IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDVA-B-D
PRODUCT NAME:	BASIC R/W TEST AND ROM INSTRUCTION EXERCISER
DATE RELEASED:	21-APRIL-76
MAINTAINER:	DIAGNOSTICS
AUTHOR:	JOHN EGOLF

1. ABSTRACT

The function of the DV11 diagnostics are to verify that the option operates according to specifications. The diagnostics verfiy that there are no malfunctions and the all operations of the DV11 are correct in its enviroment.
Parameters may be set to alert diagnostics as to the DV11 configuration by using the "TRIAL" program (DZDVE SA:210). All questions should be answered and then each diagnostic will "OVERLAY" these parameters which are stored in the "STATUS TABLE" (see section 8.4a). The alternative to "TRIAL" program is "AUTO SIZING" (see section 8.5).

DZDVA does R/W tests on both primary registers and all secondary registers. Tests are made to verify that no inter-action between secondary registers of any lines exists. DZDVA also exercises all micro-code instructions and verifies internal registers used by the micro processor. Interupts and NPRS are also tested in this diagnostic.
NOTE; FOR EASE OF DIAGNOSIS; ALL (4) LINE CARDS MAY BE PULLED OUT OF THE SYSTEM UNIT AS MAY THE TWO MODEM CONTROL MODULES. ALSO THE DIAGNOSTIC IN NO WAY READS OR USES THE ROMS THAT ARE IN THE DV11.

Currently there are six 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 and insuring that diagnosis of error will be immediate to problem
NOTE! Additional diagnostics may be added in the future.

The six diagnostics are!
1. DZDVA [REV] Basis R/W test and ROM instruction exerciser.
2. DZDVB [REV] static line card tests.
3. DZDVC [REV] 'FREE RUNNING' Rom tests part 1.
4. DZDVD [REV] 'FREE RUNNING' Rom tests part 2.
5. DZDVE [REV] Modem control and cable tests plus manual parameter input. [TRIAL PROGRAM]
6. DZDVF [REV] Asynchronous Line Card Tests.

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (WITH MINIMUM 8K MEMORY)
ASR 33 (or equilivalent)
DV11-AA MUX CNTRL UNIT
AT LEAST ONE OF THE FOLLOWING
DV11-BA 8 LINE SYNC MODULES
DV11-BB 8 LINE ASYNC MODULES
DV11-BC 4 SYNC LINES, 4 ASYNC LINES

2.2 STORAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Location 1500 thru 1736 are especially to be noted and to be untouched by operator after DV11 trial program has been executed; or after the 'AUTO SIZING' has been done.

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 leave
   leave SWR bit 7=1 to use existing parameters set up by DV11 trial program or a previously run DV11 diagnostic that used the 'AUTO SIZING',(section 7.2 and 8.4,8.5 may be helpful)
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 DV11 STATUS'
        1500    175000
        1502    000300
        1504    000226
        1506    000062
        1510    000226
        1512    000062
        1514    000226
        1516    000062
        1520    000226
        1522    000062
```

The above is only an example! This would indicate the status table starting at add. 1500 in the program. 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.

The program will type 'R' and proceed to run the diagnostic

4.1 CONTROL SWITCH SETTINGS

NOTE: If there is no read SWR(177570); SWR may be modified at Loc 176 or by hitting Control "G" <^G> on console terminal.

SW 15	Set:	Halt on error
SW 14	Set:	Loop on current test
SW 13	Set:	Inhibit error print out
SW 12	Set:	Inhibit **ALL** type out/bell on error.
SW 11	Set:	Inhibit iterations. (quick pass)
SW 10	Set:	Escape to next test
SW 09	Set:	Loop with current data
SW 08	Set:	Catch error and loop on it
SW 07	Set:	Use previous status table. CLR=do AUTO SIZE.
SW 06	Set:	Reserved
SW 05	Set:	Reserved
SW 04	Set:	Reserved
SW 03	Set:	Reserved
SW 02	Set:	Lock on selected test
SW 01	Set:	Restart program at selected test
SW 00	Set:	Reselect DV11's desired active.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DV11'S DESIRED ACTIVE. please note that a message is typed out for setting the switch register equal to DV11's active. this means if the system has four DV11s; bits 00,01,02,03 will be set in loc 'DVACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four DV11s 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 DV11s than has been given information about in trial program.

METHOD: A: Load address 200
B: Start with SW 00=1
C: Program will type message
D: Set the binary number of DV11s desired active EXAMPLE: 1=1 DV11; 3=2 DV11; 7=3 DV11; 17=4 DV11 37=5 DV11 etc. PRESS CONTINUE.
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST it is strongly suggested that at least one pass hass been made before trying to select a test that is not in the order of sequence the reason being is that the program has to clear areas and set up parameters. Also when a test is selected ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA: this switch will only work if call 'SCOP1' is in that test. The reason being that most tests deal with blocks of different data to be sent or received all at once thus in block data; one pattern cann't be singled out.

4.1.3 SWITCH REGISTER PRIORITYS

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. SW 09 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is *usually* the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a *HARD* error (constant); SW08 is best.
   (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error.
   (SW14=1, SW10=0, SW09=0, SW08=1,0)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

starting address is at 000200 there are no other starting addresses for the DV11 diagnostics previously mentioned except for DZDVE which is: 000200 for the modem control and cable tests and 000210 for the manual parameter input program.

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after *ALL* available DV11'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 four 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 FUNCTIONING of the test CAN BE INTERPEDITED.

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 to the the error message which is to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the DV11 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 1224)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 DV11 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

DV11 trial program must be run prior to the first and only the first running of any DV11 diagnostic if "AUTO SIZING" is not used.
NOTE: If no program other than a DV11 diagnostic was loaded after DV11 trial or if core memory has not been changed;or if there is no DV11 configuration changes; the DV11 trial program need never be run again. However if any of the above have been violated the DV11 trial program must be run again before running the diagnostics NOTE: An alternative to the above is attempting the 'AUTO SIZING' when program is initially started with SW07=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS (SYNC LINE CARDS ONLY)

1. Hardware must be set to FULL DUPLEX
2. Parity off.
3. All lines of a particular line card must be configured the same.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DV11 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.

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' untill all DV11'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 DZDVA-B CSR: 175000 VEC: 300 PASSES: 000001 ERRORS: 000000
```

NOTE: The numbers for CSR and VEC are not necessarily the values for the device. They are only for this example.
NOTE: DZDVE (MODEM AND CABLE TEST) END PASS message is a large "END" typed out on tty. Please note that each character printed is actually and "END PASS" indication. This was used in place of "BELL" because if sw12=1 and an error occured the BELL may be mistaken for END PASS. The pass execution is so fast that the standard END PASS was too lenghtly. THEREFORE each char is an "END PASS and the entire "END" is not required for acceptance.

8.4 KEY LOCATIONS

```
RETURN (1212)   Contains the address  where  program  will  return  when
                iteration  count  is  reached  or  if  loop  on  test is
                asserted.
NEXT   (1214)   Contains the address of the next test to be peformed.
TSTNO  (1224)   Contains the number of the test now being peformed.
RUN    (1302)   The bit  in  'RUN'  always  points  one  past  the  DV11
                currently      being      tested.      EXAMPLE:      (RUN)
                1302/0000000001000000 Means that DV11 no.05 is the  DV11
                now running.
DVCR00-DVCR17
DVST00-DVST17
(1500)-(1736)
                These locations contain the information needed  to  test
                up  to  8 (decimal) DV11s sequentialy,  they contain the
                CSR,VECTOR and STATUS concerning  the  configuration  of
                each DV11.
DVACTV (1276)   Each  bit  set  in  this  location  indicates  that  the
                associated  DV11  will  be  tested  in  turn.  EXAMPLE:
                (DVACTV)  1276/0000000000011111  means  that  DV11   no.
                00,01,02,03,04   will  be  tested.  EXAMPLE:  (DVACTV)
                1276/0000000000010001 Means that DV11 no. 00,04 will be
                tested.
DVSCR  (1356)   Contains the receiver csr  of  the  current  DV11  under
                test.
L00.03 (1412)
L04.07 (1414)
L08.11 (1416)
L12.15 (1420)   Contains the status of the current DV11 under test.
        BIT 15  Set:    Line card *NOT installed (AND WONT BE TESTED)
        BIT 14  Set:    Reserved
        BIT 13  Set:    Reserved
        BIT 12  Set:    One sync, =0:  two syncs.
        BIT 11  Set:    Async line card, =0 Sync line card.
        BIT 10  Set:    Reserved
        BIT 09   Set:   Bits per char.  (used with bit8)
        BIT 08  Set:    Bits per char.  (used with bit9)
                BIT09 BIT08 BITS PER CHAR.
                  0      0      8
                  0      1      7
                  1      0      6
                  1      1      5
        BIT 07-00       SYNC "A" for specified line card.
                        BITS 07-00 MUST BE ALL ZEROS FOR TESTING  OF  AN
                        ASYNC LINE CARD.
```

8.4A MORE ON THAT 'STATUS TABLE' (1500-1736)

```
'MAP OF DV11 STATUS'
1500    175000
1502    000300
1504    000226
1506    000062
1510    000226
1512    000062
1514    004000
1516    000000
1520    004000
1522    000000
```

SYNC 'A' AND SYNC'B' MUST BE SET TO ZEROS FOR AN ASYNC LINE CARD. The above information will be repeated for each of up to 8 DV11's in the system(these will follow under this table). EXPLANATION:

```
1500    175000  This is the system control register for the 1st DV11  in
                the system.
1502    000300  This is vector 'A' for the first DV11 in the system.
1504    000226  This represents 'SYNC A' and the software status for the
                1st line card in the 1st DV11. The bits are as follows:

        BIT 15  Set:    Line card *NOT installed (AND WONT BE TESTED)
        BIT 14  Set:    Reserved
        BIT 13  Set:    Reserved
        BIT 12  Set:    One sync, =0:  two syncs.
        BIT 11  Set:    Async line card, =0 Sync line card.
        BIT 10  Set:    Reserved
        BIT 09   Set:   Bits per char.  (used with bit8)
        BIT 08  Set:    Bits per char.  (used with bit9)
                BIT09 BIT08 BITS PER CHAR.
                  0     0      8
                  0     1      7
                  1     0      6
                  1     1      5
        BIT 07-00       SYNC 'A' for specified line card.
1506    000062  This represents 'SYNC B' for the 1st line card.
1510    000226  This is 'SYNC A' and line status for the 2nd line  card.
                (for bits defination see explanation for line card 1).
1512    000062  This is 'SYNC B' for the second line card.
1514    000226  This is 'SYNC A' and line status for the 3rd line  card.
                (for bits defination see explanation for line card 1).
1516    000062  This is 'SYNC B' for line card no. 3.
1520    000226  This is 'SYNC A' and line status for the 4th line  card.
                (for bits defination see explanation for line card 1).
1522    000062  This is SYNC B for the 4th line card.
```

The above is repeated for each DV11 in the system. The table is filled by AUTO SIZING or by the manual parameter input program as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The program will start at address 175000 and start 'REFERENCEING' address. If a NON-EX MEMORY TRAP occures; the pointer (holding 175000) is updated by 10 and the above is repeated untill address 175400 is reached. If a 'SLAVE SYNC RESPONSE' was issued by the DV11 (or any other device) (no nxm trap)(and it(SEL0)was=0) ; pointer plus 12 (SEL12) is tested to contain 177777 (MUST BE EXACTLY 177777); if a trap is encountered or if SEL12 does not contain 177777 the above updating is performed. If SEL12 was equal to 177777 the pointer is stored away and the routine continues as above:
NOTE: If the program does not find your DV11; something is wrong and AUTO SIZING should not be done.

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). Bit7 and Bit6 (RX INTERUPT AND RX INTERUPT IE) are set into DVscr register; a delay is made and if no interupt occures (because of a bad DV11) the program assumes vector address 300 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 DV11 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.

8.5.3 PARAMETER ASSUMPTIONS.

Since too much hardware would need to be turned on to SIZE the rest of the parameters; the program must assume the remaining variations. The result if not to your specific configuration may be altered by hang (toggle in) is desired. In this way 95% of the parameter setup was done by the program and 5% by you.
THEREFORE:
1) ALL LINE CARDS(4) ARE ASSUMED TO BE INSTALLED.
   Set Bit15 of status map of any (approiate) line cards missing
2) TWO SYNCS.
   Set Bit12 if you have a 4 line group set for 1 sync.
3) EIGHT BITS PER CHAR.
   Adjust bits 9 and bit 8 in status map for your correct config.
4) SYNCHRONOUS LINE CARDS INSTALLED.
   Set BIT11 for Async line and ZERO sync chars.
5) SYNC "A"=226 AND SYNC "B"=062

