25
036520  036224                  13700           DT3
                                13800
036522                          13900   CORMAX:
        000001                  14400   .END
```

```
ADRCNI= 004373   881*   917*   926#
AUDONE  003024   571    610    615    661#
AUSTRT  002446   570#   665
AUTO.S  010512   528    1370#
BINWRD  004714   967*   970*   971    1008#
BIT0   = 000001   95#    1157   1158   1769   1777   2039   2047   3866   5712   5714   5715
BIT1   = 000002   94#    533    1151   1157   1158   1531   1534   1547   1799   1807   2069   2077   5710
                  5712   5714   5715
BIT10  = 002000   85#    1515   1526   2377   2396   2418   2437   3726   3762   5778
BIT11  = 004000   84#    1979   1987
BIT12  = 010000   83#    1466   1545   2009   2017   2189   2197
BIT13  = 020000   82#    1469   1518   1549   2219   2227
BIT14  = 040000   81#    125    783    1480   1482   1549   1560   2436
BIT15  = 100000   80#    487    574    577    598    601    1453   1521   4056   4069   5871   5930
BIT2   = 000004   93#    533    714    1158   1829   1837   2099   2107
BIT3   = 000010   92#    1551   1558
BIT4   = 000020   91#    1141   1164   1166
BIT5   = 000040   90#    1663   1859   1667
BIT6   = 000100   89#    1146   1147   1553   1889   1897   2129   2137
BIT7   = 000200   88#    1147   1267   1663   1919   1927   2159   2167
BIT8   = 000400   87#    1525   1540   1542   1555   1557   1563   1566   1624   2463   3727   3763   5761
BIT9   = 001000   86#    1523   1525   1537   1563   1566   1622   1624   1949   1957   2377   2396   2418
                  2437   2461   2463   3726   3762   5759   5761   5778
BM      007054   1189#  1494
BRLVL   012252   1619   1629   1637   1649#
BPW     003730   720    806#
BRX     003732   721    807#
CHRCNT  004712   965*   968    972    988*   1006#  1007
CKSWR   007606   514    781    813    1028   1214#
CKSWR1  007666   1227#  1239
CKSWR2  007700   1230#
CKSWR3  007704   1232#
CKSWR4  007710   1233#  1241   1248
CKSWR5  010014   1215   1222   1257#
CLKX    001242   173#
CLRC    034764   4146   4195   4244   4293   4342   4391   4440   4489   4538   4587   4636   4685   4734
                 4783   4832   4881   6019#
CNERR   007277   628    1189#
CNT.MA  001702   198    366#   486    488    490    1295
CNVRT = 104411   235#   625    740    742    744    746    1062   1064   1124   1230
CONERR  007223   623    1189#
CONN    007114   1189#  1471
CONTAB  002776   632    645#
CONVRT= 104410   233#   545    631    1078   1393
CORMAX  036522   3862   3894   6046#
CRAM    006606   1189#  1432
CREAM   001320   197#   485*   1291*  1292   1294*  1298
CSR     006510   1189#  1397
CSRMAP  010514   1372#
CYCLE   010060   722    763    764    1282#
DATABP  005216   1051*  1054   1076   1079#
DATACL= 104415   243#
DATAHD  005204   1050*  1072   1075#
DELAY = 104413   239#
DEVADR  004370   879*   914    924#
DEVTAB  003010   585    650#
```

```
DH1     035714          6046#
DH2     035755          6046#
DH3     036013          6046#
DH4     036034          6046#
DH5     036076          6046#
DH6     036137          6046#
DISPLA  001200           144#   500*   506*   737*
DISPRE  000174           130#   506
DMACTV  001306           191#   523    678*   679    1282   1296   1378*  1579*  1585*  1586*  1590   1615
DMCM    007320           636   1189#
DMCR00  001500           281#
DMCR01  001510           286#
DMCR02  001520           291#
DMCR03  001530           296#
DMCR04  001540           301#
DMCR05  001550           306#
DMCR06  001560           311#
DMCR07  001570           316#
DMCR10  001600           321#
DMCR11  001610           326#
DMCR12  001620           331#
DMCR13  001630           336#
DMCR14  001640           341#
DMCR15  001650           346#
DMCR16  001660           351#
DMCR17  001670           356#
DMCSR   001404           264#   570*   604    609*   614*   649    767    804    1096   1119   1165   1300*  1309
                        1358   1679   1740   1768   1798   1828   1858   1888   1918   1948   1978   2008   2348
                        6046
DMCSRH  001406           265#  1146*  1147*  1151*  1157*  1158*  1164*  1166*  1309*  1310*  1311
DMCTL   001410           266#  1311*  1312*  1313   2038   2068   2098   2128   2158   2188   2218
DMNUM   001310           192#   481    753   1376*  1391*  1570*  1571   1580   1582
DMP04   001412           267#  1135*  1141   1178   1183   1313*  1314*  1315   2248   3714   3715   3753   3754
                        5993*  6008*
DMP06   001414           268#  1152*  1315*  1316*  2291
DMRLVL  001376           261#  1318*  1319*  1320   5961*
DMRVEC  001374           260#   770   1301*  1302*  1318   5959*
DMS100  001502           282#
DMS101  001512           287#
DMS102  001522           292#
DMS103  001532           297#
DMS104  001542           302#
DMS105  001552           307#
DMS106  001562           312#
DMS107  001572           317#
DMS110  001602           322#
DMS111  001612           327#
DMS112  001622           332#
DMS113  001632           337#
DMS114  001642           342#
DMS115  001652           347#
DMS116  001662           352#
DMS117  001672           357#
DMS200  001504           283#
DMS201  001514           288#
DMS202  001524           293#
```

```
DMS203  001534            298#
DMS204  001544            303#
DMS205  001554            308#
DMS206  001564            313#
DMS207  001574            318#
DMS210  001604            323#
DMS211  001614            328#
DMS212  001624            333#
DMS213  001634            338#
DMS214  001644            343#
DMS215  001654            348#
DMS216  001664            353#
DMS217  001674            358#
DMS300  001506            284#
DMS301  001516            289#
DMS302  001526            294#
DMS303  001536            299#
DMS304  001546            304#
DMS305  001556            309#
DMS306  001566            314#
DMS307  001576            319#
DMS310  001606            324#
DMS311  001616            329#
DMS312  001626            334#
DMS313  001636            339#
DMS314  001646            344#
DMS315  001656            349#
DMS316  001666            354#
DMS317  001676            359#
DMTLVL  001402            263#   1322*   1323*   5962*
DMTVEC  001400            262#   1320*   1321*   1322    5960*
DM.END  001700            361#   1374
DM.MAP  001500            197     280#    485     538     548     664    1292    1294    1372    1377    1607
DONE    003734            786     788*    808#   1218*
DT1     036174           6046#
DT2     036206           6046#
DT3     036224           6046#
DT4     036236           6046#
DT5     036254           6046#
DT6     036272           6046#
DT7     036304           6046#
EM1     035030           6046#
EM10    035325           6046#
EM11    035371           6046#
EM12    035403           6046#
EM13    035425           6046#
EM14    035437           6046#
EM15    035451           6046#
EM16    035475           6046#
EM17    035525           6046#
EM2     035075           6046#
EM20    035553           6046#
EM21    035606           6046#
EM22    035655           6046#
EM3     035136           6046#
EM4     035164           6046#
```

```
EM5     035225          6046#
EM6     035247          6046#
EM7     035271          6046#
ERCT00  001704           368#
ERCT01  001710           371#
ERCT02  001714           374#
ERCT03  001720           377#
ERCT04  001724           380#
ERCT05  001730           383#
ERCT06  001734           386#
ERCT07  001740           389#
ERCT10  001744           392#
ERCT11  001750           395#
ERCT12  001754           398#
ERCT13  001760           401#
ERCT14  001764           404#
ERCT15  001770           407#
ERCT16  001774           410#
ERCT17  002000           413#
ERR     002700           594     620#    624
ERRCNT  001232           165#    749     776     1089#   1307#
ERRFLG  001325           204#    483#    735#    797#    1039#   1052    1066#   1125#
ERRMSG  005172          1049#   1067    1070#
ERRPC   002770           626     642#
ERTABO  005322          1064    1099#
EXIT  = 000205            96#
EXITER  005252          1084    1099#
FLOAT   002536           587#    593
FY      002566           596#    607     611     617
HALTS   005222          1035    108
HILIM   004366           878#    905     923#
ICOUNT  001222           161#    795     800#    3856#
INBUF   007502           848     884     1204#
INCHAR  010020          1233    1261#
INIFLG  001324           203#    509     529     536#
INSTER= 104404           225#    899
INSTR = 104403           223#    1331    1383    1396    1405    1484    1493
INSTR2  004166           855     867#
INTTY   012266          1416    1433    1443    1456    1472    1657#
KMCM    007330           639    1189#
LIMITS  004314           894     905#
LINE    007016          1189#   1485
LOBITS  004372           880#    909     925#    926
LOCK    001220           160#    799#    816     818     1058    1677#   1724#   1742#   1766#   1776#   1796#   1806#   1826#
                        1836#   1856#   1866#   1886#   1896#   1916#   1926#   1946#   1956#   1976#   1986#   2006#   2016#
                        2036#   2046#   2066#   2076#   2096#   2106#   2126#   2136#   2156#   2166#   2186#   2196#   2216#
                        2226#   2246#   2261#   2289#   2304#   2332#   2350#   2374#   2385#   2393#   2415#   2426#   2434#
                        2483#   2505#   2539#   2561#   2595#   2615#   2647#   2667#   2700#   2722#   2758#   2782#   2818#
                        2838#   2870#   2890#   2922#   2942#   2974#   2994#   3026#   3046#   3078#   3098#   3130#   3150#
                        3182#   3202#   3234#   3263#   3295#   3314#   3346#   3371#   3412#   3439#   3912#   3952#   3995#
                        4019#   4096#   4137#   4186#   4235#   4284#   4333#   4382#   4431#   4480#   4529#   4578#   4627#
                        4676#   4725#   4774#   4823#   4872#   4921#   4970#   5019#   5068#   5117#   5166#   5215#   5264#
                        5313#   5362#   5411#   5460#   5509#   5558#   5607#   5656#
LOKFLG  001326           205#
LOLIM   004364           877#    907     922#
LPCNT   001224           162#    794#    795     798#
```