In all adjustments please refer to section 8.4a for greater detail.

```
 DOCUMENT
***************
 DZDVAB LST
***************
```

```
2       MAINDEC-11-DZDVA-B/<377>/BASIC DV11 CONTROLLER MODULES TESTING
        COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
        ----------------------------------------------------------------

1119    ROUTINE USED TO "AUTO SIZE" THE DV11
        CSR AND VECTOR.
        NOTE:   THE CSR MAY BE ANY WHERE IN THE FLOATING
                ADDRESS RANGE (175000:175400)
                AND THE VECTOR MAY BE ANY WHERE IN THE
                FLOATING VECTOR RANGE (300:770)


1211    *********************** TEST 1 ******************************
        VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE
        A TIME-OUT TRAP.

1240    *********************** TEST 2 ******************************
        PRIMARY REGISTER ADDRESSING TEST
        LOAD EACH PRIMARY REGISTER WITH A
        DIFFERENT NUMBER AND VERIFY EACH
        WAS INDIVIDUALLY ADDRESSED.

1309    *********************** TEST 3 ******************************
        SYSTEM CONTROL REGSTER READ/WRITE TEST.
        SET BIT2, VERIFY BIT2 WAS SET.
        CLEAR BIT2, VERIFY BIT2 WAS CLEARED.

1335    *********************** TEST 4 ******************************
        SYSTEM CONTROL REGSTER READ/WRITE TEST.
        SET BIT3, VERIFY BIT3 WAS SET.
        CLEAR BIT3, VERIFY BIT3 WAS CLEARED.

1361    *********************** TEST 5 ******************************
        SYSTEM CONTROL REGSTER READ/WRITE TEST.
        SET BIT4, VERIFY BIT4 WAS SET.
        CLEAR BIT4, VERIFY BIT4 WAS CLEARED.

1387    *********************** TEST 6 ******************************
        SYSTEM CONTROL REGSTER READ/WRITE TEST.
        SET BIT5, VERIFY BIT5 WAS SET.
        CLEAR BIT5, VERIFY BIT5 WAS CLEARED.

1413    *********************** TEST 7 ******************************
        SYSTEM CONTROL REGSTER READ/WRITE TEST.
        SET BIT6, VERIFY BIT6 WAS SET.
        CLEAR BIT6, VERIFY BIT6 WAS CLEARED.

1439    *********************** TEST 10 ******************************
        TEST THAT BIT 7(RECEIVER INTERUPT)
        CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
        IS NOT SET.
        THEN VERIFY THAT BIT 7 CAN BE WRITTEN WHEN
        BIT 9 IS SET.
```

```
1472        ************************ TEST 11 ******************************
            SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT8, VERIFY BIT8 WAS SET,
            CLEAR BIT8, VERIFY BIT8 WAS CLEARED,

1498        ************************ TEST 12 ******************************
            SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT9, VERIFY BIT9 WAS SET,
            CLEAR BIT9, VERIFY BIT9 WAS CLEARED,

1524        ************************ TEST 13 ******************************
            SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT10, VERIFY BIT10 WAS SET,
            CLEAR BIT10, VERIFY BIT10 WAS CLEARED,

1550        ************************ TEST 14 ******************************
            SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT12, VERIFY BIT12 WAS SET,
            CLEAR BIT12, VERIFY BIT12 WAS CLEARED,

1576        ************************ TEST 15 ******************************
            SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT13, VERIFY BIT13 WAS SET,
            CLEAR BIT13, VERIFY BIT13 WAS CLEARED,

1602        ************************ TEST 16 ******************************

1603        SYSTEM CONTROL REGISTER READ/WRITE TEST,
            SET BIT3+BIT0, VERIFY BIT3+BIT0 WAS SET,
            CLEAR BIT3+BIT0, VERIFY BIT3+BIT0 WAS CLEARED,

1628        ************************ TEST 17 ******************************
            TEST THAT BIT 15(NPR STATUS OVERFLOW INTERUPT)
            CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
            IS NOT SET,
            THEN VERIFY THAT BIT 15 CAN BE WRITTEN WHEN
            BIT 9 IS SET,

1660        ************************ TEST 20 ******************************
            TEST THAT BIT8 IN DVSCR CLEARS
            BIT7 OF DVSCR,

1682        ************************ TEST 21 ******************************
            RECEIVER INTERUPT CHARACTER REGISTER TEST
            TEST THAT RECEIVER INTERUPT CHARACTER REGISTER CANNOT BE WRITTEN,
            WRITE RECEIVER INTERUPT CHARACTER REGISTER WITH ALL 1'S
            AND VERIFY THAT ALL 0'S ARE READ BACK,

1704        ************************ TEST 22 ******************************
            LINE CONTROL REGISTER READ/WRITE TEST,
            SET BIT9, VERIFY BIT9 WAS SET,
            CLEAR BIT9, VERIFY BIT9 WAS CLEARED,
1732    *********************** TEST 23 ******************************
        LINE CONTROL REGISTER READ/WRITE TEST,
        SET BIT10, VERIFY BIT10 WAS SET,
        CLEAR BIT10, VERIFY BIT10 WAS CLEARED,

1760    *********************** TEST 24 ******************************
        LINE CONTROL REGISTER READ/WRITE TEST,
        SET BIT11, VERIFY BIT11 WAS SET,
        CLEAR BIT11, VERIFY BIT11 WAS CLEARED,

1788    *********************** TEST 25 ******************************
        LINE CONTROL REGISTER READ/WRITE TEST,
        SET BIT12, VERIFY BIT12 WAS SET,
        CLEAR BIT12, VERIFY BIT12 WAS CLEARED,

1816    *********************** TEST 26 ******************************
        LINE CONTROL REGISTER READ/WRITE TEST,
        SET BIT13, VERIFY BIT13 WAS SET,
        CLEAR BIT13, VERIFY BIT13 WAS CLEARED,

1844    *********************** TEST 27 ******************************
        LINE CONTROL REGISTER READ/WRITE TEST,
        SET BIT14, VERIFY BIT14 WAS SET,
        CLEAR BIT14, VERIFY BIT14 WAS CLEARED,

1872    *********************** TEST 30 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST,
        SET BIT0, VERIFY BIT0 WAS SET,
        CLEAR BIT0, VERIFY BIT0 WAS CLEARED,

1898    *********************** TEST 31 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST,
        SET BIT1, VERIFY BIT1 WAS SET,
        CLEAR BIT1, VERIFY BIT1 WAS CLEARED,

1924    *********************** TEST 32 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST,
        SET BIT2, VERIFY BIT2 WAS SET,
        CLEAR BIT2, VERIFY BIT2 WAS CLEARED,

1950    *********************** TEST 33 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST,
        SET BIT3, VERIFY BIT3 WAS SET,
        CLEAR BIT3, VERIFY BIT3 WAS CLEARED,

1976    *********************** TEST 34 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST,
        SET BIT8, VERIFY BIT8 WAS SET,
        CLEAR BIT8, VERIFY BIT8 WAS CLEARED,
2002    *********************** TEST 35 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST.
        SET BIT9, VERIFY BIT9 WAS SET.
        CLEAR BIT9, VERIFY BIT9 WAS CLEARED.

2028    *********************** TEST 36 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST.
        SET BIT10, VERIFY BIT10 WAS SET.
        CLEAR BIT10, VERIFY BIT10 WAS CLEARED.

2054    *********************** TEST 37 ******************************
        SECONDARY REGISTER SELECTOR READ/WRITE TEST.
        SET BIT11, VERIFY BIT11 WAS SET.
        CLEAR BIT11, VERIFY BIT11 WAS CLEARED.

2080    *********************** TEST 40 ******************************
        SECONDARY REGISTER ACCESS REG. READ/WRITE TEST
        SET EACH INDIVIDUAL BIT;  VERIFY EACH INDIVIDUAL BIT SET.
        CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.

2117    *********************** TEST 41 ******************************
        SPECIAL FUNCT. REGISTER READ/WRITE TEST
        SET EACH INDIVIDUAL BIT;  VERIFY EACH INDIVIDUAL BIT SET.
        CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.

2155    *********************** TEST 42 ******************************

2157    NPR STATUS REG. TEST
        TEST THAT NPR STATUS REG. CANNOT BE WRITTEN
        READ THE NPR STATUS REG. AND STORE THE DATA;
        COMPLEMENT THE DATA AND  WRITE THE NPR STATUS REG.
        VERIFING THAT THE NPR STATUS REG. DID NOT CHANGE.

2181    *********************** TEST 43 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST.
        SET BIT0, VERIFY BIT0 WAS SET.
        CLEAR BIT0, VERIFY BIT0 WAS CLEARED.

2207    *********************** TEST 44 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST.
        SET BIT1, VERIFY BIT1 WAS SET.
        CLEAR BIT1, VERIFY BIT1 WAS CLEARED.

2233    *********************** TEST 45 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST.
        SET BIT2, VERIFY BIT2 WAS SET.
        CLEAR BIT2, VERIFY BIT2 WAS CLEARED.

2259    *********************** TEST 46 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST.
        SET BIT3, VERIFY BIT3 WAS SET.
        CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
```

```
2285    *********************** TEST 47 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST,
        SET BIT4, VERIFY BIT4 WAS SET,
        CLEAR BIT4, VERIFY BIT4 WAS CLEARED,

2311    *********************** TEST 50 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST,
        SET BIT5, VERIFY BIT5 WAS SET,
        CLEAR BIT5, VERIFY BIT5 WAS CLEARED,

2337    *********************** TEST 51 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST,
        SET BIT6, VERIFY BIT6 WAS SET,
        CLEAR BIT6, VERIFY BIT6 WAS CLEARED,

2363    *********************** TEST 52 ******************************
        DV11 RESERVED REGISTER READ/WRITE TEST,
        SET BIT7, VERIFY BIT7 WAS SET,
        CLEAR BIT7, VERIFY BIT7 WAS CLEARED,

2389    *********************** TEST 53 ******************************
        TEST OF THE BYTE OPERATIONS FOR THE DV11
        SYSTEM CONTROL REG AND THE SECONDARY REG SEL,
        THE TEST WILL CLEAR DVSCR AND THE WRITE (LOW BYTE)
        BIT3; THEN VERIFY ONLY BIT3 IS SET; THEN THE
        TEST WILL WRITE BIT 8(HIGH BYTE) AND VERFIY THAT
        BIT8 AND BIT3 ARE SET,  THE EXACT PROCEEDURE
        WILL BE USED ON THE DVSRS REGISTER,

2434    *********************** TEST 54 ******************************
        SECONDARY REGISTER READ/WRITE TESTS
        READ/WRITE TEST, READ AND WRITE DIFFERENT DATA
        PATTERENS INTO THE SECONDARY REGISTERS VERIFING
        THAT WHAT WAS READ MATCHES WHAT WAS WRITTEN,

2493    *********************** TEST 55 ******************************
        INDIVIDUAL LINE DUEL ADDRESS TESTS
        THIS TEST VERIFIES THAT WRITING ONE SECONDARY
        REGISTER FOR A SPECIFIC LINE DOES NOT ALTER
        ANY OTHER SECONDARY REGISTER FOR THAT LINE,

2536    *********************** TEST 56 ******************************
        VERIFY NO LINE INTERACTION,
        THIS TEST VERIFIES THAT WRITING THE SECONDARY
        REGISTERS FOR ONE LINE DOES NOT INTERFEAR WITH
        THE SECONDARY REGISTERS OF ANOTHER LINE,

2586    PART 2
        FILL ALL RAMS WITH ALL 1'S AND THEN
        CLEAR JUST ONE BIT AT A TIME VERIFYING
        THAT ONLY THAT ONE BIT IS CLEAR AND THAT
        ALL OTHER RAMS STILL CONTAIN ALL 1'S,
        THERE SHOULD BE ONLY ONE BIT CLEARED AT
        ONE TIME,
2647      ************************ TEST 57 ******************************
          MEMORY EXTENSION READ/WRITE TEST
          VERIFY BITS 4 AND 5 OF EACH LINE
          SECONDARY REGISTERS EXERCISED ARE:
           00      TX BUS ADDRESS (PRIMARY)
           02      TX BUS ADDRESS (SECONDARY)
           04      RX BUS ADDRESS
           10      TX TABLE BASE ADDRESS
           11      RX TABLE BASE ADDRESS
          NOTE THAT ALL LINES (00-16) ARE EXERCISED.

2705      ************************ TEST 60 ******************************
          INITIALIZATION TESTS
          SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
          AND VERIFY THAT ALL THE BITS ARE CLEARED
          BY A BUS RESET

2771      ************************ TEST 61 ******************************
          INITIALIZATION TESTS
          SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
          AND VERIFY THAT ALL THE BITS ARE CLEARED
          BY A MASTER CLEAR

2836      ************************ TEST 62 ******************************
          ATTACK OF THE SPECIAL FUNCTIONS REGISTER.
          BEGIN CHECK OF THE DVSFR.
          SUMARY OF PROC. INSTRUCTIONS

          BIT14 BIT13 BIT12  INSTRUCTION
            0     0     0    BRANCH "A"
            0     0     1    ALU OPERATION
            0     1     0    RAM OPERATION
            0     1     1    DATA TRANSFER
            1     0     0    NPR OPERATION
            1     0     1    SET/CLEAR OPERATION
            1     1     0    BCC CALCULATION
            1     1     1    BRANCH "B"


2853      ************************ TEST 62 ******************************
          VERIFY THAT "ROM STEP"
          IS SELF-CLEARING AND THAT
          THE DATA IN THE DVSFR
          IS CHANGED WHEN THE ROM IS STEPPED.

2882      ************************ TEST 63 ******************************
          BASIC   TEST OF DVSFR
          TEST THAT "BRANCH A" INSTRUCTION.
          POINTS TESTED:

          BIT11  BIT10  BIT09  BIT08  BR "A"  BR "B"  FUNCTION
            0      0      0      1       L       H   PLUS 3 VOLTS
            0      1      0      1       L,H     H,H DVSCR08 (=0,=1)
            0      1      1      1       H       H   NPR SILO NOT AVAIL.
```

```
                1      1      1      1       H      H   SILO FULL
                Ø      1      1      Ø       H      H       NXM

        2960    *********************** TEST 64 ******************************
                TEST OF BRANCH B"
                TEST THAT POINT 16 (GROUND)
                MAKES LCR BIT1=1 AND BITØ=1.

        2982    *********************** TEST 65 ******************************
                TEST OF BRANCH B
                CHECKING "DEFAULT" STATES OF THE DV11 SIGNALS.
                BIT11   BIT1Ø   BITØ9   BITØ8   FUNCTION
                  1       Ø       Ø       Ø     DATA NOT AVAIL.
                  1       Ø       Ø       1     REQUEST BUS
                  1       Ø       1       Ø     MEMORY PARITY ERROR
                  1       1       1       1     WRITE INHIBIT

        3Ø4Ø    *********************** TEST 66 ******************************
                BASIC TEST OF THE
                "SET/CLEAR INSTRUCTION.
                TEST THAT THE SET/CLEAR CAN DO:
                CLEAR DVSCR Ø8
                SET DVSCR1Ø
                SET RECEIVER INTERUPT (DVSCRØ7)

        3Ø91    *********************** TEST 67 ******************************
                BASIC TEST OF THE
                "SET/CLEAR INSTRUCTION.
                TEST THAT THE SET/CLEAR CAN:

                BIT 14  BIT12  BITØ3  BITØ2  BITØ1  BITØØ  FUNCTION
                  1        1      1     Ø      Ø      Ø       SET RICR 15
                  1       1       1     Ø      Ø      1     SET RICR 14
                  1       1       1     1      Ø      Ø     SET RICR 13
                  1       1       1     1      Ø      1     SET RICR 12


        316Ø    *********************** TEST 7Ø ******************************
                BASIC TEST OF DVSFR.

        3162    TEST OF "SET/CLEAR" AND
                "BRANCH A" AND "BRANCH B" FUNCTIONS.

        3175    TEST SET/CLEAR FUNCTION
                FOR ALU BITØ2

        3187    TEST SET/CLEAR FUNCTION
                FOR RAM OUTPUT BITØØ
```

```
3200    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT01

3213    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT02

3226    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT03

3239    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT04

3252    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT05

3265    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT06

3278    TEST SET/CLEAR FUNCTION
        FOR RAM OUTPUT BIT07

3304    *********************** TEST 71 ******************************
        TEST OF "RECEIVER CHARACTER SILO"
        THRU THE USE OF THE DVSFR REG,
        TEST THE FILLING THE SILO PRODUCES "SILO FULL"
        ON EXACTLY THE 128 LOAD,
        SET/CLEAR IS USED TO STUFF SILO AND BRANCH A IS USED TO TEST SILO.
        SET/CLEAR "SILO IN" AND SET/CLEAR "SILO OUT" ARE EXERCISED TOO,

3362    *********************** TEST 72 ******************************
        TEST THAT AFTER AN INIT
        THAT "RECV CHARACTER WAITING"
        IS FALSE (HIGH) AND THEN VERIFY
        THAT WHEN "SILO IN" IS ASSERTED THAT
        THAT "RCVED CHARACTER WAITING" IS TRUE (LOW)
        AND MAKES "BRANCH A" TRUE.

3401    TEST THAT SETTING DVSCR07
        INHIBITS RCV CHAR WAITING FROM APPEARING
        TRUE; AND THAT CLEARING
        DVSCR07 MAKES IT APPEAR TRUE AGAIN,

3421    *********************** TEST 73 ******************************
        BASIC TEST OF THE "DATA TRANSFER INSTRUCTION"
        BITS 07,06,05,04 OF DVSFR INDICATE THE SOURCE
        BITS 03,02,01,00 OF DVSFR INDICATE THE DESTINATION,

3436    TEST TO XFR SOURCE REGISTERS TO THE DVRIC
        REGISTER VERIFYING THAT THE FOLLOWING REGISTERS
        ARE CLEARED AND THAT THE XFR BUS IS CLEAR AFTER
        A MSTCLR,
        REGISTER        FUNCTION
         0000           GROUND
```

```
3442    0001          GROUND
        0010          GROUND
        0011          GROUND
        0100          GROUND
        0101          MASTER SCAN 0-3/0-3
        0110          ALU RESULT 8-11/0-3
        0111          ALU RESULT 5-7/0-2
        1000          LOW BYTE=B REG 8-15 ; HIGH BYTE=GRND
        1001          LO BYTE=NPR OUT ; HI BYTE=CDC REG
        1010          RAM OUTPUT 0-2/8-10
        1011          RAM OUTPUT
        1101          NPR INPUT REGISTER
        1110          BCC REGISTER
        1111          ALU RESULT REGISTER

3474    TEST OF SET RAM OUTPUT BIT0
        AND THE USE OF THE DATA XFER INSTR.
        PLACE RAM BIT0 INTO THE DVRIC REG

3484    TEST TO SET RAM OUTPUT DATA BIT3
        AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
        TO PLACE BIT3 INTO THE DVRIC REGISTER.

3495    TEST TO SET RAM OUTPUT DATA BIT4
        AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
        TO PLACE BIT4 INTO THE DVRIC REGISTER.

3506    TEST TO SET RAM OUTPUT DATA BIT7
        AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
        TO PLACE BIT7 INTO THE DVRIC REGISTER.

3539    *********************** TEST 74 ******************************
        BASIC TEST OF THE "ALU OPERATION" INSTRUCTION.
        FIRST PART:ISSUE AN INIT AND MOVE
        THE ALU RESULT REGISTER TO THE DVRIC
        REGISTER VERIFING THAT IT IS ZERO.
        SECOND PART: DO A FUNCTION "F=A"
        THEN MOV "F" TO THE DVRIC REGISTER VERIFING IT TO
        BE ZERO
        THIRD PART: DO A FUNCTION "F=A+B"; MOVING
        "F" TO DVRIC AND MAKING SURE IT IS ZERO.
        THUS THE FOLLOWING HAS BEEN TESTED:
        ALU RESULT,"A" REG,AND "B" REG ALL ZEROED ON INIT.

3583    *********************** TEST 75 ******************************
        TEST OF ALU OPERATIONS.
        TEST OF ALL ALU OPERATIONS USED BY DV11.
        FUNCTIONS TESTED:(NOTE THAT "F" IS ALU RESULT)
        DVSFR BITS:
        BIT12 BIT05 BIT04 BIT03 BIT02 BIT01 BIT00 FUNCTION
          1     0     1     1     1     0     0   F=-1
          1     0     0     1     1     0     0   F=0
          1     0     1     1     1     1     1   F=A
          1     0     0     0     1     0     1   F=B
```

```
              1     1     1     1     1     1     1    F=A+1
              1     0     1     0     1     1     0    F=A+B

      3601    FUNCTION TESTED
              F=-1,RIC_F


      3618    TEST THAT DVRIC (NOW THAT ITS ALL 1'S)
              CAN BE CLEARED BY A MSTCLR.


      3628    NEXT SET OF FUNCTIONS:
              F=0,RIC_F


      3642    NEXT SET OF FUNCTIONS:
              F=A+1,RIC_F,A_F,F=0,F=A


      3676    NEXT SET OF FUNCTIONS:
              F=A+1,A_F,B_F,F=A+B,RIC_F


      3707    NEXT SET OF FUNCTIONS:
              F=-1,B_F,F=0,F=B


      3729    NEXT SET OF FUNCTIONS:
              CATCH "SET/CLEAR" THAT WAS MISSED.
              F=-1,S/C[ALU01=0],RIC_F


      3748    NEXT SET OF FUNCTIONS:
              CATCH ANOTHER "SET/CLEAR" THAT WAS MISSED.
              F=-1,S/C[ALU HIGH BYTE=0],RIC_F


      3770    *********************** TEST 76 ******************************
              MASTER SCANNER TEST.
              VERIFY FIRST THAT THE MASTER SCANNER
              IS CLEARED BY INIT.
              VERIFY SECONDLY THAT THE MASTER SCANNER
              CAN BE INCREMENTED FROM 0 THRU 17 BACK TO 0.

      3810    *********************** TEST 77 ******************************
              BASIC TESTS OF THE "RAM OPERATION" INSTRUCTION.
              VERIFY THE READ PORTION OF THE RAM OPERATION.
              LOAD ALL SECONDARY REGISTERS OF ALL LINES
              WITH DIFFERENT NUMBERS AND VERIFY THAT THE RAM OPERATION
              CAN READ THE CORRECT SEC. REG. INTO THE DVRIC REG.
```

```
3870   *********************** TEST 100 ******************************
       TEST OF BRANCH A TEST POINTS
       THAT WERE PREVIOUSLY SKIPPED BECAUSE
       FUNCTIONS:
       BIT11   BIT10   BIT09   BIT08   FUNCTION
         0       0       0       0     ALU 15=1,0
         1       0       0       0     ALU 13-  =1,0
         1       0       0       0           -12=1,0
         1       0       1       0     ALU 00=1,0
         1       0       1       1     ALU 01=1,0
         1       1       0       0     ALU 02=1,0
         1       1       0       1     ALU 03=1,0
         1       1       1       0     ALU 04=1,0

3891   BRANCH "A" TEST OF ALU 15


3919   BRANCH "A" TEST OF ALU 13-


3947   BRANCH "A" TEST OF ALU   -12


3975   BRANCH "A" TEST OF ALU 00


4003   BRANCH "A" TEST OF ALU 01


4031   BRANCH "A" TEST OF ALU 02


4059   BRANCH "A" TEST OF ALU 03


4087   BRANCH "A" TEST OF ALU 04


4118   *********************** TEST 101 ******************************
       TEST OF BRANCH "B" "RAM OUTPUT 0-14=0".
       TEST TO A RAM READ AND "FLOAT" A "1" FROM
       RAM 0 TO 14 ; EXPECTING "RAM 014=0" TO BE FALSE.
       THEN THE "1" IS SHIFTED INTO BIT15 AND
       "RAM 0-14=0" SHOULD BE FALSE.
       THIS ALSO TEST "BRB" [RAM OUTPUT BIT15] TRUE.

4169   *********************** TEST 102 ******************************

4170   TEST OF THE RAM WRITE OPERATION.
       WRITE ALL SECONDARY REGISTERS FOR ALL LINES
       WITH DIFFERENT DATA BY USING THE ROM
       AND VERIFY THE DATA BY THE UNIBUS.
```

```
4272     *********************** TEST 103 ******************************
         BASIC TEST FOR THE "BCC OPERATION"
         POLYNOMIAL SELECTION TABLE:
         RAM OUTPUT BIT04 BIT03  POLY
                       0     0          LRC 8
                       0     1          CRC 16
                     1     1            CRC CCITT

4281     *********************** TEST 103 ******************************

4282     TEST OF LRC 8,
         PATTERNS ARE:
         A REG  B REG  BCC (EXPECTED)
           0'S    0'S   0'S
           0'S    1'S   1'S
           1'S    0'S   1'S
           1'S    1'S   0'S

4353     *********************** TEST 104 ******************************
         TEST OF POLYNOMIAL "CRC 16"
         TEST THAT BITS 9-13 OF THE "B" REG APPEAR
         IN BITS 1-5 OF THE BCC REG,

4401     *********************** TEST 105 ******************************
         TEST OF THE BCC OPERATION USING
         USING CRC16 FOR THE POLYNOMIAL
         SPECIFIC DATA PATTERNS ARE USED TO
         ISOLATE FAULTS AS SOON AS POSSIBLE

4683     *********************** TEST 106 ******************************
         TEST TO RUN A BINARY COUNT (000-377)PATTERN
         THROUGH THE BCC GENERATION LOGIC,
         THE POLYNOMIAL USED WILL BE LRC8 ,
         THE BCC REGISTER WILL BE BUILT UP AFTER
         EACH CHARACTER -*NOT ZEROED OUT*-

4728     *********************** TEST 107 ******************************
         TEST TO RUN A BINARY COUNT (000-377)PATTERN

4730     THROUGH THE BCC GENERATION LOGIC,
         THE POLYNOMIAL USED WILL BE CRC16 ,
         THE BCC REGISTER WILL BE BUILT UP AFTER
         EACH CHARACTER -*NOT ZEROED OUT*-

4773     *********************** TEST 110 ******************************
         TEST TO RUN A BINARY COUNT (000-377)PATTERN
         THROUGH THE BCC GENERATION LOGIC,
         THE POLYNOMIAL USED WILL BE CRC,CCITT ,
         THE BCC REGISTER WILL BE BUILT UP AFTER
         EACH CHARACTER -*NOT ZEROED OUT*-
```

```
4818    *********************** TEST 111 ******************************
        TEST THAT SETTING BIT9!BIT7 AND BIT9!BIT6
        RECV IE AND RECV INTR PRODUCE AN INTERUPT ON VECTOR "A"

4858    *********************** TEST 112 ******************************
        TEST THAT SETTING BIT12 AND BIT10
        STORE IE AND NPR STAT OVFLOW PRODUCE AN INTERUPT ON VECTOR "B"

4898    *********************** TEST 113 ******************************
        TEST THAT SETTING BIT15!BIT9 AND BIT13!BIT9
        NPR STAT INTR AND NPR STAT IE PRODUCE AN INTERUPT ON VECTOR "B"

4939    *********************** TEST 114 ******************************
        TEST TO VERIFY THAT VECTOR "A"
        OCCURES BEFOR VECTOR "B" EVEN
        WHEN VECTOR "B" IS ENABLED BEFORE
        VECTOR "A".

4980    *********************** TEST 115 ******************************
        PRIORITY INTERUPT TESTS.
        SET PS TO PRIORITY 7 AND VERIFY
        THAT THE DV11 DOESN'T INTERUPT.

5005    *********************** TEST 116 ******************************
        PRIORITY INTERUPT TESTS.
        SET PS TO PRIORITY 6 AND VERIFY
        THAT THE DV11 DOESN'T INTERUPT.

5030    *********************** TEST 117 ******************************
        PRIORITY INTERUPT TESTS.
        SET PS TO PRIORITY 5 AND VERIFY
        THAT THE DV11 DOESN'T INTERUPT.

5055    *********************** TEST 120 ******************************
        PRIORITY INTERUPT TESTS.
        SET PS TO PRIORITY 4 AND VERIFY
        THAT THE DV11 DOES INTERUPT.

5081    *********************** TEST 121 ******************************
        TEST THAT BIT15(NPR STATUS INTR) WILL
        SET WHEN AN ENTRY IS MADE INTO THE
        NPR STATUS REGISTER.
        ALSO VERIFY THAT READING THE DVNSR CLEARS DVSCR BIT15.

5113    *********************** TEST 122 ******************************
        TEST TO WRITE PATTERNS THROUGH
        THE NPR STATUS REGISTER.
        BITS WRITTEN: 11,10,09,08,03,02,01,00
        (WHEN BIT 15 OF DVSCR IS SET SO SHOULD BIT 15 OF DVNSR)
```

```
5159      *********  FIRST PLANNED ATTEMPT *********
          *********      TO EXECUTE NPR,   *********
          -----------------------------------------

5163      *********************** TEST 123 ******************************
          BASIC TEST OF THE NPR OPERATION INSTRUCTION,
          TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION
          VIA THE NPR LOGIC,
          LOACATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
          PLACED INTO IT AND READ INTO THE DV11 AND XFERED
          INTO THE DVRIC REGISTER,
          NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION

5206      *********************** TEST 124 ******************************
          BASIC TEST OF THE NPR OPERATION INSTRUCTION,
          TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION

5209      VIA THE NPR LOGIC,
          LOACATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
          PLACED INTO IT AND READ INTO THE DV11 AND XFERED
          INTO THE DVRIC REGISTER,
          NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION,

5250      *********************** TEST 125 ******************************
          BASIC TEST OF THE NPR OPERATION INSTRUCTION,
          TEST THAT THE DV11 CAN "WRITTEN" INTO CORE LOCATION
          VIA THE NPR LOGIC,
          LOACATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
          WRITTEN INTO IT BY THE DV11 NPR LOGIC,
          NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION

5296      *********************** TEST 126 ******************************
          BASIC TEST OF THE NPR OPERATION INSTRUCTION,
          TEST THAT THE DV11 CAN "WRITTEN" INTO CORE LOCATION
          VIA THE NPR LOGIC,
          LOACATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
          WRITTEN INTO IT BY THE DV11 NPR LOGIC,
          NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION,

5343      *********************** TEST 127 ******************************
          BASIC TEST OF THE NPR OPERATION INSTRUCTION,
          TEST THAT THE DV11 CAN DO AN NPR
          TO A NON-EXISTANT MEMORY,
          TEST THAT BRANCH "A" -NXM H- IS SET AFTER
          THE NPR, THEN DO A "SET/CLEAR" CLEAR NXM
          AND VERIFY THAT IT IS CLEARED,
                          ;*MAINDEC-11-DZDVA-B/<377>/BASIC DV11 CONTROLLER MODULES TESTING
                          ;*COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                          ;*-------------------------------------------------------------------

                                  ;STARTING PROCEDURE
                                  ;LOAD PROGRAM
                                  ;LOAD ADDRESS 000200
                                  ;PRESS START
                                  ;PROGRAM WILL TYPE "MAINDEC-11-DZDVA-B/<377>/BASIC DV11 CONTROLLER MODULES TESTI
                                  ;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


                          ;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, DO "AUTO SIZING" ON INITAL START UP,
          000100          SW06=100
          000040          SW05=40
          000020          SW04=20
          000010          SW03=10
          000004          SW02=4                  ;LOCK ON TEST SELECT
          000002          SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
          000001          SW00=1                  ;RESELECT DV11 DESIRED ACTIVE
                                                  ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