```
LSTERR  001234  166# 492* 734* 1036 1038* 1126*
M3SKX   001244  174#
MASTEK  006142 1060 1189#
MCRLF   005672  833  956 1056 1057 1065 1189# 1330 1392
MCSRX   006072  739 1189#
MDATA   007544  986  996 1206#
MEMDAT  035010 4143 4192 4241 4290 4339 4388 4437 4486 4535 4584 4633 4682 4731
               4780 4829 4878 4927 4976 5025 5074 5123 5172 5221 5270 5319 5368
               5417 5466 5515 5564 5613 5662 6039#
MEMLD   034664 4099 4142 4191 4240 4289 4338 4387 4436 4485 4534 4583 4632 4681
               4730 4779 4828 4877 4926 4975 5024 5073 5122 5171 5220 5269 5318
               5367 5416 5465 5514 5563 5612 5661 5984#
MEMLIM  001304  190#  702* 3892
MEPASS  005733  738 1189#
MERRPC  006217 1063 1189#
MERRX   006117  745 1189#
MERR2   005760 1189# 1592
MERR3   006005  675 1189#
MILK    001322  198#  486* 747 1290* 1295* 1299
MLOCK   006043  716 1189#
MNEW    006144  670 1189#
MODU    006704 1189# 1455
MPASSX  006106  743 1189#
MPFAIL  005675 1123 1189#
MQM     005666  863 1189# 1354 1429 1439 1449 1464 1478
MR      005755  725 1189# 1348
MRESET= 004000   96#
MSTCLR= 104412  237# 1128 1726 1767 1797 1827 1857 1887 1917 1947 1977 2007 2037
               2067 2097 2127 2157 2187 2217 2247 2290 2334 2376 2417 2459 2485
               2541 2597 2649 2702 2760 2820 2872 2924 2976 3028 3080 3132 3184
               3236 3297 3348 3414 3501 3530 3568 3644 3680 3713 3752 3788 3823
               3859 3914 3954 3997 4052 4098 4139 4188 4237 4286 4335 4384 4433
               4482 4531 4580 4629 4678 4727 4776 4825 4874 4923 4972 5021 5070
               5119 5168 5217 5266 5315 5364 5413 5462 5511 5560 5609 5658 5705
               5754 5803
MTITLE  001000  138#  513
MTSTN   006130 1061 1189# 1332
MTSTPC  006031 1189#
MVECX   006100  741 1189#
NEXT    001216  159#  801 1094 1676* 1704* 1723* 1765* 1795* 1825* 1855* 1885* 1915* 1945*
               1975* 2005* 2035* 2065* 2095* 2125* 2155* 2185* 2215* 2245* 2288* 2331* 2373*
               2414* 2457* 2482* 2538* 2594* 2646* 2699* 2757* 2817* 2869* 2921* 2973* 3025*
               3077* 3129* 3181* 3233* 3294* 3345* 3411* 3470* 3499* 3528* 3566* 3609* 3642*
               3678* 3711* 3750* 3786* 3821* 3857* 3911* 3951* 3994* 4050* 4095* 4136* 4185*
               4234* 4283* 4332* 4381* 4430* 4479* 4528* 4577* 4626* 4675* 4724* 4773* 4822*
               4871* 4920* 4969* 5018* 5067* 5116* 5165* 5214* 5263* 5312* 5361* 5410* 5459*
               5508* 5557* 5606* 5655* 5703* 5752* 5801*
NOJCT   007154  525 1189# 1284
NODEV   002674  579  618#
NPRSET  034622 3614 3646 3682 3716 3755 3789 3824 3869 5966#
NUM     006450 1189# 1384
OK      002646  605  612#  641
ONE     001302  189#
PAGE00  001702  367#
PAGE01  001706  370#
PAGE02  001712  373#
```

```
PACT03  001716            376#
PACT04  001722            379#
PACT05  001726            382#
PACT06  001732            385#
PACT07  001736            388#
PACT10  001742            391#
PACT11  001746            394#
PACT12  001752            397#
PACT13  001756            400#
PACT14  001762            403#
PACT15  001766            406#
PACT16  001772            409#
PACT17  001776            412#
PARAM = 104405            227#  1333  1385  1398  1407  1486  1495
PARAM1  004234            883#   900
PARBIT= 040000             96#
PARERR  004310            886    888   890   899#  906   908   910
PASCNT  001230            164#   736*  737   748   773   1306*
PERFOR= 004537             96#
PFTAB   005430           1124   1130#
POPRO = 012600             72#  1088
POP1SP= 005726             70#
POP2SP= 022626             74#   803   5771
PRIO    006547           1189#  1415
PS    = 177776             63#   478*  713*  1619*  1629*  3478*  3482*  3506*  3510*  3532*  3546*  3570*  3586*
PUSHR0= 010046             71#  1085
PUSH1S= 005746             69#
PUSH2S= 024646             73#
QV.FLG  001327            206#   484*  752*  792
RESREG  005220           1077   1080#
RESTAR  005350           1110   1116#
RESTRT  003534            751    755   763#
RESOS = 104407            231#  1080
RETURN  001214            158#   494*  722*  726   763*  801*  805   1094*  1097  1129  1347*  1357*  1359
ROMCLK= 104414            241#  1136  1139  1176  1181  2491  2493  2511  2513  2547  2549  2567  2569
                         2602  2604  2620  2622  2654  2656  2672  2674  2708  2710  2728  2730  2766
                         2768  2788  2790  2825  2827  2843  2845  2877  2879  2895  2897  2929  2931
                         2947  2949  2981  2983  2999  3001  3033  3035  3051  3053  3085  3087  3103
                         3105  3137  3139  3155  3157  3189  3191  3207  3209  3243  3251  3269  3301
                         3303  3319  3321  3354  3358  3360  3377  3381  3383  3421  3425  3427  3445
                         3447  3480  3508  3544  3584  3621  3653  3657  3659  3689  3722  3730  3732
                         3766  3768  3795  3798  3801  3830  3833  3836  3876  3919  3922  3924  3926
                         3960  3963  3965  3967  4001  4004  4006  4008  4023  4025  4027  4054  4058
                         4061  4067  4071  4074  4104  4106  4108  4110  4149  4153  4155  4198  4202
                         4204  4247  4251  4253  4296  4300  4302  4345  4349  4351  4394  4398  4400
                         4443  4447  4449  4492  4496  4498  4541  4545  4547  4590  4594  4596  4639
                         4643  4645  4688  4692  4694  4737  4741  4743  4786  4790  4792  4835  4839
                         4841  4884  4888  4890  4933  4937  4939  4982  4986  4988  5031  5035  5037
                         5080  5084  5086  5129  5133  5135  5178  5182  5184  5227  5231  5233  5276
                         5280  5282  5325  5329  5331  5374  5378  5380  5423  5427  5429  5472  5476
                         5478  5521  5525  5527  5570  5574  5576  5619  5623  5625  5668  5672  5674
                         5722  5808  5821  5824  5827  5840  5843  5846  5852  5955  5858  5861  5869
                         5873  5885  5903  5915  5918  5928  5932  5935  5975  5991  5994  6011  6222
                         6024  6032  6034
R0      001316            195#   487*  1288#  1289*  1296
SP/ACT  001312            193#   673   1590#
```