```


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

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

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

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

          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
                          ;-------------------------------
          010000          ALU=BIT12
          020000          RAM=BIT13
          030000          XFR=BIT13+BIT12
          040000          NPR=BIT14
          050000          S.C=BIT14+BIT12
          060000          BCC=BIT14+BIT13
          070000          BRB=BIT14+BIT13+BIT12
                          ;-------------------------------


```

```
                                        ;;*******************************************************************
                                        ;-------------------------------------------------------------------
                                                ;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   004402                                .PFAIL                  ;POWER FAIL HANDLER
 000026   000340                                340                     ;SERVICE AT LEVEL 7
 000030   004002                                .HLT                    ;ERROR HANDLER
 000032   000340                                340                     ;SERVICE AT LEVEL 7
 000034   003750                                .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
 000036   000340                                340                     ;SERVICE AT LEVEL 7
          000040                        .=40
 000040   000001                                .BLKW 1                 ;SAVE FOR ACT-11 OR DDP2
 000042   000001                                .BLKW 1                 ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
 000044   000001                                .BLKW 1                 ;SAVE FOR ACT-11 OR DDP2
 000046   002560                                LOGICAL                 ;FOR USE WITH ACT-11 OR DDP2

          000174                        .=174
 000174   000000                        LIGHT:  0
          000176                        .=176
 000176   000000                        SSWR:   0

          000200                        .=200
 000200   000137  001742                        JMP     .START          ;GO TO START OF PROGRAM


          001000                        .=1000
 001000   005377  040515  047111        MTITLE: .ASCIZ  <377><12>/MAINDEC-11-DZDVA-B/<377>/BASIC DV11 CONTROLLER MODULES TESTING
(2)
          001200                        .=1200
 001200                                 LIGHTS:
 001200   177570                                177570
 001202   177570                        SWR:    177570
                                                ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                                ;-------------------------------------------------

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

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

 001214   000000                        RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
 001216   000000                        NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
 001220   000000                        LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
```

```
 001222   000003                        ICOUNT: 3                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
 001224   000000                        LPCNT:  0                       ;NUMBER OF ITEREATIONS COMPLETED
 001226   000000                        TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
 001230   000000                        PASCNT: 0                       ;NUMBER OF PASSES COMPLETED
 001232   000000                        ERRCNT: 0                       ;TOTAL NUMBER OF ERRORS
 001234   000000                        LSTERR: 0                       ;PC OF LAST ERROR CALL

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

 001236   000000                        STAT:   0                       ;DV STATUS WORD STORAGE
 001240   000000                        SYNCX:  0
 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   000001                        DVACTV: .BLKB 1                 ;DV11'S SELECTED ACTIVE.
 001301   000001                        DVNUM:  .BLKB 1                 ;OCTAL NUMBER OF DV11'S.
 001302   000001                        SAVACT: .BLKB 1                 ;ORIGINAL ACTV.  DEVICES.
 001303   000001                        SAVNUM: .BLKB 1                 ;WORKABLE NUMBER.
 001304   000001                        RUN:    .BLKB 1                 ;POINTER ONE PAST RUNNING DEVICE.
          001306                        .EVEN
 001306   001500                        CREAM:  DV.MAP                  ;TABLE POINTER.
```

```

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

001310     000             INIFLG: .BYTE   0             ;PROGRAM INITIALIZATION FLAG
001311     000             ERRFLG: .BYTE   0             ;ERROR OCCURED FLAG
001312     000             LOKFLG: .BYTE   0             ;LOCK ON CURRENT TEST FLAG
001313     000             QV.FLG: .BYTE   0             ;QUICK VERIFY FLAG.
                                                         ;ON FIRST PASS OF EACH DV11 ITERATIONS WILL BE SUPPRESSE
                           .EVEN
        000000             $Y=0

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

                           ;;*****************************************************************
                           ;-----------------------------------------------------------------
001314                     .TRPTAB:
        104400             SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001314  002634                     .SCOPE
        104401             SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001316  003020                     .SCOP1
        104402             TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001320  003044                     .TYPE
        104403             INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
001322  003120                     .INSTR
        104404             INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
001324  003224                     .INSTER
        104405             PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001326  003244                     .PARAM
        104406             SAV05=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
001330  003444                     .SAV05
        104407             RES05=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
001332  003504                     .RES05
        104410             CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
001334  003536                     .CONVRT
        104411             CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001336  003542                     .CNVRT
        104412             MSTCLR=TRAP+12          ;CALL TO ISUE A MASTER CLEAR
001340  004556                     .MSTCLR
        104413             RAMCLR=TRAP+13          ;CALL TO CLEAR THE RAMS
001342  004516                     .RAMCLR
        104414             DELAY=TRAP+14           ;CALL TO VARIABLE DELAY COUNTER
001344  004476                     .DELAY
        104415             ROMCLK=TRAP+15          ;CALL TO CLOCK ROM ONCE
001346  004566                     .ROMCLK
        104416             DATACLK=TRAP+16         ;CALL TO CLK DATA
001350  004576                     .DATACLK

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

```
                                   ;DV11 VECTOR AND REGISTER INDIRECT POINTERS

001352  000000             DVRVEC: 0               ;POINTER TO DV11 RECEIVER INTERRUPT VECTOR
001354  000000             DVRLVL: 0               ;POINTER TO DV11 RECEIVER INTERRUPT SERVICE PS
001356  000000             DVTVEC: 0               ;POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR
001360  000000             DVTLVL: 0               ;POINTER TO DV11 TRANSMITTER INTERRUPT SERVICE PS
001362  000000             DVSCR:  0               ;POINTER TO DV11 SYSTEM CONTROL REGISTER
001364  000000             DVSCRH: 0               ;POINTER TO DV11 SYSTEM CONTROL REGISTER HIGH BYTE.
001366  000000             DVRIC:  0               ;POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER
001370  000000             DVLCR:  0               ;POINTER TO DV11 LINE PRAMETER REGISTER
001372  000000             DVSRS:  0               ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
001374  000000             DVSRSH: 0               ;POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.
001376  000000             DVSRA:  0               ;POINTER TO DV11 SECONDARY REGISTER ACCESS REGISTER
001400  000000             DVSFR:  0               ;POINTER TO DV11 SPECIAL FUNCTIONS REGISTER
001402  000000             DVNSR:  0               ;POINTER TO DV11 NPR STATUS REGISTER
001404  000000             RESV16: 0               ;POINTER TO RESERVED REGISTER.


                                   ;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST
                                   ;-------------------------------------------------

001406     000             MASK.A: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 00-03
001407     000             MASK.B: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 04-07
001410     000             MASK.C: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 08-11
001411     000             MASK.D: .BYTE 000       ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 12-15

001412     010             CLK.A:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
001413     010             CLK.B:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
001414     010             CLK.C:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
001415     010             CLK.D:  .BYTE 8.        ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15

001416  000000             L00.03: 000000          ;PARAMETERS FOR LINES 00-03
001420  000000             L04.07: 000000          ;PARAMETERS FOR LINES 04-07
001422  000000             L08.11: 000000          ;PARAMETERS FOR LINES 08-11
001424  000000             L12.15: 000000          ;PARAMETERS FOR LINES 12-15

001426  000000             SYNC2A: 000000          ;SYNC 2
001430  000000             SYNC2B: 000000          ;
001432  000000             SYNC2C: 000000          ;
001434  000000             SYNC2D: 000000          ;

                                           ;SUMMARY
                                           ;-------
                           ;       MASK.X          040     5 BITS PER CHAR.
                           ;                       100     6 BITS PER CHAR.
                           ;                       200     7 BITS PER CHAR.
                           ;                       000     8 BITS PER CHAR.

                           ;       CLK.X           005     5 BITS PER CHAR.
                           ;                       006     6 BITS PER CHAR.
                           ;                       007     7 BITS PER CHAR.
                           ;                       010     8 BITS PER CHAR.
```

```
                                  ;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                  ;----------------------------------------

         001500                   .=1500
001500                            DV.MAP:
001500   000001                   DVCR00: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
001502   000001                   DVTR00: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 00
001504   000001                   DV00.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
001506   000001                   SYNA00: .BLKW 1         ;SYNC TWO
001510   000001                   DV00.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
001512   000001                   SYNB00: .BLKW 1         ;SYNC TWO
001514   000001                   DV00.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
001516   000001                   SYNC00: .BLKW 1         ;SYNC TWO
001520   000001                   DV00.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
001522   000001                   SYND00: .BLKW 1         ;SYNC TWO

001524   000001                   DVCR01: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
001526   000001                   DVTR01: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 01
001530   000001                   DV01.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
001532   000001                   SYNA01: .BLKW 1         ;SYNC TWO
001534   000001                   DV01.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
001536   000001                   SYNB01: .BLKW 1         ;SYNC TWO
001540   000001                   DV01.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
001542   000001                   SYNC01: .BLKW 1         ;SYNC TWO
001544   000001                   DV01.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
001546   000001                   SYND01: .BLKW 1         ;SYNC TWO

001550   000001                   DVCR02: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
001552   000001                   DVTR02: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 02
001554   000001                   DV02.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
001556   000001                   SYNA02: .BLKW 1         ;SYNC TWO
001560   000001                   DV02.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
001562   000001                   SYNB02: .BLKW 1         ;SYNC TWO
001564   000001                   DV02.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
001566   000001                   SYNC02: .BLKW 1         ;SYNC TWO
001570   000001                   DV02.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
001572   000001                   SYND02: .BLKW 1         ;SYNC TWO

001574   000001                   DVCR03: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
001576   000001                   DVTR03: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 03
001600   000001                   DV03.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
001602   000001                   SYNA03: .BLKW 1         ;SYNC TWO
001604   000001                   DV03.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
001606   000001                   SYNB03: .BLKW 1         ;SYNC TWO
001610   000001                   DV03.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
001612   000001                   SYNC03: .BLKW 1         ;SYNC TWO
001614   000001                   DV03.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
001616   000001                   SYND03: .BLKW 1         ;SYNC TWO

001620   000001                   DVCR04: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
001622   000001                   DVTR04: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 04
001624   000001                   DV04.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
001626   000001                   SYNA04: .BLKW 1         ;SYNC TWO
001630   000001                   DV04.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
001632   000001                   SYNB04: .BLKW 1         ;SYNC TWO
001634   000001                   DV04.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04
```

```
001636   000001                   SYNC04: .BLKW 1         ;SYNC TWO
001640   000001                   DV04.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
001642   000001                   SYND04: .BLKW 1         ;SYNC TWO

001644   000001                   DVCR05: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 05
001646   000001                   DVTR05: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 05
001650   000001                   DV05.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
001652   000001                   SYNA05: .BLKW 1         ;SYNC TWO
001654   000001                   DV05.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
001656   000001                   SYNB05: .BLKW 1         ;SYNC TWO
001660   000001                   DV05.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
001662   000001                   SYNC05: .BLKW 1         ;SYNC TWO
001664   000001                   DV05.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
001666   000001                   SYND05: .BLKW 1         ;SYNC TWO

001670   000001                   DVCR06: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 06
001672   000001                   DVTR06: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 06
001674   000001                   DV06.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
001676   000001                   SYNA06: .BLKW 1         ;SYNC TWO
001700   000001                   DV06.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
001702   000001                   SYNB06: .BLKW 1         ;SYNC TWO
001704   000001                   DV06.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
001706   000001                   SYNC06: .BLKW 1         ;SYNC TWO
001710   000001                   DV06.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
001712   000001                   SYND06: .BLKW 1         ;SYNC TWO

001714   000001                   DVCR07: .BLKW 1         ;CONTROL STATUS REGISTER FOR DV11 NUMBER 07
001716   000001                   DVTR07: .BLKW 1         ;VECTOR "A" FOR DV11 NUMBER 07
001720   000001                   DV07.A: .BLKW 1         ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
001722   000001                   SYNA07: .BLKW 1         ;SYNC TWO
001724   000001                   DV07.B: .BLKW 1         ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
001726   000001                   SYNB07: .BLKW 1         ;SYNC TWO
001730   000001                   DV07.C: .BLKW 1         ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
001732   000001                   SYNC07: .BLKW 1         ;SYNC TWO
001734   000001                   DV07.D: .BLKW 1         ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
001736   000001                   SYND07: .BLKW 1         ;SYNC TWO

001740   000000                   DV.END: 000000
```

```

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

001742  012737  000340  177776  .START: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001750  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
001754  012737  004402  000024          MOV     #.PFAIL,@#24            ;SET UP POWER FAIL VECTOR
001762  113737  001301  001303          MOVB    DVNUM,SAVNUM            ;SAVE NUMBER OF DEVICES IN SYSTEM.
001770  005037  001230                  CLR     PASCNT                  ;CLEAR PASS COUNT
001774  105037  001311                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
002000  105037  001313                  CLRB    QV.FLG                  ;ZERO QUICK VERIFY FLAG
002004  012737  001500  001306          MOV     #DV.MAP,CREAM           ;GET MAP POINTER.
002012  112737  000001  001304          MOVB    #1,RUN                  ;POINT POINTER TO FIRST DEVICE.
002020  005037  001232                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
002024  005037  001234                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
002030  012737  000001  001226          MOV     #1,TSTNO                ;SET UP FOR TEST 1
002036  012737  001742  001214          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
002044  105737  001310                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
002050  001063                          BNE     1$                      ;BR IF YES
002052  013746  000004                  MOV     4,-(SP)                 ;
002056  013746  000006                  MOV     6,-(SP)                 ;
002062  005037  000006                  CLR     6                       ;
002066  012737  002104  000004          MOV     #80$,4                  ;
002074  005777  177102                  TST     @SWR                    ;
002100  000240                          NOP                             ;
002102  000407                          BR      81$                     ;
002104  022626                  80$:    CMP     (SP)+,(SP)+             ;
002106  012737  000174  001200          MOV     #LIGHT,LIGHTS           ;
002114  012737  000176  001202          MOV     #SSWR,SWR               ;
002122  012637  000006          81$:    MOV     (SP)+,6                 ;
002126  012637  000004                  MOV     (SP)+,4                 ;
002132  104402  001000                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
002136  105137  001310                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
002142  105777  177034                  TSTB    @SWR                    ;BIT7=1??
002146  100402                          BMI     16$                     ;BR IF NO AUTO SIZE
002150  004737  006624                  JSR     PC,CSRMAP               ;GO DO THE AUTO SIZE
002154  104402  005461          16$:    TYPE    ,XHEAD          ;TYPE HEADER
002160  012737  001500  001246          MOV     #DV.MAP,TEMP1           ;SET POINTER
002166  017737  177054  001250  5$:     MOV     @TEMP1,TEMP2            ;SET DATA
002174  022737  177777  001250          CMP     #177777,TEMP2           ;ALL DONE?
002202  001406                          BEQ     1$                      ;BR IF YES
002204  104410                          CONVRT                          ;
002206  005506                          XSTATQ                          ;
002210  062737  000002  001246          ADD     #2,TEMP1                ;UPDATE POINTER
002216  000763                          BR      5$
002220  005737  000042          1$:     TST     @#42                    ;IS PROGRAM RUNNING UNDER MONITOR
002224  001030                          BNE     3$                      ;BR IF YES
002226  032777  000001  176746          BIT     #SW00,@SWR              ;SELECT SPECIFIC DEVICES??
002234  001424                          BEQ     3$                      ;BR IF NO.
002236  104402  005402                  TYPE    ,MNEW                   ;TYPE THE MESSAGE.
002242  005000                          CLR     R0                      ;ZERO DATA LIGHTS
```

```
002244  000000                          HALT                            ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
002246  127737  176730  001302          CMPB    @SWR,SAVACT             ;IS THE NUMBER VALID?
002254  101404                          BLOS    2$                      ;BR IF NUMBER IS OK.
002256  104402  005243                  TYPE    ,MERR3                  ;TELL USER OF INVALID NUMBER.
002262  000000                          HALT                            ;STOP EVERY THING.
002264  000776                          BR      .-2                     ;RESTART THE PROGRAM AGAIN.
002266  117737  176710  001300  2$:     MOVB    @SWR,DVACTV             ;GET NEW DEVICE PATTERN
002274  113700  001300                  MOVB    DVACTV,R0               ;SHOW THE USER WHAT HE SELECTED.
002300  042700  177400                  BIC     #^C<377>,R0             ;USE ONLY LOW BYTE.
002304  000000                          HALT                            ;CONTINUE DYNAMIC SWITCHES.
002306  012700  000300          3$:     MOV     #300,R0                 ;PREPARE TO CLEAR THE FLOATING
002312  012701  000302                  MOV     #302,R1                 ;VECTOR AREA. 300-776
002316  010120                  4$:     MOV     R1,(R0)+                ;START PUTTING "PC+2 - HALT"
002320  005021                          CLR     (R1)+                   ;IN VECTOR AREA.
002322  022021                          CMP     (R0)+,(R1)+             ;POP POINTERS
002324  022700  001000                  CMP     #1000,R0                ;ALL DONE??
002330  001372                          BNE     4$                      ;BR IF NO.

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

002332  012737  000340  177776  .BEGIN: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
002340  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
002344  005737  000042                  TST     @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
002350  001023                          BNE     3$                      ;BR IF YES
002352  032777  000004  176622          BIT     #BIT2,@SWR              ;CHECK FOR LOCK ON TEST
002360  001411                          BEQ     1$                      ;BR IF NO LOCK DESIRED.
002362  104402  005301                  TYPE    ,MLOCK                  ;TYPE LOCK SELECTED.
002366  012737  000240  002702          MOV     #NOP,TTST               ;ADJUST SCOPE ROUTINE.
002374  012737  000240  002704          MOV     #NOP,TTST+2             ;SET UP TO LOCK
002402  000406                          BR      2$                      ;CONTINUE ALONG.
002404  013737  003014  002702  1$:     MOV     BRW,TTST                ;PREPARE NORMAL SCOPE ROUTINE
002412  013737  003016  002704          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
002420                          2$:
002420  012737  005666  001214  3$:     MOV     #CYCLE,RETURN           ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
002426  104402  005171          4$:     TYPE    ,MR                     ;TYPE R
002432  000177  176556                  JMP     @RETURN                 ;START TESTING
```

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

002436  000005                        .EOP:   RESET                   ;MAKE THE  WORLD CLEAN AGAIN.
002440  005037  001234                        CLR     LSTERR          ;CLEAR LAST ERROR PC
002444  105037  001311                        CLRB    ERRFLG          ;CLEAR ERROR FLAG
002450  005237  001230                        INC     PASCNT          ;UPDATE PASS COUNT
002454  013777  001230  176516                MOV     PASCNT,@LIGHTS  ;DISPLAY PASS COUNT
002462  104402  005145                        TYPE    ,MEPASS         ;TYPE END PASS
002466  104402  005330                        TYPE    ,MCSRX          ;TYPE CSR
002472  104411  002604                        CNVRT   ,XCSR           ;SHOW IT
002476  104402  005336                        TYPE    ,MVECX          ;TYPE VECTOR
002502  104411  002612                        CNVRT   ,XVEC           ;SHOW IT
002506  104402  005344                        TYPE    ,MPASSX         ;TYPE PASSES
002512  104411  002620                        CNVRT   ,XPASS          ;SHOW IT
002516  104402  005355                        TYPE    ,MERRX          ;TYPE ERRORS
002522  104411  002626                        CNVRT   ,XERR           ;SHOW IT
002526  105337  001303                        DECB    SAVNUM          ;ARE ALL DEVICES TESTED?
002532  001017                                BNE     RESTRT          ;BR IF NO.
002534  112737  000377  001313                MOVB    #377,QV,FLG     ;SET THE QUICK VERIFY FLAG.
002542  113737  001301  001303                MOVB    DVNUM,SAVNUM    ;RESTORE THE COUNT
002550  013701  000042                        MOV     @#42,R1         ;CHECK FOR ACT-11 OR DDP
002554  001406                                BEQ     RESTRT          ;IF NOT, CONTINUE TESTING
002556  000005                                RESET                   ;STOP THE SHOW--CLEAR THE WORLD
002560                                LOGICAL:
002560  004711                                JSR     PC,(R1)
002562  000240                                NOP
002564  000240                                NOP
002566  000240                                NOP
002570  000240                                NOP
002572  012737  005666  001214        RESTRT: MOV     #CYCLE,RETURN
002600  000137  005666                        JMP     CYCLE
002604  000001                        XCSR:   1
002606     006     002                        .BYTE   6,2
002610  001362                                DVSCR
002612  000001                        XVEC:   1
002614     003     002                        .BYTE   3,2
002616  001352                                DVRVEC
002620  000001                        XPASS:  1
002622     006     002                        .BYTE   6,2
002624  001230                                PASCNT
002626  000001                        XERR:   1
002630     006     002                        .BYTE   6,2
002632  001232                                ERRCNT

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

002634                                .SCOPE:
002634  022737  177570  001202                CMP     #177570,SWR     ;IS THERE A REAL SWR?
002642  001411                                BEQ     64$             ;BR IF YES
002644  017746  176336                        MOV     @TKDBR,-(SP)    ;SAVE KEYBOARD CHAR
002650  042716  000200                        BIC     #BIT7,(SP)      ;CLEAR PARITY BIT
```

```
002654  122726  000007                        CMPB    #7,(SP)+        ;WAS IT CNTRL 'G' ?
002660  001002                                BNE     .+6             ;BR IF NO.
002662  004737  004640                        JSR     PC,SERV,G       ;SERVICE "CNTRL 'G'".
002666  005037  001234                64$:    CLR     LSTERR          ;CLEAR LAST ERROR PC.
002672  010016                                MOV     R0,(SP)         ;SAVE R0 ON THE STACK
002674  032777  040000  176300                BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
002702  001407                        TTST:   BEQ     1$              ;BR IF NO,  (IF LOCK SW01=1; THIS LOC =240)
002704  000437                                BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
002706  105777  176272                        TSTB    @TKCSR          ;KEYBOAD DONE?
002712  100034                                BPL     3$              ;BR IF NO. (LOCK; HIT KEY TO GOTO NEXT TEST)
002714  017700  176266                        MOV     @TKDBR,R0       ;CLEAR DONE BIT
002720  000415                                BR      2$              ;CONTINUE
002722  032777  004000  176252        1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
002730  001011                                BNE     2$              ;BR IF YES
002732  105737  001313                        TSTB    QV,FLG          ;HAVE PASSES BEECOMPLETED?
002736  001406                                BEQ     2$              ;BR IF QUICK PASS.
002740  005237  001224                        INC     LPCNT           ;UPDATE ITERATION COUNTER
002744  023737  001224  001222                CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
002752  001014                                BNE     3$              ;BR IF NOT YET
002754  105037  001311                2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
002760  005037  001224                        CLR     LPCNT           ;START ICOUNTER AT 0
002764  005037  001220                        CLR     LOCK
002770  012737  000020  001222                MOV     #20,ICOUNT      ;RESET ITERATIONS
002776  013737  001216  001214                MOV     NEXT,RETURN     ;GET NEXT TEST
003004  011600                        3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
003006  022626                                POP2SP                  ;FAKE AN "RTI"
003010  000177  176200                        JMP     @RETURN         ;GO DO THE TEST
003014  001407                        BRW:    1407
003016  000437                        BRX:    437

                                              ;CHECK FOR FREEZE ON CURRENT DATA
                                              ;-------------------------------

003020  032777  001000  176154        .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
003026  001405                                BEQ     1$              ;BR IF NOT SET.
003030  005737  001220                        TST     LOCK
003034  001402                                BEQ     1$
003036  013716  001220                        MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
003042  000002                        1$:     RTI                     ;GO BACK.

                                              ;TELETYPE OUTPUT ROUTINE
                                              ;----------------------

003044  010546                        .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
003046  017605  000002                        MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
003052  062766  000002  000002                ADD     #2,2(SP)        ;POP OVER ADDRESS.
003060  032777  010000  176114        1$:     BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
003066  001012                                BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
003070  105715                                TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
003072  100002                                BPL     2$              ;BR IF NUMBER IS PLUS
003074  104402  005104                        TYPE    ,MCRLF          ;TYPE A CR/LF!
003100  105777  176104                2$:     TSTB    @TPCSR          ;TTY READY?
003104  100375                                BPL     2$              ;BR IF NO.
003106  112577  176100                        MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
003112  001362                                BNE     1$              ;IF NOT ZERO KEEP PRINTING!
003114  012605                        3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
ØØ3116  ØØØØØ2                                  RTI                     ;GO HOME
                                                ;-------------------------

ØØ312Ø  Ø1Ø346                          .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
ØØ3122  Ø1Ø446                                  MOV     R4,-(SP)        ;SAVE R4 ON STACK
ØØ3124  Ø17637  ØØØØØ4  ØØ3142                  MOV     @4(SP),.MSG
ØØ3132  Ø62766  ØØØØØ2  ØØØØØ4                  ADD     #2,4(SP)
ØØ314Ø  1Ø44Ø2                          .INST1: TYPE
ØØ3142  ØØØØØØ                          .MSG:   Ø
ØØ3144  Ø127Ø4  ØØ552Ø                          MOV     #INBUF,R4
ØØ315Ø  Ø127Ø3  ØØØØØ7                          MOV     #7,R3
ØØ3154  1Ø5777  176Ø24                  1$:     TSTB    @TKCSR
ØØ316Ø  1ØØ375                                  BPL     1$
ØØ3162  117714  176Ø2Ø                          MOVB    @TKDBR,(R4)
ØØ3166  142714  ØØØ2ØØ                          BICB    #2ØØ,(R4)
ØØ3172  122427  ØØØØ15                          CMPB    (R4)+,#15
ØØ3176  ØØ1417                                  BEQ     INSTR2
ØØ32ØØ  1Ø5777  176ØØ4                  2$:     TSTB    @TPCSR
ØØ32Ø4  1ØØ375                                  BPL     2$
ØØ32Ø6  Ø17777  175774  175776                  MOV     @TKDBR,@TPDBR
ØØ3214  ØØ53Ø3                                  DEC     R3
ØØ3216  ØØ1356                                  BNE     1$
ØØ322Ø  Ø126Ø4                                  MOV     (SP)+,R4
ØØ3222  Ø126Ø3                                  MOV     (SP)+,R3
ØØ3224  1Ø44Ø2  ØØ51ØØ                  .INSTE: TYPE    ,MQM
ØØ323Ø  Ø1Ø346                                  MOV     R3,-(SP)
ØØ3232  Ø1Ø446                                  MOV     R4,-(SP)
ØØ3234  ØØØ741                                  BR      .INST1
ØØ3236  Ø126Ø4                          INSTR2: MOV     (SP)+,R4        ;RESTORE R4
ØØ324Ø  Ø126Ø3                                  MOV     (SP)+,R3        ;RESTORE R3
ØØ3242  ØØØØØ2                                  RTI

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

ØØ3244  Ø1Ø546                          .PARAM: MOV     R5,-(SP)
ØØ3246  Ø1Ø446                                  MOV     R4,-(SP)
ØØ325Ø  Ø166Ø5  ØØØØØ4                          MOV     4(SP),R5
ØØ3254  Ø12537  ØØ3434                          MOV     (R5)+,LOLIM
ØØ326Ø  Ø12537  ØØ3436                          MOV     (R5)+,HILIM
ØØ3264  Ø12537  ØØ344Ø                          MOV     (R5)+,DEVADR
ØØ327Ø  112537  ØØ3442                          MOVB    (R5)+,LOBITS
ØØ3274  112537  ØØ3443                          MOVB    (R5)+,ADRCNT
ØØ33ØØ  Ø1Ø566  ØØØØØ4                          MOV     R5,4(SP)
ØØ33Ø4  ØØ5ØØ5                          PARAM1: CLR     R5
ØØ33Ø6  Ø127Ø4  ØØ552Ø                          MOV     #INBUF,R4
ØØ3312  122714  ØØØØ15                          CMPB    #15,(R4)
ØØ3316  ØØ142Ø                                  BEQ     PARERR
ØØ332Ø  121427  ØØØØ6Ø                  1$:     CMPB    (R4),#6Ø
ØØ3324  ØØ2415                                  BLT     PARERR
ØØ3326  121427  ØØØØ67                          CMPB    (R4),#67
ØØ3332  ØØ3Ø12                                  BGT     PARERR
ØØ3334  142714  ØØØØ6Ø                          BICB    #6Ø,(R4)
ØØ334Ø  1524Ø5                                  BISB    (R4)+,R5
ØØ3342  122714  ØØØØ15                          CMPB    #15,(R4)
ØØ3346  ØØ14Ø6                                  BEQ     LIMITS
```

```
ØØ335Ø  ØØ63Ø5                                  ASL     R5
ØØ3352  ØØ63Ø5                                  ASL     R5
ØØ3354  ØØ63Ø5                                  ASL     R5
ØØ3356  ØØØ76Ø                                  BR      1$
ØØ336Ø  1Ø44Ø4                          PARERR: INSTER
ØØ3362  ØØØ75Ø                                  BR      PARAM1

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

ØØ3364  Ø2Ø537  ØØ3436                  LIMITS: CMP     R5,HILIM
ØØ337Ø  1Ø1373                                  BHI     PARERR
ØØ3372  Ø2Ø537  ØØ3434                          CMP     R5,LOLIM
ØØ3376  1Ø377Ø                                  BLO     PARERR
ØØ34ØØ  1337Ø5  ØØ3442                          BITB    LOBITS,R5
ØØ34Ø4  ØØ1365                                  BNE     PARERR

                                                ;STORE NUMBER AT SPECIFIED ADDRESS

ØØ34Ø6  Ø137Ø4  ØØ344Ø                          MOV     DEVADR,R4
ØØ3412  Ø1Ø524                          1$:     MOV     R5,(R4)+
ØØ3414  Ø627Ø5  ØØØØØ2                          ADD     #2,R5
ØØ342Ø  1Ø5337  ØØ3443                          DECB    ADRCNT
ØØ3424  ØØ1372                                  BNE     1$
ØØ3426  Ø126Ø4                                  MOV     (SP)+,R4
ØØ343Ø  Ø126Ø5                                  MOV     (SP)+,R5
ØØ3432  ØØØØØ2                                  RTI
ØØ3434  ØØØØØØ                          LOLIM:  Ø
ØØ3436  ØØØØØØ                          HILIM:  Ø
ØØ344Ø  ØØØØØØ                          DEVADR: Ø
ØØ3442  ØØØØØØ                          LOBITS: Ø
        ØØ3443                          ADRCNT=LOBITS+1

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

ØØ3444  Ø16637  ØØØØØ4  ØØ1276          .SAVØ5: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                                ;SAVE RØ-R5

ØØ3452  Ø1Ø537  ØØ1272                  SVØ5:   MOV     R5,SAVR5        ;SAVE R5
ØØ3456  Ø1Ø437  ØØ127Ø                          MOV     R4,SAVR4        ;SAVE R4
ØØ3462  Ø1Ø337  ØØ1266                          MOV     R3,SAVR3        ;SAVE R3
ØØ3466  Ø1Ø237  ØØ1264                          MOV     R2,SAVR2        ;SAVE R2
ØØ3472  Ø1Ø137  ØØ1262                          MOV     R1,SAVR1        ;SAVE R1
ØØ3476  Ø1ØØ37  ØØ126Ø                          MOV     RØ,SAVRØ        ;SAVE RØ
ØØ35Ø2  ØØØØØ2                                  RTI                     ;LEAVE.

                                                ;RESTORE RØ-R5

ØØ35Ø4  Ø137ØØ  ØØ126Ø                  .RESØ5: MOV     SAVRØ,RØ        ;RESTORE RØ
ØØ351Ø  Ø137Ø1  ØØ1262                          MOV     SAVR1,R1        ;RESTORE R1
ØØ3514  Ø137Ø2  ØØ1264                          MOV     SAVR2,R2        ;RESTORE R2
ØØ352Ø  Ø137Ø3  ØØ1266                          MOV     SAVR3,R3        ;RESTORE R3
ØØ3524  Ø137Ø4  ØØ127Ø                          MOV     SAVR4,R4        ;RESTORE R4
ØØ353Ø  Ø137Ø5  ØØ1272                          MOV     SAVR5,R5        ;RESTORE R5
```