```
SAVNUM  001314  194#  481*  750*  753*  1583*
SAVPC   001276  187#  624*  644   931*  1101
SAVR0   001260  180#  940*  945
SAVR1   001262  181#  630*  939*  946   6046
SAVR2   001264  182#  938*  947   6046
SAVR3   001266  183#  937*  948
SAVR4   001270  184#  936*  949   6046
SAVR5   001272  185#  935*  950   6046
SAVSP   001274  186#
SAVOS = 104406  229#  1040
SCOPE = 104400  217#  1691  1711  1753  1783  1813  1843  1873  1903  1933  1963  1993  2023
                2053  2083  2113  2143  2173  2203  2233  2276  2319  2361  2402  2443  2470
                2526  2582  2634  2686  2743  2805  2857  2909  2961  3013  3065  3117  3169
                3221  3282  3333  3399  3459  3488  3516  3552  3593  3629  3665  3697  3738
                3774  3809  3844  3897  3940  3982  4038  4084  4118  4167  4216  4265  4314
                4363  4412  4461  4510  4559  4608  4657  4706  4755  4804  4853  4902  4951
                5000  5049  5098  5147  5196  5245  5294  5343  5392  5441  5490  5539  5588
                5637  5686  5731  5786  5944
SCOP1 = 104401  219#  1685  1735  1748  1775  1782  1805  1812  1835  1842  1865  1872  1895
                1902  1925  1932  1955  1962  1985  1992  2015  2022  2045  2052  2075  2082
                2105  2112  2135  2142  2165  2172  2195  2202  2225  2232  2257  2271  2300
                2314  2343  2356  2384  2392  2401  2425  2433  2442  2501  2521  2557  2577
                2611  2629  2643  2681  2718  2738  2778  2800  2834  2852  2886  2904  2938
                2956  2990  3008  3042  3060  3094  3112  3146  3164  3198  3216  3258  3276
                3310  3328  3367  3390  3434  3454  3884  3933  3974  4015  4034  4117  4162
                4211  4260  4309  4358  4407  4456  4505  4554  4603  4652  4701  4750  4799
                4848  4897  4946  4995  5044  5093  5142  5191  5240  5289  5338  5387  5436
                5485  5534  5583  5632  5681
SETC    034776  4930  4979  5028  5077  5126  5175  5224  5273  5322  5371  5420  5469  5518
                5567  5616  5665  6029#
SETVEC  034600  3474  3502  3539  3579  5956#
SKIP    002632  575   578   599   602   608#
SOFTSW  010052  1231  1270#
SPACNT= 004713  966*  990   993*  1007#
SPDAT   035020  4145  4194  4243  4292  4341  4390  4439  4488  4537  4586  4635  4684  4733
                4782  4831  4880  4929  4978  5027  5076  5125  5174  5223  5272  5321  5370
                5419  5468  5517  5566  5615  5664  6042#
SPEED   007340  1189#  1442
SPLC    034720  4101  4144  4193  4242  4291  4340  4389  4438  4487  4536  4585  4634  4683
                4732  4781  4830  4879  4928  4977  5026  5075  5124  5173  5222  5271  5320
                5369  5418  5467  5516  5565  5614  5663  6001#
STACK = 001200  64#   479   692   1095  1118  3487  3515  5777
STAT    001240  172#
STAT1   001366  253#  1303#  3533  3571  4056  4069  5871  5930
STAT2   001370  254#  1304*
STAT3   001372  255#  1305*
STRTSW  001236  171#  515*  518*  519   521   531   533   668   714   723   1325  1349*  1379
                1603
SVOS    004402  935#
SWF_G   010016  482*  827   1226*  1253*  1259#
SWMES   007205  1189#  1229
SWMES1  007215  1189#  1232
SWP     001202  145#  499*  501   505*  515   673   678   783   790   914   929   1029  1034
                1083  1090  1092  1154  1214  1252*
SWREG   000176  131#  505   1214  1272
SWSS  = 000001  45#   519   1379  1603
```

```
SW01  = 000002          44#   723   1325   1349
SW02  = 000004          43#
SW03  = 000010          42#   668
SW04  = 000020          41#
SW05  = 000040          40#
SW06  = 000100          39#  1154
SW07  = 000200          38#
SW08  = 000400          37#  1090
SW09  = 001000          36#   814
SW10  = 002000          35#  1092
SW11  = 004000          34#   790
SW12  = 010000          33#   829   1029
SW13  = 020000          32#  1034
SW14  = 040000          31#
SW15  = 100000          30#
TDATA   024542        4100   4102   4119#
TEMP    001416         273#   973   1120*  1121*  1162*  1167*  1173*  1185*  5706*  5726*  5755*  5762*
TEMP1   001246         175#   539*  1191   1615*  1616*  5707*  5727*  5728
TEMP2   001250         176#   540*  1193
TEMP3   001252         177#   542*   568*   620   629*  1195   1388   1391   1503*
TEMP4   001254         178#   543*  1197   1401   1404   1410   1413   1489   1492   1498   1501
TEMP5   001256         179#   544*  1199   1382*  1395*  1601
TIMER = 104416         245#
TKCSR   001204         150#   850   1216   1261   1657
TKDBR   001206         151#   852    858   1219   1221   1263   1659
.LAST = 033724        1352   6046#
TPCSR   001210         152#   834    856   1031   1264   1660
TPDBR   001212         153#   836*   858*  1033*  1266*  1662*
TPPOK   004730        1019#
TSTAC   001226         163#   493*  1104   1132   1336   1343   1345   1675*  1703*  1722*  1764*  1794*  1824*
                      1854*  1884*  1914*  1944*  1974*  2004*  2034*  2064*  2094*  2124*  2154*  2184*  2214*
                      2244*  2287*  2330*  2372*  2413*  2456*  2481*  2537*  2593*  2645*  2698*  2756*  2815*
                      2868*  2920*  2972*  3024*  3076*  3128*  3180*  3232*  3293*  3344*  3410*  3469*  3498*
                      3527*  3565*  3608*  3641*  3677*  3710*  3749*  3785*  3820*  3955*  3910*  3950*  3993*
                      4049*  4094*  4135*  4184*  4233*  4282*  4331*  4380*  4429*  4478*  4527*  4576*  4625*
                      4674*  4723*  4772*  4821*  4870*  4919*  4968*  5017*  5066*  5115*  5164*  5213*  5262*
                      5311*  5360*  5409*  5458*  5507*  5556*  5605*  5654*  5702*  5751*  5800*

TST1    012320        1339   1357   1675#
TST10   013176        1855   1884#
TST100  025256        4234   4282#
TST101  025432        4283   4331#
TST102  025606        4332   4380#
TST103  025762        4381   4429#
TST104  026136        4430   4478#
TST105  026312        4479   4527#
TST106  026466        4528   4576#
TST107  026642        4577   4625#
TST11   013274        1885   1914#
TST110  027016        4626   4674#
TST111  027172        4675   4723#
TST112  027346        4724   4772#
TST113  027522        4773   4821#
TST114  027676        4822   4870#
TST115  030052        4871   4919#
TST116  030226        4920   4968#
TST117  030402        4969   5017#
```

```
TST12   013372          1915    1944#
TST120  030556          5018    5066#
TST121  030732          5067    5115#
TST122  031106          5116    5164#
TST123  031262          5165    5213#
TST124  031436          5214    5262#
TST125  031612          5263    5311#
TST126  031766          5312    5360#
TST127  032142          5361    5409#
TST13   013470          1945    1974#
TST130  032316          5410    5458#
TST131  032472          5459    5507#
TST132  032646          5508    5556#
TST133  033022          5557    5605#
TST134  033176          5606    5654#
TST135  033352          5655    5702#
TST136  033532          5703    5751#
TST137  033724          5752    5800#   6046
TST14   013566          1975    2004#
TST140= ****** U        5801
TST15   013664          2005    2034#
TST16   013762          2035    2064#
TST17   014060          2065    2094#
TST2    012426          1676    1703#
TST20   014156          2095    2124#
TST21   014254          2125    2154#
TST22   014352          2155    2184#
TST23   014450          2185    2214#
TST24   014546          2215    2244#
TST25   014672          2245    2287#
TST26   015016          2288    2330#
TST27   015146          2331    2372#
TST3    012456          1704    1722#
TST30   015306          2373    2413#
TST31   015450          2414    2456#
TST32   015534          2457    2481#
TST33   015734          2482    2537#
TST34   016134          2538    2593#
TST35   016310          2594    2645#
TST36   016464          2646    2698#
TST37   016664          2699    2756#
TST4    012606          1723    1764#
TST40   017104          2757    2816#
TST41   017260          2817    2868#
TST42   017434          2869    2920#
TST43   017610          2921    2972#
TST44   017764          2973    3024#
TST45   020140          3025    3076#
TST46   020314          3077    3128#
TST47   020470          3129    3180#
TST5    012704          1765    1794#
TST50   020644          3181    3232#
TST51   021072          3233    3293#
TST52   021242          3294    3344#
TST53   021510          3345    3410#
TST54   021732          3411    3469#
```