```
003534  000002                            RTI                    ;LEAVE

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

003536  104402  005104           .CONVR:  TYPE    ,MCRLF
003542  010046                   .CNVRT:  MOV     R0,-(SP)
003544  010146                            MOV     R1,-(SP)
003546  010346                            MOV     R3,-(SP)
003550  010446                            MOV     R4,-(SP)
003552  010546                            MOV     R5,-(SP)
003554  017601  000012                    MOV     @12(SP),R1
003560  062766  000002  000012            ADD     #2,12(SP)
003566  012137  003742                    MOV     (R1)+,WRDCNT
003572  112137  003744           1$:      MOVB    (R1)+,CHRCNT
003576  112137  003745                    MOVB    (R1)+,SPACNT
003602  013137  003746                    MOV     @(R1)+,BINWRD
003606  013704  003746           2$:      MOV     BINWRD,R4
003612  113705  003744                    MOVB    CHRCNT,R5
003616  012700  005562                    MOV     #TEMP,R0
003622  010403                   3$:      MOV     R4,R3
003624  042703  177770                    BIC     #177770,R3
003630  062703  000060                    ADD     #060,R3
003634  110320                            MOVB    R3,(R0)+
003636  000241                            CLC
003640  006004                            ROR     R4
003642  000241                            CLC
003644  006004                            ROR     R4
003646  000241                            CLC
003650  006004                            ROR     R4
003652  005305                            DEC     R5
003654  001362                            BNE     3$
003656  012703  005624                    MOV     #MDATA,R3
003662  114023                   4$:      MOVB    -(R0),(R3)+
003664  105337  003744                    DECB    CHRCNT
003670  001374                            BNE     4$
003672  105737  003745                    TSTB    SPACNT
003676  001405                            BEQ     6$
003700  112723  000040           5$:      MOVB    #040,(R3)+
003704  105337  003745                    DECB    SPACNT
003710  001373                            BNE     5$
003712  105013                   6$:      CLRB    (R3)
003714  104402  005624                    TYPE    ,MDATA
003720  005337  003742                    DEC     WRDCNT
003724  001322                            BNE     1$
003726  012605                            MOV     (SP)+,R5
003730  012604                            MOV     (SP)+,R4
003732  012603                            MOV     (SP)+,R3
003734  012601                            MOV     (SP)+,R1
003736  012600                            MOV     (SP)+,R0
003740  000002                            RTI
003742  000000                   WRDCNT:  0
003744  000000                   CHRCNT:  0
        003745                   SPACNT=CHRCNT+1
003746  000000                   BINWRD:  0
```

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

003750  011646                   .TRPSR:  MOV     (SP),-(SP)             ;GET PC OF RETURN
003752  162716  000002                    SUB     #2,(SP)                ;=PC OF TRAP
003756  017616  000000                    MOV     @(SP),(SP)             ;GET TRP
003762  006316                   TRPOK:   ASL     (SP)                   ;MULTIPLY TRAP ARG BY 2
003764  042716  177001                    BIC     #177001,(SP)           ;CLEAR UNWANTED BITS
003770  062716  001314                    ADD     #.TRPTAB,(SP)          ;POINTER TO SUBROUTINE ADDRESS
003774  017616  000000                    MOV     @(SP),(SP)             ;SUBROUTINE ADDRESS
004000  000136                            JMP     @(SP)+                 ;GO TO SUBROUTINE

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

004002                           .HLT:
004002  022737  177570  001202            CMP     #177570,SWR            ;IS THERE A REAL SWR?
004010  001411                            BEQ     64$                    ;BR IF YES
004012  017746  175170                    MOV     @TKDBR,-(SP)           ;SAVE KEYBOARD CHAR
004016  042716  000200                    BIC     #BIT7,(SP)             ;CLEAR PARITY BIT
004022  122726  000007                    CMPB    #7,(SP)+               ;WAS IT CNTRL "G" ?
004026  001002                            BNE     .+6                    ;BR IF NO.
004030  004737  004640                    JSR     PC,SERV.G              ;SERVICE "CNTRL "G"".
004034  032777  010000  175140   64$:     BIT     #SW12,@SWR             ;BELL ON ERROR?
004042  001406                            BEQ     XBX                    ;BR IF NO BELL
004044  105777  175140                    TSTB    @TPCSR                 ;TTY READY.
004050  100003                            BPL     XBX                    ;DON'T WAIT IF TTY NOT READY.
004052  112777  000207  175132            MOVB    #207,@TPDBR            ;PUSH A BELL AT THE TTY.
004060  032777  020000  175114   XBX:     BIT     #SW13,@SWR             ;DELETE ERROR PRINT OUT?
004066  001105                            BNE     HALTS                  ;BR IF NO PRINT OUT WANTED.
004070  021637  001234                    CMP     (SP),LSTERR            ;WAS THIS ERROR FOUND LAST TIME?
004074  001404                            BEQ     1$                     ;BR IF YES
004076  011637  001234                    MOV     (SP),LSTERR            ;RECORD BEING HERE
004102  105037  001311                    CLRB    ERRFLG                 ;PREPARE HEADER
004106  104406                   1$:      SAV05                          ;SAVE ALL PROC REGISTERS
004110  011605                            MOV     (SP),R5                ;GET THE PC OF ERROR
004112  162705  000002                    SUB     #2,R5                  ;GET ADDRESS OF TRAP CALL
004116  011504                            MOV     (R5),R4                ;GET HLT INSTRUCTION
004120  006304                            ASL     R4                     ;MULT BY TWO
004122  061504                            ADD     (R5),R4                ;DOUBLE IT
004124  006304                            ASL     R4                     ;MULT AGAIN
004126  042704  177001                    BIC     #177001,R4             ;CLEAR JUNK
004132  062704  033124                    ADD     #.ERRTAB,R4            ;GET POINTER
004136  012437  004252                    MOV     (R4)+,ERRMSG           ;GET ERROR MESSAGE
004142  012437  004264                    MOV     (R4)+,DATAHD           ;GET DATA HEADRER
004146  011437  004276                    MOV     (R4),DATABP            ;GET DATA TABLE
004152  105737  001311                    TSTB    ERRFLG                 ;TYPE HEADREER
004156  001403                            BEQ     TYPMSG                 ;BR IF YES
004160  005737  004276                    TST     DATABP                 ;DOES DATA TABLE EXIST?
004164  001040                            BNE     TYPDAT                 ;BR IF YES.
004166  104402  005104           TYPMSG:  TYPE    ,MCRLF
004172  104402  005104                    TYPE    ,MCRLF
004176  005737  001220                    TST     LOCK
```

```
004202  001402                                  BEQ     1$
004204  104402  005400                          TYPE    ,MASTEK
004210  104402  005366                  1$:     TYPE    ,MTSTN
004214  104411  004374                          CNVRT   ,XTSTN          ;SHOW IT
004220  104402  005454                          TYPE    ,MERRPC         ;TYPE PC.
004224  104411  004366                          CNVRT   ,ERTAB0         ;SHOW IT
004230  104402  005104                          TYPE    ,MCRLF          ;GIVE A CR/LF
004234  112737  177777  001311                  MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
004242  005737  004252                          TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
004246  001402                                  BEQ     WRKO.FM         ;BR IF NO.
004250  104402                                  TYPE                    ;TYPE
004252  000000                          ERRMSG: 0                       ;     ERROR MESSAGE
004254                                  WRKO.FM:                        ;
004254  005737  004264                          TST     DATAHD          ;DATA HEADER?
004260  001402                                  BEQ     TYPDAT          ;BR IF NO
004262  104402                                  TYPE                    ;TYPE
004264  000000                          DATAHD: 0                       ;     DATA HEADER
004266  005737  004276                  TYPDAT: TST     DATABP          ;DATA TABLE?
004272  001402                                  BEQ     RESREG          ;BR IF NO.
004274  104410                                  CONVRT                  ;SHOW
004276  000000                          DATABP: 0                       ;     DATA TABLE
004300  104407                          RESREG: RES05                   ;RESTORE PROC REGISTERS
004302  005777  174674                  HALTS:  TST     @SWR            ;HALT ON ERROR?
004306  100005                                  BPL     EXITER          ;BR IF NO HALT ON ERROR
004310  010046                                  PUSHR0                  ;SAVE RO
004312  016600  000002                          MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
004316  000000                                  HALT                    ;HALT
004320  012600                                  POPR0                   ;GET RO
004322  005237  001232                  EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
004326  032777  000400  174646                  BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
004334  001007                                  BNE     1$              ;BR IF YES
004336  032777  002000  174636                  BIT     #SW10,@SWR      ;GOTO NEXT TEST?
004344  001407                                  BEQ     2$              ;BR IF NO
004346  013737  001216  001214                  MOV     NEXT,RETURN     ;SET FOR NEXT TEST
004354  012706  001200                  1$:     MOV     #STACK,SP       ;RESET SP
004360  000177  174630                          JMP     @RETURN         ;GOTO SPECIFIED TEST
004364  000002                          2$:     RTI                     ;RETURN
004366  000001                          ERTAB0: 1
004370     006     002                          .BYTE   6,2
004372  001276                                  SAVPC
004374  000001                          XTSTN:  1
004376     003     002                          .BYTE   3,2
004400  001226                                  TSTNO
                                                ;ENTER HERE ON POWER FAILURE
                                                ;---------------------------



004402                                  .PFAIL:
004402  012737  004414  000024                  MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
004410  000000                                  HALT                            ;HALT ON POWER DOWN NORMAL
004412  000777                                  BR      .

                                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

004414                                  RESTAR:
004414  012737  004402  000024                  MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
```

```
004422  012706  001200                          MOV     #STACK,SP       ;RESET THE STACK POINTER
004426  005037  005562                          CLR     TEMP            ;READY FOR TIMMER
004432  005237  005562                          INC     TEMP            ;PLUS ONE TO THE TIMER!
004436  001375                                  BNE     .-4             ;BR IF MORE TO GO
004440  104402  005107                          TYPE    ,MPFAIL         ;TYPE THE MESSAGE
004444  104411  004470                          CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
004450  105037  001311                          CLRB    ERRFLG          ;START CLEAN
004454  005037  001234                          CLR     LSTERR          ;"""""""""""
004460  104412                                  MSTCLR                  ;START CLEAN UP OF DEVICE
004462  104413                                  RAMCLR                  ;CLEAR IT ALL!
004464  000177  174524                          JMP     @RETURN         ;START DOING THAT TEST AGAIN.
004470  000001                          PFTAB:  1
004472     003     002                  .BYTE   3,2
004474  001226                                  TSTNO
004476  010046                          .DELAY: MOV     R0,-(SP)
004500  013700  004514                          MOV     1$,R0
004504  005300                                  DEC     R0
004506  001376                                  BNE     .-2
004510  012600                                  MOV     (SP)+,R0
004512  000002                                  RTI
004514  000036                          1$:     30.

004516                                  .RAMCLR:
004516  012777  004000  174636                  MOV     #MRESET,@DVSCR  ;ISSUE A MASTER CLEAR
004524  010146                                  MOV     R1,-(SP)        ;SAVE R1 ON THE STACK
004526  010446                                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
004530  013701  001372                          MOV     DVSRS,R1        ;GET SECONDARY SEL. REG.
004534  013704  001376                          MOV     DVSRA,R4        ;GET SECONDARY REGISTER ACCESS REG.
004540  005014                          1$:     CLR     (R4)            ;ZERO THE SECONDARY REGISTER.
004542  062711  170361                          ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,(R1)
004546  001374                                  BNE     1$
004550  012604                                  MOV     (SP)+,R4        ;RESTORE R4
004552  012601                                  MOV     (SP)+,R1        ;RESTORE R1
004554  000002                                  RTI

004556                                  .MSTCLR:
004556  012777  004000  174576                  MOV     #MRESET,@DVSCR  ;ISSUE MASTER CLEAR.
004564  000002                                  RTI

004566                                  .ROMCLK:
004566  052777  000002  174566                  BIS     #BIT1,@DVSCR
004574  000002                                  RTI

004576                                  .DATACLK:
004576  010046                                  MOV     R0,-(SP)
004600  005000                                  CLR     R0
004602  052777  000400  174560                  BIS     #BIT8,@DVLCR
004610  017737  174554  004636          1$:     MOV     @DVLCR,3$
004616  106037  004637                          RORB    3$+1
004622  103003                                  BCC     2$
004624  005200                                  INC     R0
004626  001370                                  BNE     1$
004630  104000                                  HLT     0
004632  012600                          2$:     MOV     (SP)+,R0
004634  000002                                  RTI
004636  000001                          3$:     .BLKW   1
```

```
911
912  004640  032777  004000  174336  SERV.G: BIT     #4000,@TKCSR      ;RX BUSY?
913  004646  001374                          BNE     SERV.G            ;BR IF YES
914  004650  017737  174326  005072          MOV     @SWR,90$          ;SAVE (SWR).
915  004656  013777  005072  174316  1$:     MOV     90$,@SWR          ;
916  004664  104402  005052                  TYPE    ,89$              ;
917  004670  104411  005064                  CNVRT   ,88$              ;
918  004674  104402  005074                  TYPE    ,91$              ;
919  004700  105777  174300                  TSTB    @TKCSR            ;WAIT FOR DONE.
920  004704  100375                          BPL     .-4
921  004706  017746  174274                  MOV     @TKDBR,-(SP)      ;
922  004712  042716  000200                  BIC     #BIT7,(SP)        ;
923  004716  122726  000015                  CMPB    #15,(SP)+         ;
924  004722  001450                          BEQ     5$                ;
925  004724  005077  174252                  CLR     @SWR              ;
926  004730  105777  174254          2$:     TSTB    @TPCSR            ;
927  004734  100375                          BPL     .-4
928  004736  016677  177776  174246          MOV     -2(SP),@TPDBR     ;
929  004744  000241                          CLC                       ;
930  004746  006177  174230                  ROL     @SWR              ;
931  004752  006177  174224                  ROL     @SWR              ;
932  004756  006177  174220                  ROL     @SWR              ;
933  004762  103735                          BCS     1$                ;ERROR
934  004764  026627  177776  000060          CMP     -2(SP),#60        ;
935  004772  002731                          BLT     1$                ;
936  004774  026627  177776  000067          CMP     -2(SP),#67        ;
937  005002  003325                          BGT     1$                ;
938  005004  042766  177770  177776          BIC     #^C<7>,-2(SP)     ;
939  005012  056677  177776  174162          BIS     -2(SP),@SWR       ;
940  005020  105777  174160                  TSTB    @TKCSR            ;
941  005024  100375                          BPL     .-4               ;
942  005026  017746  174154                  MOV     @TKDBR,-(SP)
943  005032  042716  000200                  BIC     #BIT7,(SP)
944  005036  122726  000015                  CMPB    #15,(SP)+
945  005042  001332                          BNE     2$                ;
946  005044  104402  005104          5$:     TYPE    ,MCRLF            ;
947  005050  000207                          RTS     PC
948
949  005052  020377  051450  051127  89$:    .ASCIZ  <377>? (SWR)=/?
950  005060  036451  000057
951                                  .EVEN
952  005064  000001                  88$:    1
953  005066     006     000                  .BYTE   6,0
954  005070  005072                          90$
955  005072  000000                  90$:    .WORD   0
956  005074  036457  000057          91$:    .ASCIZ  ?/=/?
957                                  .EVEN
958  005100  020040  000077          MQM:    .ASCIZ  /  ?/
(2)  005104  005015     000          MCRLF:  .ASCIZ   <15><12>
(2)  005107     377  053520  020122  MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
(2)  005145     377  047105  020104  MEPASS: .ASCIZ  <377>/END PASS DZDVA-B  /
(2)  005171     377  000122          MR:     .ASCIZ  <377>/R/
(2)  005174  050377  047522  051107  MERR2:  .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2)  005243     377  047111  052523  MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
(2)  005267     377  042524  052123  MTSTPC: .ASCIZ  <377>/TEST PC=/
(2)  005301     377  047514  045503  MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
```

```
(2)  005330  051503  035122  000040  MCSRX:  .ASCIZ   /CSR: /
(2)  005336  042526  035103  000040  MVECX:  .ASCIZ  /VEC: /
(2)  005344  040520  051523  051505  MPASSX: .ASCIZ  /PASSES: /
(2)  005355     105  051122  051117  MERRX:  .ASCIZ  /ERRORS: /
(2)  005366  042524  052123  047040  MTSTN:  .ASCIZ  /TEST NO: /
(2)  005400  000052                  MASTEK: .ASCIZ  /*/
(2)  005402  051777  052105  051440  MNEW:   .ASCIZ  <377>/SET SWITCH REG TO DV11'S DESIRED ACTIVE./
(2)  005454  041520  020072     000  MERRPC: .ASCIZ  /PC: /
(2)  005461     377  040515  020120  XHEAD:  .ASCIZ  <377>/MAP OF DV11 STATUS/<377>
(2)                                  .EVEN
(2)  005506  000002                  XSTATQ: 2
959  005510     006     003                  .BYTE   6,3
960  005512  001246                          TEMP1
961  005514     006     002                  .BYTE   6,2
962  005516  001250                          TEMP2
963                                  .EVEN
964
965                                          ;BUFFERS FOR INPUT-OUTPUT
966
967  005520  000000                  INBUF:  0
968          005562                  .=.+40
969  005562  000000                  TEMP:   0
970          005624                  .=.+40
971  005624  000000                  MDATA:  0
972          005666                  .=.+40
```

```


                                                   ;
                                                   ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DV11'S
                                                   ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                   ;AND RUNS THE SPECIFIED DV11'S.   THIS ROUTINE *MUST*
                                                   ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                   ;SETUP NECESSARY.
                                                   ;

005666  105737  001300          CYCLE:  TSTB    DVACTV          ;ARE ANY DV11'S TO BE TESTED?
005672  001004                          BNE     1$              ;BR IF OK.
005674  104402  005174                  TYPE    ,MERR2          ;NO DV11'S SELECTED!!
005700  000000                          HALT                    ;STOP THE SHOW.
005702  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
005704  133737  001304  001300  1$:     BITB    RUN,DVACTV      ;IS THIS ONE "ACTIVE"
005712  001020                          BNE     2$              ;BR IF GOOD ONE FOUND.
005714  000241                          CLC                     ;CLEAR PROC. CARRY BIT.
005716  106137  001304                  ROLB    RUN             ;UPDATE POINTER
005722  105537  001304                  ADCB    RUN             ;CATCH CARRY FROM RUN
005726  062737  000024  001306          ADD     #24,CREAM       ;UPDATE ADDRESS POINTER.
005734  022737  001740  001306          CMP     #DV.END,CREAM
005742  001360                          BNE     1$              ;KEEP GOING; NOT ALL TESTED FOR.
005744  012737  001500  001306          MOV     #DV.MAP,CREAM   ;RESET ADDRESS POINTER.
005752  000754                          BR      1$              ;KEEP LOOKING FOR ACTIVE DV11
005754  000241                  2$:     CLC                     ;CLEAR PROC. CARRY.
005756  106137  001304                  ROLB    RUN             ;UPDATE POINTER.
005762  105537  001304                  ADCB    RUN             ;CATCH CARRY.
005766  013700  001306                  MOV     CREAM,R0        ;GET ADDRESS POINTER.
005772  062737  000024  001306          ADD     #24,CREAM       ;UPDATE.
006000  022737  001740  001306          CMP     #DV.END,CREAM
                                                                ;ALL DONE?
006006  001003                          BNE     3$              ;BR IF NO.
006010  012737  001500  001306          MOV     #DV.MAP,CREAM   ;RESTORE POINTER.
006016  012037  001362          3$:     MOV     (R0)+,DVSCR     ;LOAD SYSTEM CTRL. REG
006022  012037  001352                  MOV     (R0)+,DVRVEC    ;LOAD VECTOR
006026  012037  001416                  MOV     (R0)+,L00.03    ;GET LINE PARAMETERS. 00-03
006032  012037  001426                  MOV     (R0)+,SYNC2A    ;
006036  012037  001420                  MOV     (R0)+,L04.07    ;                     04-07
006042  012037  001430                  MOV     (R0)+,SYNC2B    ;
006046  012037  001422                  MOV     (R0)+,L08.11    ;                     08-11
006052  012037  001432                  MOV     (R0)+,SYNC2C    ;
006056  012037  001424                  MOV     (R0)+,L12.15    ;                     12-15
006062  012037  001434                  MOV     (R0)+,SYNC2D    ;
006066  012700  000002                  MOV     #2,R0           ;SAVE CORE THIS WAY!
006072  013737  001362  001364          MOV     DVSCR,DVSCRH    ;GET SYS CTRL. REG HIGH BYTE.
006100  005237  001364                  INC     DVSCRH          ;GOT IT.
006104  013737  001364  001366          MOV     DVSCRH,DVRIC    ;GET NXT REC. CHAR REG.
006112  005237  001366                  INC     DVRIC           ;GOT IT
006116  013737  001366  001370          MOV     DVRIC,DVLCR     ;GET LN. PAR.REG.
006124  060037  001370                  ADD     R0,DVLCR        ;GOT IT
006130  013737  001370  001372          MOV     DVLCR,DVSRS     ;GET SEC. REG. SEL. REG.
006136  060037  001372                  ADD     R0,DVSRS        ;GOT IT
006142  013737  001372  001374          MOV     DVSRS,DVSRSH    ;GET HIGH BYTE.
006150  005237  001374                  INC     DVSRSH          ;GOT IT
006154  013737  001374  001376          MOV     DVSRSH,DVSRA    ;SEC. REG. ACCESS.
006162  005237  001376                  INC     DVSRA           ;GOT IT
```

```
006166  013737  001376  001400          MOV     DVSRA,DVSFR     ;SPEC. FUN. REG.
006174  060037  001400                  ADD     R0,DVSFR        ;
006200  013737  001400  001402          MOV     DVSFR,DVNSR     ;NPR STAT. REG.
006206  060037  001402                  ADD     R0,DVNSR        ;
006212  013737  001402  001404          MOV     DVNSR,RESV16    ;RESERVED REG
006220  060037  001404                  ADD     R0,RESV16       ;

006224  013737  001352  001354          MOV     DVRVEC,DVRLVL   ;PTY LVL
006232  060037  001354                  ADD     R0,DVRLVL       ;
006236  013737  001354  001356          MOV     DVRLVL,DVTVEC   ;TX VEC
006244  060037  001356                  ADD     R0,DVTVEC       ;
006250  013737  001356  001360          MOV     DVTVEC,DVTLVL   ;TX LVL
006256  060037  001360                  ADD     R0,DVTLVL

006262  012700  001416                  MOV     #L00.03,R0      ;LOAD STAUS 00-03
006266  012701  001406                  MOV     #MASK.A,R1      ;PREPARE MASK.
006272  012702  001412                  MOV     #CLK.A,R2       ;PREPARE CLOCKS
006276  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

006302  012700  001420                  MOV     #L04.07,R0      ;LOAD STAUS 00-03
006306  012701  001407                  MOV     #MASK.B,R1      ;PREPARE MASK.
006312  012702  001413                  MOV     #CLK.B,R2       ;PREPARE CLOCKS
006316  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

006322  012700  001422                  MOV     #L08.11,R0      ;LOAD STAUS 00-03
006326  012701  001410                  MOV     #MASK.C,R1      ;PREPARE MASK.
006332  012702  001414                  MOV     #CLK.C,R2       ;PREPARE CLOCKS
006336  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.

006342  012700  001424                  MOV     #L12.15,R0      ;LOAD STAUS 00-03
006346  012701  001411                  MOV     #MASK.D,R1      ;PREPARE MASK.
006352  012702  001415                  MOV     #CLK.D,R2       ;PREPARE CLOCKS
006356  004737  006516                  JSR     PC,FIX.00       ;GO AND CALCULATE CONFIGURATION.
006362  032777  000002  172612          BIT     #SW01,@SWR
006370  001445                          BEQ     7$
006372                          4$:
006372  005737  000042                  TST     @#42
006376  001042                          BNE     7$
006400  104402  005104                  TYPE    ,MCRLF
006404  104403                          INSTR
006406  005366                          MTSTN
006410  104405                          PARAM
006412  000001                          1
006414  001000                          1000
006416  001226                          TSTNO
006420     000                  .BYTE   0
006421     001                  .BYTE   1
006422  012700  007256                  MOV     #TST1,R0
006426  022710                  5$:     CMP     (PC)+,(R0)
006430  012737                          MOV     (PC)+,@(PC)+
006432  001015                          BNE     6$
006434  023760  001226  000002          CMP     TSTNO,2(R0)
006442  001011                          BNE     6$
006444  022760  001226  000004          CMP     #TSTNO,4(R0)
006452  001005                          BNE     6$
006454  010037  001214                  MOV     R0,RETURN
```

```
ØØ646Ø  1Ø44Ø2  ØØ51Ø4                   TYPE    ,MCRLF
ØØ6464  ØØØ412                           BR      8$
ØØ6466  ØØ572Ø                   6$:     TST     (RØ)+
ØØ647Ø  Ø2ØØ27  Ø31174                   CMP     RØ,#TLAST+1Ø
ØØ6474  ØØ1354                           BNE     5$
ØØ6476  1Ø44Ø2  ØØ51ØØ                   TYPE    ,MQM
ØØ65Ø2  ØØØ733                           BR      4$
ØØ65Ø4  Ø12737  ØØ7256  ØØ1214   7$:     MOV     #TST1,RETURN       ;PREPARE RETURN ADDRESS
ØØ6512  ØØØ177  172476           8$:     JMP     @RETURN            ;GO START TESTING.

ØØ6516  Ø11ØØ3                   FIX.ØØ: MOV     (RØ),R3            ;GET PARAMETERS.
ØØ652Ø  Ø427Ø3  176377                   BIC     #^C<14ØØ>,R3       ;CLEAR JUNK.
ØØ6524  ØØ57Ø3                           TST     R3                 ;TEST FOR EIGHT BITS.
ØØ6526  ØØ1ØØ4                           BNE     1$                 ;BR IF NOT 8 BITS.
ØØ653Ø  1Ø5Ø11                           CLRB    (R1)               ;SET
ØØ6532  112712  ØØØØ1Ø                   MOVB    #8.,(R2)           ;
ØØ6536  ØØØ424                           BR      4$
ØØ654Ø  Ø227Ø3  ØØØ4ØØ           1$:     CMP     #4ØØ,R3            ;CHECK FOR SEVEN BITS.
ØØ6544  ØØ1ØØ5                           BNE     2$                 ;BR IF NOT 7 BITS.
ØØ6546  112711  ØØØ2ØØ                   MOVB    #2ØØ,(R1)          ;
ØØ6552  112712  ØØØØØ7                   MOVB    #7,(R2)            ;
ØØ6556  ØØØ414                           BR      4$
ØØ656Ø  Ø227Ø3  ØØ1ØØØ           2$:     CMP     #1ØØØ,R3           ;CHECK FOR SIX BITS.
ØØ6564  ØØ1ØØ5                           BNE     3$                 ;BR IF NOT SIX BITS.
ØØ6566  112711  ØØØ3ØØ                   MOVB    #3ØØ,(R1)          ;
ØØ6572  112712  ØØØØØ6                   MOVB    #6,(R2)            ;