```
TST55   022026            3470    3498#
TST56   022120            3499    3527#
TST57   022240            3528    3565#
TST6    013002            1795    1824#
TST60   022404            3566    3608#
TST61   022510            3609    3641#
TST62   022624            3642    3677#
TST63   022726            3678    3710#
TST64   023064            3711    3749#
TST65   023210            3750    3785#
TST66   023320            3786    3820#
TST67   023426            3821    3855#
TST7    013100            1825    1854#
TST70   023624            3857    3910#
TST71   023752            3911    3950#
TST72   024104            3951    3993#
TST73   024304            3994    4049#
TST74   024440            4050    4094#
TST75   024552            4095    4135#
TST76   024726            4136    4184#
TST77   025102            4185    4233#
TST     003612             717#    718#    720#    721#    784#
TWOSYN= 010000              96#
TYPDAT  005206            1055    1073    1076#
TYPE  = 104402             221#    513     525     537     622     627     635     639     670     675     716     725     739
                           739     741     743     745     833     846     863     956     996    1056    1057    1060    1061
                          1063    1065    1069    1074    1123    1229    1232    1284    1330    1348    1354    1392    1414
                          1428    1431    1438    1441    1448    1454    1463    1470    1477    1592
TYPMSG  005106            1053    1056#
VEC     006526            1189#   1406
VECMAP  012010            1591    1603#
WHICH   012002            1394    1599#
WRDCNT  004710             964#    997#   1005#
WRKO.F  005174            1068    1071#
XBX     005000            1030    1032    1034#
XCSR    003546             740     765#
XERR    003570             746     774#
XHEAD   006224             537    1189#
XLOC    003022             595#    613#    616     647     660#
XPASS   003562             744     771#
XSTATQ  007454             546    1189#
XTSTN   005330            1062    1102#
XVEC    003554             742     768#
ZERO    001300             188#
$000  = ******               1
$ORAP = 177777               1#   1666#   1669    1671#   1695#   1698    1699#   1712#   1715    1718#   1754#   1757    1760#
                          1784#   1787    1790#   1814#   1817    1820#   1844#   1847    1850#   1874#   1877    1880#   1904#
                          1907    1910#   1934#   1937    1940#   1964#   1967    1970#   1994#   1997    2000#   2024#   2027
                          2030#   2054#   2057    2060#   2084#   2087    2090#   2114#   2117    2120#   2144#   2147    2150#
                          2174#   2177    2180#   2204#   2207    2210#   2234#   2237    2240#   2277#   2280    2283#   2320#
                          2323    2326#   2362#   2365    2368#   2403#   2406    2409#   2444#   2447    2452#   2471#   2474
                          2477#   2527#   2530    2533#   2583#   2586    2589#   2635#   2638    2641#   2687#   2690    2694#
                          2744#   2747    2752#   2806#   2809    2812#   2858#   2861    2864#   2910#   2913    2916#   2962#
                          2965    2968#   3014#   3017    3020#   3066#   3069    3072#   3118#   3121    3124#   3170#   3173
                          3176#   3222#   3225    3228#   3283#   3286    3289#   3334#   3337    3340#   3400#   3403    3406#
                          3460#   3463    3465#   3489#   3492    3494#   3517#   3520    3523#   3555#   3558    3561#   3595#
```

$ENDAD 003522
$N = 000137

3602 3604# 3632# 3635 3637# 3668# 3671 3673# 3700# 3703 3706# 3739# 3742
3745# 3775# 3778 3781# 3810# 3813 3816# 3845# 3848 3851# 3901# 3904 3906#
3941# 3944 3946# 3983# 3986 3989# 4039# 4042 4045# 4085# 4088 4090# 4123#
4126 4131# 4172# 4175 4180# 4221# 4224 4229# 4270# 4273 4278# 4319# 4322
4327# 4368# 4371 4376# 4417# 4420 4425# 4466# 4469 4474# 4515# 4518 4523#
4564# 4567 4572# 4613# 4616 4621# 4662# 4665 4670# 4711# 4714 4719# 4760#
4763 4768# 4809# 4812 4817# 4858# 4861 4866# 4907# 4910 4915# 4956# 4959
4964# 5005# 5008 5013# 5054# 5057 5062# 5103# 5106 5111# 5152# 5155 5160#
5201# 5204 5209# 5250# 5253 5258# 5299# 5302 5307# 5348# 5351 5356# 5397#
5400 5405# 5446# 5449 5454# 5495# 5498 5503# 5544# 5547 5552# 5593# 559[illegible]
5601# 5642# 5645 5650# 5691# 5694 5698# 5732# 5735 5747# 5787# 5790 5796#
123 511 757# 1081
1# 1666 1671 1673 1679# 1695 1699 1701 1706# 1712 1718 1720 1726
1727# 1754 1760 1762 1767 1768# 1784 1790 1792 1797 1798# 1814 1820
1822 1827 1828# 1844 1850 1852 1857 1858# 1874 1880 1882 1887 1888#
1904 1910 1912 1917 1918# 1934 1940 1942 1947 1948# 1964 1970 1972
1977 1978# 1994 2000 2002 2007 2008# 2024 2030 2032 2037 2038# 2054
2060 2062 2067 2068# 2084 2090 2092 2097 2098# 2114 2120 2122 2127
2128# 2144 2150 2152 2157 2158# 2174 2180 2182 2187 2188# 2204 2210
2212 2217 2218# 2234 2240 2242 2247 2248# 2277 2283 2285 2290 2291#
2320 2326 2328 2334 2335# 2362 2368 2370 2376 2377# 2403 2409 2411
2417 2418# 2444 2452 2454 2459 2460# 2471 2477 2479 2485 2486# 2523
2533 2535 2541 2542# 2583 2589 2591 2597 2598# 2635 2641 2643 2649
2650# 2687 2694 2696 2702 2703# 2744 2752 2754 2760 2761# 2806 2812
2814 2820 2821# 2858 2864 2866 2872 2873# 2910 2916 2918 2924 2925#
2962 2968 2970 2976 2977# 3014 3020 3022 3028 3029# 3066 3072 3074
3080 3081# 3118 3124 3126 3132 3133# 3170 3176 3178 3184 3185# 3222
3228 3230 3236 3237# 3283 3289 3291 3297 3298# 3334 3340 3342 3348
3349# 3400 3406 3408 3414 3415# 3460 3465 3467 3472 3474# 3489 3494
3496 3501 3502# 3517 3523 3525 3530 3531# 3555 3561 3563 3568 3569#
3599 3604 3606 3611 3613# 3632 3637 3639 3644 3645# 3668 3673 3675
3680 3681# 3700 3706 3708 3713 3714# 3739 3745 3747 3752 3753# 3775
3781 3783 3788 3789# 3810 3816 3818 3823 3824# 3845 3851 3853 3859
3860# 3901 3906 3908 3914 3915# 3941 3946 3948 3954 3955# 3983 3989
3991 3997 3998# 4039 4045 4047 4052 4053# 4085 4090 4092 4098 4099#
4123 4131 4133 4139 4140# 4172 4180 4182 4188 4189# 4221 4229 4231
4237 4[illegible] 4270 4278 4279 4280 4286 4287# 4319 4327 4328 4329 4335
4336# 4368 4376 4377 4378 4384 4385# 4417 4425 4426 4427 4433 4434#
4466 4474 4475 4476 4482 4483# 4515 4523 4524 4525 4531 4532# 4564
4572 4573 4574 4580 4581# 4613 4621 4622 4623 4629 4630# 4662 4670
4671 4672 4678 4679# 4711 4719 4720 4721 4727 4729# 4760 4768 4769
4770 4776 4777# 4809 4817 4818 4819 4825 4826# 4858 4866 4867 4868
4874 4875# 4907 4915 4916 4917 4923 4924# 4956 4964 4965 4966 4972
4973# 5005 5013 5014 5015 5021 5022# 5054 5062 5063 5064 5070 5071#
5103 5111 5112 5113 5119 5120# 5152 5160 5161 5162 5168 5169# 5201
5209 5210 5211 5217 5218# 5250 5258 5259 5260 5266 5267# 5299 5307
5308 5309 5315 5316# 5348 5356 5357 5358 5364 5365# 5397 5405 5406
5407 5413 5414# 5446 5454 5455 5456 5462 5463# 5495 5503 5504 5505
5511 5512# 5544 5552 5553 5554 5560 5561# 5593 5601 5602 5603 5609
5610# 5642 5650 5651 5652 5658 5659# 5691 5698 5699 5700 5705 5706#
5732 5747 5748 5749 5754 5755# 5787 5796 5797 5798 5803 5804# 6046#

$$ = 000141

1# 1676 1679# 1704 1706# 1723 1727# 1765 1768# 1795 1798# 1825 1828#
1855 1858# 1885 1888# 1915 1918# 1945 1948# 1975 1978# 2005 2008# 2035
2038# 2065 2068# 2095 2098# 2125 2129# 2155 2158# 2185 2189# 2215 2218#
2245 2248# 2288 2291# 2331 2335# 2373 2377# 2414 2418# 2457 2460# 2482
2486# 2538 2542# 2594 2598# 2646 2650# 2699 2703# 2757 2761# 28[illegible] 2821#