ØØ6576  ØØØ4Ø4                           BR      4$
ØØ66ØØ  112711  ØØØ34Ø           3$:     MOVB    #34Ø,(R1)          ;IF NONE OF THE ABOVE; MUST BE 5 BITS.
ØØ66Ø4  112712  ØØØØØ5                   MOVB    #5,(R2)            ;
ØØ661Ø  Ø3271Ø  Ø4ØØØØ           4$:     BIT     #PARBIT,(RØ)       ;PARITY ENABLED?
ØØ6614  ØØ14Ø1                           BEQ     5$                 ;IF =Ø; THEN NO PARITY.
ØØ6616  1Ø5212                           INCB    (R2)               ;PLUS ONE TO THE CLOCK!
ØØ662Ø  ØØØ2Ø7                   5$:     RTS     PC                 ;

                                 ;*ROUTINE USED TO "AUTO SIZE" THE DV11
                                 ;*CSR AND VECTOR.
                                 ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
                                 ;*        ADDRESS RANGE (175ØØØ:1754ØØ)
                                 ;*        AND THE VECTOR MAY BE ANY WHERE IN THE
                                 ;*        FLOATING VECTOR RANGE (3ØØ:77Ø)
                                 ;*

ØØ6622                           AUTO.SIZE:
ØØ6622  ØØØØØ5                           RESET                      ;INSURE A BUS INIT.
ØØ6624  Ø127Ø2  ØØ15ØØ           CSRMAP: MOV     #DV.MAP,R2         ;LOAD MAP POINTER.
ØØ663Ø  ØØ5Ø22                   1$:     CLR     (R2)+              ;ZERO ENTIRE MAP
ØØ6632  Ø227Ø2  ØØ174Ø                   CMP     #DV.END,R2         ;ALL DONE?
ØØ6636  ØØ1374                           BNE     1$                 ;BR IF NO
ØØ664Ø  1Ø5Ø37  ØØ13Ø1                   CLRB    DVNUM              ;SET OCTAL NUMBER OF DV11'S TO Ø
ØØ6644  Ø127Ø2  ØØ15ØØ                   MOV     #DV.MAP,R2
ØØ665Ø  Ø127Ø1  175ØØØ                   MOV     #175ØØØ,R1         ;SET FOR FIRST ADDRESS TO BE TESTED
ØØ6654  Ø12737  ØØ7Ø74  ØØØØØ4           MOV     #6$,@#4            ;SET FOR NON-EXISTANT DEVICE TIME OUT
ØØ6662  ØØ5711                   2$:     TST     (R1)               ;IF DV11 DVSCR S/B Ø
ØØ6664  ØØ1Ø37                           BNE     3$                 ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DV11
ØØ6666  Ø22761  177777  ØØØØ12           CMP     #177777,12(R1)     ;IF DV11 THEN DVSFR S/B ALL 1'S ON INIT!
ØØ6674  ØØ1Ø33                           BNE     3$                 ;BR IF NOT DV11
```

```
ØØ6676  ØØ5761  ØØØØ16                   TST     16(R1)             ;IF DV11 THEN RESV16 S/B ALL Ø'S
ØØ67Ø2  ØØ1Ø3Ø                           BNE     3$                 ;BR IF NOT DV11
                                 ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
ØØ67Ø4  Ø1Ø122                           MOV     R1,(R2)+           ;STORE CSR IN CORE TABLE.
ØØ67Ø6  ØØ5722                           TST     (R2)+              ;POP OVER VECTOR STORE AREA
ØØ671Ø  Ø52722  ØØØ226                   BIS     #226,(R2)+         ;SET LINE CARD 1 STAT AND SYNC
ØØ6714  Ø52722  ØØØØ62                   BIS     #62,(R2)+          ;
ØØ672Ø  Ø52722  ØØØ226                   BIS     #226,(R2)+         ;SET LINE CARD 2 STAT AND SYNC
ØØ6724  Ø52722  ØØØØ62                   BIS     #62,(R2)+          ;
ØØ673Ø  Ø52722  ØØØ226                   BIS     #226,(R2)+         ;SET LINE CARD 3 STAT AND SYNC
ØØ6734  Ø52722  ØØØØ62                   BIS     #62,(R2)+          ;
ØØ674Ø  Ø52722  ØØØ226                   BIS     #226,(R2)+         ;SET LINE CARD 4 STAT AND SYNC
ØØ6744  Ø52722  ØØØØ62                   BIS     #62,(R2)+          ;
ØØ675Ø  1Ø5237  ØØ13Ø1                   INCB    DVNUM              ;UPDATE DEVICE COUNTER
ØØ6754  122737  ØØØØ1Ø  ØØ13Ø1           CMPB    #1Ø,DVNUM          ;ARE MAX. NO. OF DEV FOUND?
ØØ6762  ØØ14Ø5                           BEQ     1ØØ$               ;YES DON'T LOOK FOR ANY MORE.
ØØ6764  Ø627Ø1  ØØØØ1Ø           3$:     ADD     #1Ø,R1             ;UPDATE CSR POINTER ADDRESS
ØØ677Ø  Ø227Ø1  1754ØØ                   CMP     #1754ØØ,R1
ØØ6774  ØØ1332                           BNE     2$                 ;BR IF MORE ADDRESS TO CHECK.
ØØ6776  Ø12722  177777           1ØØ$:   MOV     #177777,(R2)+      ;TERMINATER.
ØØ7ØØ2  1Ø5Ø37  ØØ13ØØ                   CLRB    DVACTV
ØØ7ØØ6  1Ø5737  ØØ13Ø1                   TSTB    DVNUM              ;WERE ANY DV11'S FOUND AT ALL?
ØØ7Ø12  ØØ1423                           BEQ     5$                 ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
ØØ7Ø14  1137Ø1  ØØ13Ø1                   MOVB    DVNUM,R1
ØØ7Ø2Ø  11Ø137  ØØ13Ø3                   MOVB    R1,SAVNUM          ;SAVE NUMBER OF DEVICES
ØØ7Ø24  ØØØ241                   4$:     CLC
ØØ7Ø26  1Ø6137  ØØ13ØØ                   ROLB    DVACTV             ;GENERATE ACTIVE REGISTER OF DEVICES.
ØØ7Ø32  1Ø5237  ØØ13ØØ                   INCB    DVACTV             ;SET THE BIT
ØØ7Ø36  ØØ53Ø1                           DEC     R1
ØØ7Ø4Ø  ØØ1371                           BNE     4$                 ;BR IF MORE TO GENERATE
ØØ7Ø42  Ø12737  ØØØØØ6  ØØØØØ4           MOV     #6,@#4             ;RESTORE TRAP VECTOR
ØØ7Ø5Ø  113737  ØØ13ØØ  ØØ13Ø2           MOVB    DVACTV,SAVACT      ;SAVE ACTIVE REGISTER
ØØ7Ø56  ØØØ137  ØØ71Ø2                   JMP     VECMAP             ;GO FIND THE VECTOR NOW.
ØØ7Ø62  1Ø44Ø2  ØØ5174           5$:     TYPE    ,MERR2             ;NOTIFY OPR THAT NO DV11'S FOUND.
ØØ7Ø66  ØØ5ØØØ                           CLR     RØ                 ;MAKE DATA LIGHTS ZERO
ØØ7Ø7Ø  ØØØØØØ                           HALT                       ;STOP THE SHOW
ØØ7Ø72  ØØØ776                           BR      .-2                ;DISABLE CONT. SW.
ØØ7Ø74  Ø12716  ØØ6764           6$:     MOV     #3$,(SP)           ;ENTERED BY NON-EXISTANT TIME-OUT.
ØØ71ØØ  ØØØØØ2                           RTI                        ;RETURN TO MAINSTREAM

ØØ71Ø2  Ø12737  ØØØ34Ø  ØØØØ22   VECMAP: MOV     #34Ø,@#22          ;SET IOT TRAP PRIO TO 7
ØØ711Ø  Ø12737  ØØ7232  ØØØØ2Ø           MOV     #4$,@#2Ø           ;SET IOT TRAP VECTOR
ØØ7116  Ø127Ø2  ØØ15ØØ                   MOV     #DV.MAP,R2         ;SET SOFTWARE POINTER
ØØ7122  Ø127ØØ  ØØØ3ØØ                   MOV     #3ØØ,RØ            ;FLOATING VECTORS START HERE.
ØØ7126  Ø127Ø1  ØØØ3Ø2                   MOV     #3Ø2,R1            ;PC OF IOT INSTR.
ØØ7132  Ø1Ø12Ø                   1$:     MOV     R1,(RØ)+           ;START FILLING VECTOR AREA
ØØ7134  Ø12721  ØØØØØ4                   MOV     #4,(R1)+           ;WITH .+2; IOT
ØØ714Ø  Ø22Ø21                           CMP     (RØ)+,(R1)+        ;ADD 2 TO RØ +R1
ØØ7142  Ø2Ø127  ØØ1ØØØ                   CMP     R1,#1ØØØ
ØØ7146  1Ø1771                           BLOS    1$                 ;BR IF MORE TO FILL
ØØ715Ø  113737  ØØ13ØØ  ØØ1246           MOVB    DVACTV,TEMP1       ;STORE TEMPORALLY
ØØ7156  ØØ6Ø37  ØØ1246           2$:     ROR     TEMP1              ;BRING OUT A BIT
ØØ7162  1Ø3Ø34                           BCC     5$                 ;BR IF ALL DONE
ØØ7164  ØØ5Ø37  177776                   CLR     PS                 ;ZERO CPU PRIO
ØØ717Ø  Ø12772  ØØ13ØØ  ØØØØØØ           MOV     #BIT9+BIT7+BIT6,@(R2)
ØØ7176  ØØ5ØØØ                           CLR     RØ                 ;ATTEMPT TO FORCE AN INTERUPT
```

```
007200  005200                          INC     R0              ;STALL
007202  001376                          BNE     .-2             ;       FOR TIME TO INTERUPT
007204  052762  000300  000002          BIS     #300,2(R2)      ;NO INTERUPT ASSUME 300 AND FIX DV11 LATER
007212  042772  176777  000000  3$:     BIC     #^C<BIT9>,@(R2)
007220  005072  000000                  CLR     @(R2)
007224  062702  000024                  ADD     #24,R2          ;POP SOFTWARE POINTER
007230  000752                          BR      2$              ;KEEP GOING
007232  051662  000002          4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
007236  042762  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
007244  022626                          CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
007246  012716  007212                  MOV     #3$,(SP)        ;SET FOR RETURN
007252  000002                          RTI
007254  000207                  5$:     RTS     PC              ;ALL DONE WITH "AUTO SIZING"
```

```
                                        ;*********************** TEST 1 ******************************
                                        ;*VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE
                                        ;*A TIME-OUT TRAP.
                                        ;;*************************************************************


                                ;       TEST 1
                                ;---------------
007256  012737  000001  001226  TST1:   MOV     #1,TSTNO
007264  012737  007366  001216          MOV     #TST2,NEXT
007272  012737  007324  001220          MOV     #1$,LOCK
007300  012700  000010                  MOV     #8.,R0          ;SET FOR MAX. 8 PRI. REGISTERS
007304  013701  001362                  MOV     DVSCR,R1        ;GET FIRST PRI. ADDRESS
007310  012737  007360  000004          MOV     #2$,4           ;SET FOR TIME-OUT TRAP.
007316  012737  000340  000006          MOV     #340,6          ;SAFE GUARD.
007324  005711                  1$:     TST     (R1)            ;REFERENCE THE ADDRESS.
007326  000240                          NOP                     ;STALL
007330  000240                          NOP                     ;       FOR TIME.
007332  104401                          SCOP1                   ;IF SW09=1; GOTO 1$
007334  062701  000002                  ADD     #2,R1           ;UPDATE TO NEXT ADDRESS.
007340  005300                          DEC     R0              ;ARE ALL ADDRESS CHECKED?
007342  001370                          BNE     1$              ;BR IF NO.
007344  012737  000006  000004          MOV     #6,4            ;RESET TRAP ZONE.
007352  005037  000006                  CLR     @#6             ;
007356  104400                          SCOPE                   ;SCOPE THIS TEST
007360  011602                  2$:     MOV     (SP),R2         ;SAVE THE TRAP PC
007362  104001                          HLT     1               ;REPORT TIME-OUT TRAP
007364  000002                          RTI                     ;RETURN TO MAIN PROGRAM


                                        ;*********************** TEST 2 ******************************
                                        ;*PRIMARY REGISTER ADDRESSING TEST
                                        ;*LOAD EACH PRIMARY REGISTER WITH A
                                        ;*DIFFERENT NUMBER AND VERIFY EACH
                                        ;*WAS INDIVIDUALLY ADDRESSED.
                                        ;;*************************************************************


                                ;       TEST 2
                                ;---------------
007366  012737  000002  001226  TST2:   MOV     #2,TSTNO
007374  012737  007620  001216          MOV     #TST3,NEXT
007402  012737  007436  001220          MOV     #1$,LOCK
007410  012700  007600                  MOV     #3$,R0          ;SET DATA TABLE POINTER
007414  013703  001362                  MOV     DVSCR,R3        ;SET DV POINTER
007420  005013                          CLR     (R3)            ;START REG AT ZERO
007422  005077  171744                  CLR     @DVSRS          ;ZERO SEC. REG SEL.
007426  005077  171744                  CLR     @DVSRA          ;ZERO SEC REG ACCESS
007432  012702  000010                  MOV     #8.,R2          ;SET FOR EIGHT PRIMARY REGISTERS.
007436  011005                  1$:     MOV     (R0),R5         ;PUT DATA INTO EXPECTED
007440  010513                          MOV     R5,(R3)         ;WRITE EXPECTED INTO DV REGISTER
007442  011304                          MOV     (R3),R4         ;READ REGISTER INTO FOUND LOC
007444  020504                          CMP     R5,R4           ;DOES EXPECTED=RECEIVED?
007446  001401                          BEQ     64$             ;BR IF YES
007450  104003                          HLT     3               ;THIS IS A DATA ERROR *NOT* A DUEL ADDRESSING ERROR. NOT
007452  104401                  64$:    SCOP1                   ;SW09=1?
007454  022023                          CMP     (R0)+,(R3)+     ;POP DATA POINTERS AND HRDW POINTER
007456  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR!
```

```
007462  001002                          BNE     4$              ;BR IF NOT DVNSR
007464  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS AROUND DVNSR
007466  005302                          DEC     R2              ;UPDATE REGISTER COUNTER
007470  020337  001370          4$:     CMP     R3,DVLCR        ;DON'T DO THE DVLCR!
007474  001002                          BNE     6$              ;BR IF NOT THE DVLCR
007476  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS AROUND THE DVLCR
007500  005302                          DEC     R2              ;UPDATE THE REGISTER COUNTER
007502  005302                  6$:     DEC     R2              ;DITTO
007504  001354                          BNE     1$              ;BR IF MORE TO GO
                                ;CHECK DUEL ADDRESSING......
007506  012700  007600                  MOV     #3$,R0          ;SET DATA POINTER
007512  013703  001362                  MOV     DVSCR,R3        ;SET HRDW POINTER
007516  012737  007530  001220          MOV     #2$,LOCK        ;SET IF SW09=1
007524  012702  000010                  MOV     #8.,R2          ;SET EIGHT PRIMARY REGISTERS
007530  011005                  2$:     MOV     (R0),R5         ;LOAD DATA INTO EXPECTED
007532  011304                          MOV     (R3),R4         ;READ THE DV REGISTER
007534  020504                          CMP     R5,R4           ;DOES THE DATA COMPARE
007536  001401                          BEQ     65$             ;BR IF OK
007540  104003                          HLT     3               ;NOW THIS WAS A DUEL ADDRESSING ERROR.
007542  104401                  65$:    SCOP1                   ;SW09=1?
007544  022023                          CMP     (R0)+,(R3)+     ;POP POINTERS
007546  020337  001402                  CMP     R3,DVNSR        ;DON'T DO THE DVNSR
007552  001002                          BNE     5$              ;BR IF NOT DVNSR
007554  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS
007556  005302                          DEC     R2              ;SET REG COUNTER
007560  020337  001370          5$:     CMP     R3,DVLCR        ;DON'T DO THE DVLCR
007564  001002                          BNE     7$              ;BR IF NOT DVLCR
007566  022320                          CMP     (R3)+,(R0)+     ;POP POINTERS
007570  005302                          DEC     R2              ;SET REG POINTER
007572  005302                  7$:     DEC     R2              ;DITTO
007574  001355                          BNE     2$              ;BR IF MORE TO GO
007576  104400                          SCOPE                   ;SCOPE THIS TEST
007600  000010                  3$:     .WORD   000010          ;DVSCR
007602  000000                          .WORD   000000          ;DVRIC
007604  000000                          .WORD   SKIP            ;DVLCR
007606  001400                          .WORD   001400          ;DVSRS
007610  000300                          .WORD   000300          ;DVSRA
007612  100000                          .WORD   100000          ;DVSFR
007614  000000                          .WORD   SKIP            ;DVNSR
007616  000060                          .WORD   000060          ;RESV16


                                        ;*********************** TEST 3 ******************************
                                        ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                                        ;*SET BIT2, VERIFY BIT2 WAS SET.
                                        ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 3
                                ;--------------
007620  012737  000003  001226  TST3:   MOV     #3,TSTNO
007626  012737  007674  001216          MOV     #TST4,NEXT
007634  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
007640  012705  000004                  MOV     #BIT2,R5        ;SET "EXPECTED ".
007644  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
007646  011304                          MOV     (R3),R4         ;READ THE REGISTER.
```

```
007650  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
007652  001401                          BEQ     1$              ;ARE THEY THE SAME?
007654  104003                          HLT     3               ;COMPARISON ERROR.
007656  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT2
007660  011304                          MOV     (R3),R4         ;READ THE REGISTER.
007662  005005                          CLR     R5              ;SET "EXPECTED"
007664  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
007666  001401                          BEQ     2$              ;BR IF OK
007670  104003                          HLT     3               ;COMPARISON ERROR
007672  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 4 ******************************
                                        ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                                        ;*SET BIT3, VERIFY BIT3 WAS SET.
                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 4
                                ;--------------
007674  012737  000004  001226  TST4:   MOV     #4,TSTNO
007702  012737  007750  001216          MOV     #TST5,NEXT
007710  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
007714  012705  000010                  MOV     #BIT3,R5        ;SET "EXPECTED ".
007720  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
007722  011304                          MOV     (R3),R4         ;READ THE REGISTER.
007724  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
007726  001401                          BEQ     1$              ;ARE THEY THE SAME?
007730  104003                          HLT     3               ;COMPARISON ERROR.
007732  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3
007734  011304                          MOV     (R3),R4         ;READ THE REGISTER.
007736  005005                          CLR     R5              ;SET "EXPECTED"
007740  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
007742  001401                          BEQ     2$              ;BR IF OK
007744  104003                          HLT     3               ;COMPARISON ERROR
007746  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 5 ******************************
                                        ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                                        ;*SET BIT4, VERIFY BIT4 WAS SET.
                                        ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 5
                                ;--------------
007750  012737  000005  001226  TST5:   MOV     #5,TSTNO
007756  012737  010024  001216          MOV     #TST6,NEXT
007764  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED.
007770  012705  000020                  MOV     #BIT4,R5        ;SET "EXPECTED ".
007774  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
007776  011304                          MOV     (R3),R4         ;READ THE REGISTER.
010000  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
010002  001401                          BEQ     1$              ;ARE THEY THE SAME?
010004  104003                          HLT     3               ;COMPARISON ERROR.
010006  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT4
```

```
010010  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010012  005005                           CLR     R5               ;SET "EXPECTED"
010014  020504                           CMP     R5,R4            ;R5=GOOD; R4=?
010016  001401                           BEQ     2$               ;BR IF OK
010020  104003                           HLT     3                ;COMPARISON ERROR
010022  104400                   2$:     SCOPE                    ;SCOPE THIS TEST


                                         ;*********************** TEST 6 ******************************
                                         ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                                         ;*SET BIT5, VERIFY BIT5 WAS SET.
                                         ;*CLEAR BIT5, VERIFY BIT5 WAS CLEARED.
                                         ;;*************************************************************

                                 ;  TEST 6
                                 ;--------------
010024  012737  000006  001226   TST6:   MOV     #6,TSTNO
010032  012737  010100  001216           MOV     #TST7,NEXT
010040  013703  001362                   MOV     DVSCR,R3         ;SET REGISTER TO BE TESTED.
010044  012705  000040                   MOV     #BIT5,R5         ;SET "EXPECTED ".
010050  010513                           MOV     R5,(R3)          ;WRITE THE REGISTER.
010052  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010054  020504                           CMP     R5,R4            ;R5=GOOD; R4=UNKNOWN.
010056  001401                           BEQ     1$               ;ARE THEY THE SAME?
010060  104003                           HLT     3                ;COMPARISON ERROR.
010062  040513                   1$:     BIC     R5,(R3)          ;CLEAR BIT5
010064  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010066  005005                           CLR     R5               ;SET "EXPECTED"
010070  020504                           CMP     R5,R4            ;R5=GOOD; R4=?
010072  001401                           BEQ     2$               ;BR IF OK
010074  104003                           HLT     3                ;COMPARISON ERROR
010076  104400                   2$:     SCOPE                    ;SCOPE THIS TEST


                                         ;*********************** TEST 7 ******************************
                                         ;*SYSTEM CONTROL REGSTER READ/WRITE TEST.
                                         ;*SET BIT6, VERIFY BIT6 WAS SET.
                                         ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
                                         ;;*************************************************************

                                 ;  TEST 7
                                 ;--------------
010100  012737  000007  001226   TST7:   MOV     #7,TSTNO
010106  012737  010154  001216           MOV     #TST10,NEXT
010114  013703  001362                   MOV     DVSCR,R3         ;SET REGISTER TO BE TESTED.
010120  012705  000100                   MOV     #BIT6,R5         ;SET "EXPECTED ".
010124  010513                           MOV     R5,(R3)          ;WRITE THE REGISTER.
010126  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010130  020504                           CMP     R5,R4            ;R5=GOOD; R4=UNKNOWN.
010132  001401                           BEQ     1$               ;ARE THEY THE SAME?
010134  104003                           HLT     3                ;COMPARISON ERROR.
010136  040513                   1$:     BIC     R5,(R3)          ;CLEAR BIT6
010140  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010142  005005                           CLR     R5               ;SET "EXPECTED"
010144  020504                           CMP     R5,R4            ;R5=GOOD; R4=?
010146  001401                           BEQ     2$               ;BR IF OK
```

```
010150  104003                           HLT     3                ;COMPARISON ERROR
010152  104400                   2$:     SCOPE                    ;SCOPE THIS TEST


                                         ;*********************** TEST 10 *****************************
                                         ;*TEST THAT BIT 7(RECEIVER INTERUPT)
                                         ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                         ;*IS NOT SET.
                                         ;*THEN VERIFY THAT BIT 7 CAN BE WRITTEN WHEN
                                         ;*BIT 9 IS SET.
                                         ;;*************************************************************

                                 ;  TEST 10
                                 ;--------------
010154  012737  000010  001226   TST10:  MOV     #10,TSTNO
010162  012737  010250  001216           MOV     #TST11,NEXT
010170  013703  001362                   MOV     DVSCR,R3         ;SET REGISTER POINTER INTO R3
010174  005005                           CLR     R5               ;SET EXPECTED RESULT
010176  012713  000200                   MOV     #BIT7,(R3)       ;ATTEMPT TO SET BIT 7
010202  011304                           MOV     (R3),R4          ;READ BACK REGISTER
010204  001401                           BEQ     1$               ;BIT 7 SHOULD NOT BE SET
010206  104003                           HLT     3                ;DVSCR NOT ALL 0'S
010210  012705  001200           1$:     MOV     #BIT7+BIT9,R5    ;SET BIT 7+BIT9 IN THE DVSCR
010214  010513                           MOV     R5,(R3)          ;
010216  011304                           MOV     (R3),R4          ;READ REGISTER
010220  020504                           CMP     R5,R4            ;IS BIT 9 AND BIT 7 SET?
010222  001401                           BEQ     2$               ;BR IF OK
010224  104003                           HLT     3                ;DVSCR ERROR
010226  042705  000200           2$:     BIC     #BIT7,R5         ;CLEAR BIT 7
010232  042713  000200                   BIC     #BIT7,(R3)       ;DITTO
010236  011304                           MOV     (R3),R4          ;READ REGISTER
010240  020504                           CMP     R5,R4            ;COMPARE OK?
010242  001401                           BEQ     3$               ;BR IF YES
010244  104003                           HLT     3                ;DVSCR ERROR
010246  104400                   3$:     SCOPE                    ;SCOPE THIS TEST


                                         ;*********************** TEST 11 *****************************
                                         ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                         ;*SET BIT8, VERIFY BIT8 WAS SET.
                                         ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
                                         ;;*************************************************************

                                 ;  TEST 11
                                 ;--------------
010250  012737  000011  001226   TST11:  MOV     #11,TSTNO
010256  012737  010324  001216           MOV     #TST12,NEXT
010264  013703  001362                   MOV     DVSCR,R3         ;SET REGISTER TO BE TESTED.
010270  012705  000400                   MOV     #BIT8,R5         ;SET "EXPECTED ".
010274  010513                           MOV     R5,(R3)          ;WRITE THE REGISTER.
010276  011304                           MOV     (R3),R4          ;READ THE REGISTER.
010300  020504                           CMP     R5,R4            ;R5=GOOD; R4=UNKNOWN.
010302  001401                           BEQ     1$               ;ARE THEY THE SAME?
010304  104003                           HLT     3                ;COMPARISON ERROR.
010306  040513                   1$:     BIC     R5,(R3)          ;CLEAR BIT8
010310  011304                           MOV     (R3),R4          ;READ THE REGISTER.
```

```
010312  005005                          CLR     R5                  ;SET "EXPECTED"
010314  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
010316  001401                          BEQ     2$                  ;BR IF OK
010320  104003                          HLT     3                   ;COMPARISON ERROR
010322  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;********************** TEST 12 *****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT9, VERIFY BIT9 WAS SET.
                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.
                                        ;*************************************************************

                                ;  TEST 12
                                ;--------------
010324  012737  000012  001226  TST12:  MOV     #12,TSTNO
010332  012737  010400  001216          MOV     #TST13,NEXT
010340  013703  001362                  MOV     DVSCR,R3            ;SET REGISTER TO BE TESTED.
010344  012705  001000                  MOV     #BIT9,R5            ;SET "EXPECTED ".
010350  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER.
010352  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010354  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN.
010356  001401                          BEQ     1$                  ;ARE THEY THE SAME?
010360  104003                          HLT     3                   ;COMPARISON ERROR.
010362  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT9
010364  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010366  005005                          CLR     R5                  ;SET "EXPECTED"
010370  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
010372  001401                          BEQ     2$                  ;BR IF OK
010374  104003                          HLT     3                   ;COMPARISON ERROR
010376  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;********************** TEST 13 *****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
                                        ;*************************************************************

                                ;  TEST 13
                                ;--------------
010400  012737  000013  001226  TST13:  MOV     #13,TSTNO
010406  012737  010454  001216          MOV     #TST14,NEXT
010414  013703  001362                  MOV     DVSCR,R3            ;SET REGISTER TO BE TESTED.
010420  012705  002000                  MOV     #BIT10,R5           ;SET "EXPECTED ".
010424  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER.
010426  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010430  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN.
010432  001401                          BEQ     1$                  ;ARE THEY THE SAME?
010434  104003                          HLT     3                   ;COMPARISON ERROR.
010436  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT10
010440  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010442  005005                          CLR     R5                  ;SET "EXPECTED"
010444  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
010446  001401                          BEQ     2$                  ;BR IF OK
010450  104003                          HLT     3                   ;COMPARISON ERROR
```

```
010452  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;********************** TEST 14 *****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT12, VERIFY BIT12 WAS SET.
                                        ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
                                        ;*************************************************************

                                ;  TEST 14
                                ;--------------
010454  012737  000014  001226  TST14:  MOV     #14,TSTNO
010462  012737  010530  001216          MOV     #TST15,NEXT
010470  013703  001362                  MOV     DVSCR,R3            ;SET REGISTER TO BE TESTED.
010474  012705  010000                  MOV     #BIT12,R5           ;SET "EXPECTED ".
010500  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER.
010502  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010504  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN.
010506  001401                          BEQ     1$                  ;ARE THEY THE SAME?
010510  104003                          HLT     3                   ;COMPARISON ERROR.
010512  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT12
010514  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010516  005005                          CLR     R5                  ;SET "EXPECTED"
010520  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
010522  001401                          BEQ     2$                  ;BR IF OK
010524  104003                          HLT     3                   ;COMPARISON ERROR
010526  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;********************** TEST 15 *****************************
                                        ;*SYSTEM CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT13, VERIFY BIT13 WAS SET.
                                        ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
                                        ;*************************************************************

                                ;  TEST 15
                                ;--------------
010530  012737  000015  001226  TST15:  MOV     #15,TSTNO
010536  012737  010604  001216          MOV     #TST16,NEXT
010544  013703  001362                  MOV     DVSCR,R3            ;SET REGISTER TO BE TESTED.
010550  012705  020000                  MOV     #BIT13,R5           ;SET "EXPECTED ".
010554  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER.
010556  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010560  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN.
010562  001401                          BEQ     1$                  ;ARE THEY THE SAME?
010564  104003                          HLT     3                   ;COMPARISON ERROR.
010566  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT13
010570  011304                          MOV     (R3),R4             ;READ THE REGISTER.
010572  005005                          CLR     R5                  ;SET "EXPECTED"
010574  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
010576  001401                          BEQ     2$                  ;BR IF OK
010600  104003                          HLT     3                   ;COMPARISON ERROR
010602  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;********************** TEST 16 *****************************
```

```
                                                ;*SYSTEM CONTROL REGISTER READ/WRITE TEST,
                                                ;*SET BIT3+BIT0, VERIFY BIT3+BIT0 WAS SET,
                                                ;*CLEAR BIT3+BIT0, VERIFY BIT3+BIT0 WAS CLEARED,
                                                ;;*************************************************************

                                        ;  TEST 16
                                        ;--------------
010604  012737  000016  001226  TST16:  MOV     #16,TSTNO
010612  012737  010660  001216          MOV     #TST17,NEXT
010620  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER TO BE TESTED,
010624  012705  000011                  MOV     #BIT3+BIT0,R5   ;SET "EXPECTED ",
010630  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER,
010632  011304                          MOV     (R3),R4         ;READ THE REGISTER,
010634  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN,
010636  001401                          BEQ     1$              ;ARE THEY THE SAME?
010640  104003                          HLT     3               ;COMPARISON ERROR,
010642  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3+BIT0
010644  011304                          MOV     (R3),R4         ;READ THE REGISTER,
010646  005005                          CLR     R5              ;SET "EXPECTED"
010650  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
010652  001401                          BEQ     2$              ;BR IF OK
010654  104003                          HLT     3               ;COMPARISON ERROR
010656  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                                ;*********************** TEST 17 ******************************
                                                ;*TEST THAT BIT 15(NPR STATUS OVERFLOW INTERUPT)
                                                ;*CANNOT BE WRITTEN WHEN BIT 9 (SYSTEM MAINTAINCE)
                                                ;*IS NOT SET,
                                                ;*THEN VERIFY THAT BIT 15 CAN BE WRITTEN WHEN
                                                ;*BIT 9 IS SET,
                                                ;;*************************************************************

                                        ;  TEST 17
                                        ;--------------
010660  012737  000017  001226  TST17:  MOV     #17,TSTNO
010666  012737  010754  001216          MOV     #TST20,NEXT
010674  013703  001362                  MOV     DVSCR,R3        ;SET REGISTER POINTER INTO R3
010700  005005                          CLR     R5              ;SET EXPECTED RESULT
010702  012713  100000                  MOV     #BIT15,(R3)     ;ATTEMPT TO SET BIT 15
010706  011304                          MOV     (R3),R4         ;READ BACK REGISTER
010710  001401                          BEQ     1$              ;BIT 15 SHOULD NOT BE SET
010712  104003                          HLT     3               ;DVSCR NOT ALL 0'S
010714  012705  101000          1$:     MOV     #BIT15+BIT9,R5  ;SET BIT 15+BIT9 IN THE DVSCR
010720  010513                          MOV     R5,(R3)         ;
010722  011304                          MOV     (R3),R4         ;READ REGISTER
010724  020504                          CMP     R5,R4           ;IS BIT 9 AND BIT 15 SET?
010726  001401                          BEQ     2$              ;BR IF OK
010730  104003                          HLT     3               ;DVSCR ERROR
010732  042705  100000          2$:     BIC     #BIT15,R5       ;CLEAR BIT 15
010736  042713  100000                  BIC     #BIT15,(R3)     ;DITTO
010742  011304                          MOV     (R3),R4         ;READ REGISTER
010744  020504                          CMP     R5,R4           ;COMPARE OK?
010746  001401                          BEQ     3$              ;BR IF YES
010750  104003                          HLT     3               ;DVSCR ERROR
010752  104400                  3$:     SCOPE                   ;SCOPE THIS TEST
```

```

                                                ;*********************** TEST 20 ******************************
                                                ;*TEST THAT BIT8 IN DVSCR CLEARS
                                                ;*BIT7 OF DVSCR,
                                                ;;*************************************************************

                                        ;  TEST 20
                                        ;--------------
010754  012737  000020  001226  TST20:  MOV     #20,TSTNO
010762  012737  011032  001216          MOV     #TST21,NEXT
010770  013703  001362                  MOV     DVSCR,R3        ;GET POINTER
010774  012705  000400                  MOV     #BIT8,R5        ;GET GOOD
011000  012713  000200                  MOV     #BIT7,(R3)      ;SET BIT7
011004  052713  000400                  BIS     #BIT8,(R3)      ;SET BIT8
011010  027777  170352  170350          CMP     @DVRIC,@DVRIC   ;WASTE TIME
011016  011304                          MOV     (R3),R4         ;READ SCR
011020  020504                          CMP     R5,R4           ;BIT8 ONLY SET?
011022  001401                          BEQ     1$              ;BR IF YES
011024  104003                          HLT     3               ;BIT8 NOT ONLY BIT SET OR 8 ISN'T SET!
011026  005013                  1$:     CLR     (R3)            ;LEAVE REG=0
011030  104400                          SCOPE                   ;SCOPE


                                                ;*********************** TEST 21 ******************************
                                                ;*RECEIVER INTERUPT CHARACTER REGISTER TEST
                                                ;*TEST THAT RECEIVER INTERUPT CHARACTER REGISTER CANNOT BE WRITTEN,
                                                ;*WRITE RECEIVER INTERUPT CHARACTER REGISTER WITH ALL 1'S
                                                ;*AND VERIFY THAT ALL 0'S ARE READ BACK,
                                                ;;*************************************************************

                                        ;  TEST 21
                                        ;--------------
011032  012737  000021  001226  TST21:  MOV     #21,TSTNO
011040  012737  011072  001216          MOV     #TST22,NEXT
011046  013703  001366                  MOV     DVRIC,R3
                                                                ;GET THE REGISTER,
011052  005005                          CLR     R5              ;SET EXPECTED (ZERO)
011054  012713  177777                  MOV     #-1,(R3)        ;WRITE REGISTER WITH ALL 1'S
011060  011304                          MOV     (R3),R4         ;READ THE REGISTER,
011062  020504                          CMP     R5,R4           ;IS THE REGISTER EQUAL TO ZERO,
011064  001401                          BEQ     1$              ;BR IF OK,
011066  104003                          HLT     3               ;REGISTER NOT ZERO,
011070  104400                  1$:     SCOPE                   ;SCOPE THIS TEST,


                                                ;*********************** TEST 22 ******************************
                                                ;*LINE CONTROL REGISTER READ/WRITE TEST,
                                                ;*SET BIT9, VERIFY BIT9 WAS SET,
                                                ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED,
                                                ;;*************************************************************

                                        ;  TEST 22
                                        ;--------------
011072  012737  000022  001226  TST22:  MOV     #22,TSTNO
011100  012737  011156  001216          MOV     #TST23,NEXT
011106  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED,
011112  012705  001000                  MOV     #BIT9,R5        ;SET "EXPECTED ".
011116  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011120  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011122  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011126  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011130  001401                          BEQ     1$              ;ARE THEY THE SAME?
011132  104003                          HLT     3               ;COMPARISON ERROR.
011134  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT9
011136  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011140  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011144  005005                          CLR     R5              ;SET "EXPECTED"
011146  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011150  001401                          BEQ     2$              ;BR IF OK
011152  104003                          HLT     3               ;COMPARISON ERROR
011154  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 23 *****************************
                                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT10, VERIFY BIT10 WAS SET.
                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 23
                                ;--------------
011156  012737  000023  001226  TST23:  MOV     #23,TSTNO
011164  012737  011242  001216          MOV     #TST24,NEXT
011172  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
011176  012705  002000                  MOV     #BIT10,R5       ;SET "EXPECTED ".
011202  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011204  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011206  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011212  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011214  001401                          BEQ     1$              ;ARE THEY THE SAME?
011216  104003                          HLT     3               ;COMPARISON ERROR.
011220  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT10
011222  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011224  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011230  005005                          CLR     R5              ;SET "EXPECTED"
011232  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011234  001401                          BEQ     2$              ;BR IF OK
011236  104003                          HLT     3               ;COMPARISON ERROR
011240  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 24 *****************************
                                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT11, VERIFY BIT11 WAS SET.
                                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 24
                                ;--------------
011242  012737  000024  001226  TST24:  MOV     #24,TSTNO
011250  012737  011326  001216          MOV     #TST25,NEXT
011256  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
011262  012705  004000                  MOV     #BIT11,R5       ;SET "EXPECTED ".
011266  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011270  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011272  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011276  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011300  001401                          BEQ     1$              ;ARE THEY THE SAME?
011302  104003                          HLT     3               ;COMPARISON ERROR.
011304  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT11
011306  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011310  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011314  005005                          CLR     R5              ;SET "EXPECTED"
011316  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011320  001401                          BEQ     2$              ;BR IF OK
011322  104003                          HLT     3               ;COMPARISON ERROR
011324  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 25 *****************************
                                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT12, VERIFY BIT12 WAS SET.
                                        ;*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 25
                                ;--------------
011326  012737  000025  001226  TST25:  MOV     #25,TSTNO
011334  012737  011412  001216          MOV     #TST26,NEXT
011342  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
011346  012705  010000                  MOV     #BIT12,R5       ;SET "EXPECTED ".
011352  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
011354  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011356  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011362  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
011364  001401                          BEQ     1$              ;ARE THEY THE SAME?
011366  104003                          HLT     3               ;COMPARISON ERROR.
011370  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT12
011372  011304                          MOV     (R3),R4         ;READ THE REGISTER.
011374  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011400  005005                          CLR     R5              ;SET "EXPECTED"
011402  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
011404  001401                          BEQ     2$              ;BR IF OK
011406  104003                          HLT     3               ;COMPARISON ERROR
011410  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 26 *****************************
                                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT13, VERIFY BIT13 WAS SET.
                                        ;*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.
                                        ;;***********************************************************

                                ;  TEST 26
                                ;--------------
011412  012737  000026  001226  TST26:  MOV     #26,TSTNO
011420  012737  011476  001216          MOV     #TST27,NEXT
011426  013703  001370                  MOV     DVLCR,R3        ;SET REGISTER TO BE TESTED.
```

```
011432  012705  020000                  MOV     #BIT13,R5         ;SET "EXPECTED ".
011436  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
011440  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011442  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011446  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
011450  001401                          BEQ     1$                ;ARE THEY THE SAME?
011452  104003                          HLT     3                 ;COMPARISON ERROR.
011454  040513                  1$:     BIC     R5,(R3)           ;CLEAR BIT13
011456  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011460  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011464  005005                          CLR     R5                ;SET "EXPECTED"
011466  020504                          CMP     R5,R4             ;R5=GOOD; R4=?
011470  001401                          BEQ     2$                ;BR IF OK
011472  104003                          HLT     3                 ;COMPARISON ERROR
011474  104400                  2$:     SCOPE                     ;SCOPE THIS TEST



                                        ;*********************** TEST 27 ******************************
                                        ;*LINE CONTROL REGISTER READ/WRITE TEST.
                                        ;*SET BIT14, VERIFY BIT14 WAS SET.
                                        ;*CLEAR BIT14, VERIFY BIT14 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 27
                                ;--------------
011476  012737  000027  001226  TST27:  MOV     #27,TSTNO
011504  012737  011562  001216          MOV     #TST30,NEXT
011512  013703  001370                  MOV     DVLCR,R3          ;SET REGISTER TO BE TESTED.
011516  012705  040000                  MOV     #BIT14,R5         ;SET "EXPECTED ".
011522  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
011524  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011526  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011532  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
011534  001401                          BEQ     1$                ;ARE THEY THE SAME?
011536  104003                          HLT     3                 ;COMPARISON ERROR.
011540  040513                  1$:     BIC     R5,(R3)           ;CLEAR BIT14
011542  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011544  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4 ;CLEAR UNWANTED BITS
011550  005005                          CLR     R5                ;SET "EXPECTED"
011552  020504                          CMP     R5,R4             ;R5=GOOD; R4=?
011554  001401                          BEQ     2$                ;BR IF OK
011556  104003                          HLT     3                 ;COMPARISON ERROR
011560  104400                  2$:     SCOPE                     ;SCOPE THIS TEST


                                        ;*********************** TEST 30 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT0, VERIFY BIT0 WAS SET.
                                        ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 30
                                ;--------------
011562  012737  000030  001226  TST30:  MOV     #30,TSTNO
011570  012737  011636  001216          MOV     #TST31,NEXT
011576  013703  001372                  MOV     DVSRS,R3          ;SET REGISTER TO BE TESTED.
```

```
011602  012705  000001                  MOV     #BIT0,R5          ;SET "EXPECTED ".
011606  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
011610  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011612  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
011614  001401                          BEQ     1$                ;ARE THEY THE SAME?
011616  104003                          HLT     3                 ;COMPARISON ERROR.
011620  040513                  1$:     BIC     R5,(R3)           ;CLEAR BIT0
011622  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011624  005005                          CLR     R5                ;SET "EXPECTED"
011626  020504                          CMP     R5,R4             ;R5=GOOD; R4=?
011630  001401                          BEQ     2$                ;BR IF OK
011632  104003                          HLT     3                 ;COMPARISON ERROR
011634  104400                  2$:     SCOPE                     ;SCOPE THIS TEST



                                        ;*********************** TEST 31 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT1, VERIFY BIT1 WAS SET.
                                        ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 31
                                ;--------------
011636  012737  000031  001226  TST31:  MOV     #31,TSTNO
011644  012737  011712  001216          MOV     #TST32,NEXT
011652  013703  001372                  MOV     DVSRS,R3          ;SET REGISTER TO BE TESTED.
011656  012705  000002                  MOV     #BIT1,R5          ;SET "EXPECTED ".
011662  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
011664  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011666  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
011670  001401                          BEQ     1$                ;ARE THEY THE SAME?
011672  104003                          HLT     3                 ;COMPARISON ERROR.
011674  040513                  1$:     BIC     R5,(R3)           ;CLEAR BIT1
011676  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011700  005005                          CLR     R5                ;SET "EXPECTED"
011702  020504                          CMP     R5,R4             ;R5=GOOD; R4=?
011704  001401                          BEQ     2$                ;BR IF OK
011706  104003                          HLT     3                 ;COMPARISON ERROR
011710  104400                  2$:     SCOPE                     ;SCOPE THIS TEST



                                        ;*********************** TEST 32 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT2, VERIFY BIT2 WAS SET.
                                        ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 32
                                ;--------------
011712  012737  000032  001226  TST32:  MOV     #32,TSTNO
011720  012737  011766  001216          MOV     #TST33,NEXT
011726  013703  001372                  MOV     DVSRS,R3          ;SET REGISTER TO BE TESTED.
011732  012705  000004                  MOV     #BIT2,R5          ;SET "EXPECTED ".
011736  010513                          MOV     R5,(R3)           ;WRITE THE REGISTER.
011740  011304                          MOV     (R3),R4           ;READ THE REGISTER.
011742  020504                          CMP     R5,R4             ;R5=GOOD; R4=UNKNOWN.
```

```
011744  001401                          BEQ     1$                  ;ARE THEY THE SAME?
011746  104003                          HLT     3                   ;COMPARISON ERROR,
011750  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT2
011752  011304                          MOV     (R3),R4             ;READ THE REGISTER,
011754  005005                          CLR     R5                  ;SET "EXPECTED"
011756  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
011760  001401                          BEQ     2$                  ;BR IF OK
011762  104003                          HLT     3                   ;COMPARISON ERROR
011764  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;*********************** TEST 33 *****************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST,
                                        ;*SET BIT3, VERIFY BIT3 WAS SET,
                                        ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED,
                                        ;;*************************************************************

                                ;  TEST 33
                                ;--------------
011766  012737  000033  001226  TST33:  MOV     #33,TSTNO
011774  012737  012042  001216          MOV     #TST34,NEXT
012002  013703  001372                  MOV     DVSRS,R3            ;SET REGISTER TO BE TESTED,
012006  012705  000010                  MOV     #BIT3,R5            ;SET "EXPECTED ",
012012  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER,
012014  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012016  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN,
012020  001401                          BEQ     1$                  ;ARE THEY THE SAME?
012022  104003                          HLT     3                   ;COMPARISON ERROR,
012024  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT3
012026  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012030  005005                          CLR     R5                  ;SET "EXPECTED"
012032  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
012034  001401                          BEQ     2$                  ;BR IF OK
012036  104003                          HLT     3                   ;COMPARISON ERROR
012040  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;*********************** TEST 34 *****************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST,
                                        ;*SET BIT8, VERIFY BIT8 WAS SET,
                                        ;*CLEAR BIT8, VERIFY BIT8 WAS CLEARED,
                                        ;;*************************************************************

                                ;  TEST 34
                                ;--------------
012042  012737  000034  001226  TST34:  MOV     #34,TSTNO
012050  012737  012116  001216          MOV     #TST35,NEXT
012056  013703  001372                  MOV     DVSRS,R3            ;SET REGISTER TO BE TESTED,
012062  012705  000400                  MOV     #BIT8,R5            ;SET "EXPECTED ",
012066  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER,
012070  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012072  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN,
012074  001401                          BEQ     1$                  ;ARE THEY THE SAME?
012076  104003                          HLT     3                   ;COMPARISON ERROR,
012100  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT8
012102  011304                          MOV     (R3),R4             ;READ THE REGISTER,
```

```
012104  005005                          CLR     R5                  ;SET "EXPECTED"
012106  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
012110  001401                          BEQ     2$                  ;BR IF OK
012112  104003                          HLT     3                   ;COMPARISON ERROR
012114  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;*********************** TEST 35 *****************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST,
                                        ;*SET BIT9, VERIFY BIT9 WAS SET,
                                        ;*CLEAR BIT9, VERIFY BIT9 WAS CLEARED,