```
                    2869    2873#   2921    2925#   2973    2977#   3025    3029#   3077    3081#   3129    3133#   3181
                    3185#   3233    3237#   3294    3298#   3345    3349#   3411    3415#   3470    3474#   3499    3502#
                    3528    3531#   3566    3569#   3609    3613#   3642    3645#   3678    3681#   3711    3714#   3750
                    3753#   3786    3789#   3821    3824#   3857    3860#   3911    3915#   3951    3955#   3994    3998#
                    4050    4053#   4095    4099#   4136    4140#   4185    4189#   4234    4238#   4283    4287#   4332
                    4336#   4381    4385#   4430    4434#   4479    4483#   4528    4532#   4577    4581#   4626    4630#
                    4675    4679#   4724    4728#   4773    4777#   4822    4826#   4871    4875#   4920    4924#   4969
                    4973#   5018    5022#   5067    5071#   5116    5120#   5165    5169#   5214    5218#   5263    5267#
                    5312    5316#   5361    5365#   5410    5414#   5459    5463#   5508    5512#   5557    5561#   5606
                    5610#   5655    5659#   5703    5706#   5752    5755#   5801    5804#
$       = 000017     1#     209#    217     219#    221#    223#    225#    227#    229#    231#    233#    235#    237#
                     239#   241#    243#    245#    247#
.       = 036522     108#   109     112#    119#    124#    129#    133#    137#    139#    191#    192#    193#    194#
                     274#   279#    281#    282#    283#    284#    286#    287#    288#    289#    291#    292#    293#
                     294#   296#    297#    298#    299#    301#    302#    303#    304#    306#    307#    308#    309#
                     311#   312#    313#    314#    316#    317#    318#    319#    321#    322#    323#    324#    326#
                     327#   328#    329#    331#    332#    333#    334#    336#    337#    338#    339#    341#    342#
                     343#   344#    346#    347#    348#    349#    351#    352#    353#    354#    356#    357#    358#
                     359#   517     527     659#    677     1112    1122    1170#   1205#   1207#   1262    1265    1286
                    1380    1424    1544    1548    1595    1626    1658    1661    3867    4057    4070    5792    5872
                    5889    5907    5922    5931    5939    5947#   5950#
.BEGIN  003152       692#
.CNVRT  004472       236     957#
.CONVR  004466       234     956#
.DATAC  005552       244    1161#
.DELAY  005436       240    1134#
.EOP    003364       733#   5801
.ERRTA  036316      1048    6046#
.HLT    004750       115    1028#
.INSTE  004154       226     863#
.INSTR  004050       224     842#
.INST1  004070       846#    866
.MSG    004072       844#    847#
.MSTCL  005466       238    1145#
.PARAM  004174       228     874#
.PFAIL  005336       113     480    1109#   1117    5773    5776
.RESOS  004434       232     945#
.ROMCL  005504       242    1150#
.SAVOS  004374       230     931#
.SCOPE  003576       218     781#
.SCOP1  003736       220     813#
.START  002002       134     478#    494     1249
.TIMER  005616       246    1172#
.TRPSR  004716       117    1016#
.TRPTA  001330       216#   1021
.TYPE   003766       222     824#
```

```
DMEND      1#    727
DMFRNT     1#
HLT        2#   1693  1710  1734  1747  1774  1781  1804  1811  1834  1841  1864  1871  1894  1301
             1929  1931  1954  1961  1984  1991  2014  2021  2044  2051  2074  2081  2104  2111  2134
             2141  2164  2171  2194  2201  2224  2231  2256  2270  2299  2313  2342  2355  2383  2391
             2400  2424  2432  2441  2469  2500  2520  2556  2576  2610  2628  2662  2680  2717  2737
             2777  2799  2833  2851  2885  2903  2937  2955  2989  3007  3041  3059  3093  3111  3145
             3163  3197  3215  3257  3275  3309  3327  3366  3389  3433  3453  3484  3486  3512  3514
             3553  3588  3596  3628  3664  3696  3737  3773  3808  3843  3883  3932  3973  4014  4033
             4080  4116  4161  4210  4259  4308  4357  4406  4455  4504  4553  4602  4651  4700  4749
             4798  4847  4896  4945  4994  5043  5092  5141  5190  5239  5288  5337  5386  5435  5484
             5533  5582  5631  5680  5720  5730  5764  5775  5783  5890  5908  5923  5940
$ADD1      1#    1666
$ADD2      1#    1695
$ADD3      1#    1712  2320
$ADD4      1#    1754  1784  1814  1844  1874  1904  1934  1964  1994  2024  2054  2084  2114  2144
             2174  2204
$ALU       1#    4123  4172  4221  4270  4319  4368  4417  4466  4515  4564  4613  4662  4711  4760
             4809  4858  4907  4956  5005  5054  5103  5152  5201  5250  5299  5348  5397  5446  5495
             5544  5593  5642
$ALUO      1#    4085
$AUTO      1#     549
$BR        1#    3298  3315
$BUFFE     1#    1201
$COMP      1#    5716
$CYCLE     1#    1273
$EOP       1#     727
$FINI      1#    6046
$FLOAT     1#    2487  2506  2543  2562  2599  2616  2651  2668  2704  2723  2762  2783  2922  2939
             2874  2891  2926  2943  2978  2995  3030  3047  3082  3099  3134  3151  3186  3203
$GETPA     1#
$HEADE     1#
$INTR      1#    3460  3489
$IR        1#    2362  2403
$MARHI     1#    4056  4069  5871  5930
$MEMFL     1#    3915  3955
$MEMO      1#    3941
$MEM1      1#    3901
$MEM2      1#    3983
$MEM3      1#    4039
$MOCK      1#
$MSG       1#    1189
$NOISE     1#    5787
$NPRBI     1#    3845
$NPR1      1#    3599  3632  3668
$NPR2      1#    3700  3739
$NPR3      1#    3775  3810
$PFAIL     1#    1105
$POWER     1#    5732
$PRIO      1#    3517  3555
$PROC1     1#    2444
$PROC2     1#    2471  2527  2583  2635  2687  2744  2806  2858  2910  2962  3014  3066  3118  3170
$PROC3     1#    3222
$PROC4     1#    3283
$PROC5     1#
$SLEEP     1#    1383  1396  1405  1484  1493
```

```
$RAMCL      1#  1133
$RCLK       1#  1136  1139  1176  1181  2491  2493  2511  2513  2547  2549  2567  2569  2602  2604
            2620  2622  2654  2656  2672  2674  2708  2710  2728  2730  2766  2768  2788  2790  2825
            2827  2843  2845  2877  2879  2895  2897  2929  2931  2947  2949  2981  2983  2999  3001
            3033  3035  3051  3053  3085  3087  3103  3105  3137  3139  3155  3157  3189  3191  3207
            3209  3243  3251  3269  3301  3303  3319  3321  3354  3358  3360  3377  3381  3383  3421
            3425  3427  3445  3447  3480  3508  3544  3584  3621  3653  3657  3659  3689  3722  3730
            3732  3766  3768  3795  3798  3801  3830  3833  3836  3876  3919  3922  3924  3926  3960
            3963  3965  3967  4001  4004  4006  4008  4023  4025  4027  4054  4058  4061  4067  4071
            4074  4104  4106  4108  4110  4149  4153  4155  4198  4202  4204  4247  4251  4253  4296
            4300  4302  4345  4349  4351  4394  4398  4400  4443  4447  4449  4492  4496  4498  4541
            4545  4547  4590  4594  4596  4639  4643  4645  4688  4692  4694  4737  4741  4743  4786
            4790  4792  4835  4839  4841  4884  4888  4890  4933  4937  4939  4982  4986  4988  5031
            5035  5037  5080  5084  5086  5129  5133  5135  5178  5182  5184  5227  5231  5233  5276
            5280  5282  5325  5329  5331  5374  5378  5380  5423  5427  5429  5472  5476  5478  5521
            5525  5527  5570  5574  5576  5619  5623  5625  5668  5672  5674  5722  5808  5821  5824
            5827  5840  5843  5846  5852  5855  5858  5861  5869  5873  5885  5903  5915  5918  5928
            5932  5935  5975  5991  5994  6011  6022  6024  6032  6034
$SCOPE      1#  777
$SIMBC      1#
$SOFTC      1#  1209
$SPF        1#  3350  3372
$SP1        1#  3334
$SP2        1#  3400
$TIMER      1#  5691
$TRPDE      1#  217  219  221  223  225  227  229  231  233  235  237  239  241  243
            245
$TSTN       1#  1673  1701  1720  1762  1792  1822  1852  1882  1912  1942  1972  2002  2032  2062
            2092  2122  2152  2182  2212  2242  2285  2328  2370  2411  2454  2479  2535  2591  2643
            2696  2754  2814  2866  2918  2970  3022  3074  3126  3178  3230  3291  3342  3408  3467
            3496  3525  3563  3606  3639  3675  3708  3747  3783  3818  3853  3908  3948  3991  4047
            4092  4133  4182  4231  4280  4329  4378  4427  4476  4525  4574  4623  4672  4721  4770
            4819  4868  4917  4966  5015  5064  5113  5162  5211  5260  5309  5358  5407  5456  5505
            5554  5603  5652  5700  5749  5798
$VARIA      1#  136
$WRFLT      1#  2249  2262  2292  2305
$WP46       1#  2234  2277
$XZ         1#  1666  1671  1695  1699  1712  1718  1754  1760  1784  1790  1814  1820  1844  1850
            1874  1880  1904  1910  1934  1940  1964  1970  1994  2000  2024  2030  2054  2060  2084
            2090  2114  2120  2144  2150  2174  2180  2204  2210  2234  2240  2277  2283  2320  2326
            2362  2368  2403  2409  2444  2452  2471  2477  2527  2533  2583  2589  2635  2641  2687
            2694  2744  2752  2806  2812  2858  2864  2910  2916  2962  2968  3014  3020  3066  3072
            3118  3124  3170  3176  3222  3228  3283  3289  3334  3340  3400  3406  3460  3465  3489
            3494  3517  3523  3555  3561  3599  3604  3632  3637  3668  3673  3700  3706  3739  3745
            3775  3781  3810  3816  3845  3851  3901  3906  3941  3946  3983  3989  4039  4045  4085
            4090  4123  4131  4172  4180  4221  4229  4270  4278  4319  4327  4368  4376  4417  4425
            4466  4474  4515  4523  4564  4572  4613  4621  4662  4670  4711  4719  4760  4768  4809
            4817  4858  4866  4907  4915  4956  4964  5005  5013  5054  5062  5103  5111  5152  5160
            5201  5209  5250  5258  5299  5307  5348  5356  5397  5405  5446  5454  5495  5503  5544
            5552  5593  5601  5642  5650  5691  5698  5732  5747  5787  5796
```

. ABS. 036522 000

ERRORS DETECTED: 0

```
DZDMC,DZDMC/SQL/CRF+IPLUTL,DZDMC
RUN-TIME:  18 25 1 SECONDS
RUN-TIME RATIO: 1675/46=35.8
CORE USED:  27K  (53 PAGES)
```