                                        ;;*************************************************************

                                ;  TEST 35
                                ;--------------
012116  012737  000035  001226  TST35:  MOV     #35,TSTNO
012124  012737  012172  001216          MOV     #TST36,NEXT
012132  013703  001372                  MOV     DVSRS,R3            ;SET REGISTER TO BE TESTED,
012136  012705  001000                  MOV     #BIT9,R5            ;SET "EXPECTED ",
012142  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER,
012144  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012146  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN,
012150  001401                          BEQ     1$                  ;ARE THEY THE SAME?
012152  104003                          HLT     3                   ;COMPARISON ERROR,
012154  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT9
012156  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012160  005005                          CLR     R5                  ;SET "EXPECTED"
012162  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
012164  001401                          BEQ     2$                  ;BR IF OK
012166  104003                          HLT     3                   ;COMPARISON ERROR
012170  104400                  2$:     SCOPE                       ;SCOPE THIS TEST


                                        ;*********************** TEST 36 *****************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST,
                                        ;*SET BIT10, VERIFY BIT10 WAS SET,
                                        ;*CLEAR BIT10, VERIFY BIT10 WAS CLEARED,
                                        ;;*************************************************************

                                ;  TEST 36
                                ;--------------
012172  012737  000036  001226  TST36:  MOV     #36,TSTNO
012200  012737  012246  001216          MOV     #TST37,NEXT
012206  013703  001372                  MOV     DVSRS,R3            ;SET REGISTER TO BE TESTED,
012212  012705  002000                  MOV     #BIT10,R5           ;SET "EXPECTED ",
012216  010513                          MOV     R5,(R3)             ;WRITE THE REGISTER,
012220  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012222  020504                          CMP     R5,R4               ;R5=GOOD; R4=UNKNOWN,
012224  001401                          BEQ     1$                  ;ARE THEY THE SAME?
012226  104003                          HLT     3                   ;COMPARISON ERROR,
012230  040513                  1$:     BIC     R5,(R3)             ;CLEAR BIT10
012232  011304                          MOV     (R3),R4             ;READ THE REGISTER,
012234  005005                          CLR     R5                  ;SET "EXPECTED"
012236  020504                          CMP     R5,R4               ;R5=GOOD; R4=?
012240  001401                          BEQ     2$                  ;BR IF OK
012242  104003                          HLT     3                   ;COMPARISON ERROR
```

```
012244  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 37 ******************************
                                        ;*SECONDARY REGISTER SELECTOR READ/WRITE TEST.
                                        ;*SET BIT11, VERIFY BIT11 WAS SET.
                                        ;*CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
                                        ;;*************************************************************

                                ;  TEST 37
                                ;--------------
012246  012737  000037  001226  TST37:  MOV     #37,TSTNO
012254  012737  012322  001216          MOV     #TST40,NEXT
012262  013703  001372                  MOV     DVSRS,R3        ;SET REGISTER TO BE TESTED.
012266  012705  004000                  MOV     #BIT11,R5       ;SET "EXPECTED ".
012272  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
012274  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012276  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
012300  001401                          BEQ     1$              ;ARE THEY THE SAME?
012302  104003                          HLT     3               ;COMPARISON ERROR.
012304  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT11
012306  011304                          MOV     (R3),R4         ;READ THE REGISTER.
012310  005005                          CLR     R5              ;SET "EXPECTED"
012312  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
012314  001401                          BEQ     2$              ;BR IF OK
012316  104003                          HLT     3               ;COMPARISON ERROR
012320  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 40 ******************************
                                        ;*SECONDARY REGISTER ACCESS REG. READ/WRITE TEST
                                        ;*SET EACH INDIVIDUAL BIT; VERIFY EACH INDIVIDUAL BIT SET.
                                        ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
                                        ;;*************************************************************



                                ;  TEST 40
                                ;--------------
012322  012737  000040  001226  TST40:  MOV     #40,TSTNO
012330  012737  012424  001216          MOV     #TST41,NEXT
012336  013703  001376                  MOV     DVSRA,R3                ;SET CSR POINTER INTO R3
012342  012705  177777                  MOV     #-1,R5          ;SET EXPECTED TO ALL 1'S
012346  010513                          MOV     R5,(R3)         ;WRITE REGISTER WITH EXPECTED DATA
012350  011304                          MOV     (R3),R4         ;READ THE REGISTER
012352  020504                          CMP     R5,R4           ;WAS READ EQUAL TO WRITTEN??
012354  001401                          BEQ     1$              ;BR IF YES
012356  104003                          HLT     3               ;PRIMARY REGISTER READ/WRITE TEST
012360  005005                  1$:     CLR     R5              ;SET DATA TO ZERO
012362  010513                          MOV     R5,(R3)         ;WRITE REGISTER
012364  011304                          MOV     (R3),R4         ;READ REGISTER
012366  020504                          CMP     R5,R4           ;REGISTER WRITTEN OK?
012370  001401                          BEQ     2$              ;BR IF YES
012372  104003                          HLT     3               ;PRIMARY REGISTER DATA ERROR
012374  012705  000001          2$:     MOV     #1,R5           ;SET FOR "FLOATING 1"
012400  010513                  3$:     MOV     R5,(R3)         ;WRITE DATA
012402  011304                          MOV     (R3),R4         ;READ DATA
```

```
012404  020504                          CMP     R5,R4           ;DATA OK?
012406  001401                          BEQ     4$              ;BR IF YES
012410  104003                          HLT     3               ;DATA ERROR
012412  000241                  4$:     CLC                     ;ZERO CPU CARRY
012414  006105                          ROL     R5              ;UPDATE DATA
012416  001370                          BNE     3$              ;BR IF MORE TO GO
012420  005013                          CLR     (R3)            ;LEAVE REG ALL 0'S
012422  104400                          SCOPE                   ;SCOPE THIS TEST.


                                        ;*********************** TEST 41 ******************************
                                        ;*SPECIAL FUNCT. REGISTER READ/WRITE TEST
                                        ;*SET EACH INDIVIDUAL BIT; VERIFY EACH INDIVIDUAL BIT SET.
                                        ;*CLEAR EACH INDIVIDUAL BIT; VERIFY CLEAR.
                                        ;;*************************************************************



                                ;  TEST 41
                                ;--------------
012424  012737  000041  001226  TST41:  MOV     #41,TSTNO
012432  012737  012534  001216          MOV     #TST42,NEXT
012440  012777  000010  166714          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
012446  013703  001400                  MOV     DVSFR,R3                ;SET CSR POINTER INTO R3
012452  012705  177777                  MOV     #-1,R5          ;SET EXPECTED TO ALL 1'S
012456  010513                          MOV     R5,(R3)         ;WRITE REGISTER WITH EXPECTED DATA
012460  011304                          MOV     (R3),R4         ;READ THE REGISTER
012462  020504                          CMP     R5,R4           ;WAS READ EQUAL TO WRITTEN??
012464  001401                          BEQ     1$              ;BR IF YES
012466  104003                          HLT     3               ;PRIMARY REGISTER READ/WRITE TEST
012470  005005                  1$:     CLR     R5              ;SET DATA TO ZERO
012472  010513                          MOV     R5,(R3)         ;WRITE REGISTER
012474  011304                          MOV     (R3),R4         ;READ REGISTER
012476  020504                          CMP     R5,R4           ;REGISTER WRITTEN OK?
012500  001401                          BEQ     2$              ;BR IF YES
012502  104003                          HLT     3               ;PRIMARY REGISTER DATA ERROR
012504  012705  000001          2$:     MOV     #1,R5           ;SET FOR "FLOATING 1"
012510  010513                  3$:     MOV     R5,(R3)         ;WRITE DATA
012512  011304                          MOV     (R3),R4         ;READ DATA
012514  020504                          CMP     R5,R4           ;DATA OK?
012516  001401                          BEQ     4$              ;BR IF YES
012520  104003                          HLT     3               ;DATA ERROR
012522  000241                  4$:     CLC                     ;ZERO CPU CARRY
012524  006105                          ROL     R5              ;UPDATE DATA
012526  001370                          BNE     3$              ;BR IF MORE TO GO
012530  005013                          CLR     (R3)            ;LEAVE REG ALL 0'S
012532  104400                          SCOPE                   ;SCOPE THIS TEST.


                                        ;*********************** TEST 42 ******************************

                                        ;*NPR STATUS REG. TEST
                                        ;*TEST THAT NPR STATUS REG. CANNOT BE WRITTEN
                                        ;*READ THE NPR STATUS REG. AND STORE THE DATA;
                                        ;*COMPLEMENT THE DATA AND  WRITE THE NPR STATUS REG.
                                        ;*VERIFING THAT THE NPR STATUS REG. DID NOT CHANGE.
                                        ;;*************************************************************
```

```

                                       ;  TEST 42
                                       ;--------------
 012534  012737  000042  001226  TST42:  MOV     #42,TSTNO
 012542  012737  012576  001216          MOV     #TST43,NEXT
 012550  013703  001402                  MOV     DVNSR,R3
                                                                 ;GET THE REGISTER.
 012554  011305                          MOV     (R3),R5         ;READ THE REGISTER INTO R5
 012556  010504                          MOV     R5,R4           ;SAVE REG INTO R4
 012560  005104                          COM     R4              ;MAKE R4 OPPOSITE TO REGISTER
 012562  010413                          MOV     R4,(R3)         ;WRITE THE REGISTER WITH THE COMPLIMENT
 012564  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012566  020504                          CMP     R5,R4           ;IS THE REGISTER EQUAL TO ZERO.
 012570  001401                          BEQ     1$              ;BR IF OK.
 012572  104003                          HLT     3               ;REGISTER NOT ZERO.
 012574  104400                  1$:     SCOPE                   ;SCOPE THIS TEST.


                                         ;*********************** TEST 43 *****************************
                                         ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                         ;*SET BIT0, VERIFY BIT0 WAS SET.
                                         ;*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
                                         ;;**************************************************************

                                       ;  TEST 43
                                       ;--------------
 012576  012737  000043  001226  TST43:  MOV     #43,TSTNO
 012604  012737  012652  001216          MOV     #TST44,NEXT
 012612  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
 012616  012705  000001                  MOV     #BIT0,R5        ;SET "EXPECTED ".
 012622  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 012624  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012626  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 012630  001401                          BEQ     1$              ;ARE THEY THE SAME?
 012632  104003                          HLT     3               ;COMPARISON ERROR.
 012634  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT0
 012636  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012640  005005                          CLR     R5              ;SET "EXPECTED"
 012642  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 012644  001401                          BEQ     2$              ;BR IF OK
 012646  104003                          HLT     3               ;COMPARISON ERROR
 012650  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                         ;*********************** TEST 44 *****************************
                                         ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                         ;*SET BIT1, VERIFY BIT1 WAS SET.
                                         ;*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
                                         ;;**************************************************************

                                       ;  TEST 44
                                       ;--------------
 012652  012737  000044  001226  TST44:  MOV     #44,TSTNO
 012660  012737  012726  001216          MOV     #TST45,NEXT
 012666  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
 012672  012705  000002                  MOV     #BIT1,R5        ;SET "EXPECTED ".
```

```
 012676  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 012700  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012702  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 012704  001401                          BEQ     1$              ;ARE THEY THE SAME?
 012706  104003                          HLT     3               ;COMPARISON ERROR.
 012710  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT1
 012712  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012714  005005                          CLR     R5              ;SET "EXPECTED"
 012716  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 012720  001401                          BEQ     2$              ;BR IF OK
 012722  104003                          HLT     3               ;COMPARISON ERROR
 012724  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                         ;*********************** TEST 45 *****************************
                                         ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                         ;*SET BIT2, VERIFY BIT2 WAS SET.
                                         ;*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
                                         ;;**************************************************************

                                       ;  TEST 45
                                       ;--------------
 012726  012737  000045  001226  TST45:  MOV     #45,TSTNO
 012734  012737  013002  001216          MOV     #TST46,NEXT
 012742  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
 012746  012705  000004                  MOV     #BIT2,R5        ;SET "EXPECTED ".
 012752  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 012754  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012756  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 012760  001401                          BEQ     1$              ;ARE THEY THE SAME?
 012762  104003                          HLT     3               ;COMPARISON ERROR.
 012764  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT2
 012766  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 012770  005005                          CLR     R5              ;SET "EXPECTED"
 012772  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
 012774  001401                          BEQ     2$              ;BR IF OK
 012776  104003                          HLT     3               ;COMPARISON ERROR
 013000  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                         ;*********************** TEST 46 *****************************
                                         ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                         ;*SET BIT3, VERIFY BIT3 WAS SET.
                                         ;*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
                                         ;;**************************************************************

                                       ;  TEST 46
                                       ;--------------
 013002  012737  000046  001226  TST46:  MOV     #46,TSTNO
 013010  012737  013056  001216          MOV     #TST47,NEXT
 013016  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
 013022  012705  000010                  MOV     #BIT3,R5        ;SET "EXPECTED ".
 013026  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
 013030  011304                          MOV     (R3),R4         ;READ THE REGISTER.
 013032  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
 013034  001401                          BEQ     1$              ;ARE THEY THE SAME?
```

```
013036  104003                          HLT     3               ;COMPARISON ERROR.
013040  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT3
013042  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013044  005005                          CLR     R5              ;SET "EXPECTED"
013046  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
013050  001401                          BEQ     2$              ;BR IF OK
013052  104003                          HLT     3               ;COMPARISON ERROR
013054  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 47 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT4, VERIFY BIT4 WAS SET.
                                        ;*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
                                        ;;**************************************************************

                                ;  TEST 47
                                ;--------------
013056  012737  000047  001226  TST47:  MOV     #47,TSTNO
013064  012737  013132  001216          MOV     #TST50,NEXT
013072  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
013076  012705  000020                  MOV     #BIT4,R5        ;SET "EXPECTED ".
013102  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
013104  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013106  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013110  001401                          BEQ     1$              ;ARE THEY THE SAME?
013112  104003                          HLT     3               ;COMPARISON ERROR.
013114  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT4
013116  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013120  005005                          CLR     R5              ;SET "EXPECTED"
013122  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
013124  001401                          BEQ     2$              ;BR IF OK
013126  104003                          HLT     3               ;COMPARISON ERROR
013130  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 50 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT5, VERIFY BIT5 WAS SET.
                                        ;*CLEAR BIT5, VERIFY BIT5 WAS CLEARED.
                                        ;;**************************************************************

                                ;  TEST 50
                                ;--------------
013132  012737  000050  001226  TST50:  MOV     #50,TSTNO
013140  012737  013206  001216          MOV     #TST51,NEXT
013146  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
013152  012705  000040                  MOV     #BIT5,R5        ;SET "EXPECTED ".
013156  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
013160  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013162  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013164  001401                          BEQ     1$              ;ARE THEY THE SAME?
013166  104003                          HLT     3               ;COMPARISON ERROR.
013170  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT5
013172  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013174  005005                          CLR     R5              ;SET "EXPECTED"
```

```
013176  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
013200  001401                          BEQ     2$              ;BR IF OK
013202  104003                          HLT     3               ;COMPARISON ERROR
013204  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 51 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT6, VERIFY BIT6 WAS SET.
                                        ;*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
                                        ;;**************************************************************

                                ;  TEST 51
                                ;--------------
013206  012737  000051  001226  TST51:  MOV     #51,TSTNO
013214  012737  013262  001216          MOV     #TST52,NEXT
013222  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
013226  012705  000100                  MOV     #BIT6,R5        ;SET "EXPECTED ".
013232  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
013234  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013236  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013240  001401                          BEQ     1$              ;ARE THEY THE SAME?
013242  104003                          HLT     3               ;COMPARISON ERROR.
013244  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT6
013246  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013250  005005                          CLR     R5              ;SET "EXPECTED"
013252  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
013254  001401                          BEQ     2$              ;BR IF OK
013256  104003                          HLT     3               ;COMPARISON ERROR
013260  104400                  2$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 52 ******************************
                                        ;*DV11 RESERVED REGISTER READ/WRITE TEST.
                                        ;*SET BIT7, VERIFY BIT7 WAS SET.
                                        ;*CLEAR BIT7, VERIFY BIT7 WAS CLEARED.
                                        ;;**************************************************************

                                ;  TEST 52
                                ;--------------
013262  012737  000052  001226  TST52:  MOV     #52,TSTNO
013270  012737  013336  001216          MOV     #TST53,NEXT
013276  013703  001404                  MOV     RESV16,R3       ;SET REGISTER TO BE TESTED.
013302  012705  000200                  MOV     #BIT7,R5        ;SET "EXPECTED ".
013306  010513                          MOV     R5,(R3)         ;WRITE THE REGISTER.
013310  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013312  020504                          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
013314  001401                          BEQ     1$              ;ARE THEY THE SAME?
013316  104003                          HLT     3               ;COMPARISON ERROR.
013320  040513                  1$:     BIC     R5,(R3)         ;CLEAR BIT7
013322  011304                          MOV     (R3),R4         ;READ THE REGISTER.
013324  005005                          CLR     R5              ;SET "EXPECTED"
013326  020504                          CMP     R5,R4           ;R5=GOOD; R4=?
013330  001401                          BEQ     2$              ;BR IF OK
013332  104003                          HLT     3               ;COMPARISON ERROR
013334  104400                  2$:     SCOPE                   ;SCOPE THIS TEST
```

```


                                                 ;*********************** TEST 53 ******************************
                                                 ;*TEST OF THE BYTE OPERATIONS FOR THE DV11
                                                 ;*SYSTEM CONTROL REG AND THE SECONDARY REG SEL.
                                                 ;*THE TEST WILL CLEAR DVSCR AND THE WRITE (LOW BYTE)
                                                 ;*BIT3; THEN VERIFY ONLY BIT3 IS SET; THEN THE
                                                 ;*TEST WILL WRITE BIT 8(HIGH BYTE) AND VERFIY THAT
                                                 ;*BIT8 AND BIT3 ARE SET.  THE EXACT PROCEEDURE
                                                 ;*WILL BE USED ON THE DVSRS REGISTER.
                                                 ;;***************************************************************

                                         ;  TEST 53
                                         ;--------------
013336  012737  000053  001226   TST53:  MOV     #53,TSTNO
013344  012737  013472  001216           MOV     #TST54,NEXT
013352  013703  001362                   MOV     DVSCR,R3             ;SET DVSCR POINTER
013356  005013                           CLR     (R3)                 ;MAKE SURE IT =0
013360  012705  000010                   MOV     #BIT3,R5             ;LOAD EXPECTED RESULTS
013364  110513                           MOVB    R5,(R3)              ;"WRITE" BYTE (LOW) BIT3
013366  011304                           MOV     (R3),R4              ;READ (WORD) RESULT
013370  020504                           CMP     R5,R4                ;MAKE SURE ONLY BIT3 IS SET
013372  001401                           BEQ     1$                   ;BR IF OK
013374  104003                           HLT     3                    ;DVSCR WRONG
013376  012705  000410           1$:     MOV     #BIT8+BIT3,R5        ;SET EXPECTED DATA
013402  112763  000001  000001           MOVB    #BIT0,1(R3)          ;"WRITE" BYTE (HIGH) BIT0 [BIT8 OF WORD]
013410  011304                           MOV     (R3),R4              ;READ (WORD) RESULT
013412  020504                           CMP     R5,R4                ;OK?
013414  001401                           BEQ     2$                   ;
013416  104003                           HLT     3                    ;DVSCR WRONG
013420  005013                   2$:     CLR     (R3)                 ;LEAVE REGISTERED CLEARED
013422  013703  001372                   MOV     DVSRS,R3             ;GET NEXT REGISTER FOR BYTE TEST.
013426  005013                           CLR     (R3)                 ;MAKE SURE WORD IS =0.
013430  012705  000010                   MOV     #BIT3,R5             ;SET EXPECTED
013434  110513                           MOVB    R5,(R3)              ;WRITE BYTE (LOW) BIT3
013436  011304                           MOV     (R3),R4              ;READ RESULT
013440  020504                           CMP     R5,R4                ;OK?
013442  001401                           BEQ     4$                   ;
013444  104003                           HLT     3                    ;DVSRS WRONG
013446  012705  000410           4$:     MOV     #BIT8+BIT3,R5        ;SET EXPECTED
013452  112763  000001  000001           MOVB    #BIT0,1(R3)          ;WRITE BYTE (HIGH) BIT0 [BIT8 OF WORD]
013460  011304                           MOV     (R3),R4              ;READ RESULT
013462  020504                           CMP     R5,R4                ;OK
013464  001401                           BEQ     5$                   ;
013466  104003                           HLT     3                    ;DVSRS FAILED
013470  104400                   5$:     SCOPE                        ;SCOPE TEST

```

```
                                                 ;*********************** TEST 54 ******************************
                                                 ;*SECONDARY REGISTER READ/WRITE TESTS
                                                 ;*READ/WRITE TEST. READ AND WRITE DIFFERENT DATA
                                                 ;*PATTERENS INTO THE SECONDARY REGISTERS VERIFING
                                                 ;*THAT WHAT WAS READ MATCHES WHAT WAS WRITTEN.
                                                 ;;***************************************************************

                                         ;  TEST 54
                                         ;--------------
013472  012737  000054  001226   TST54:  MOV     #54,TSTNO
013500  012737  013674  001216           MOV     #TST55,NEXT
013506  012737  013524  001220           MOV     #1$,LOCK
013514  005000                           CLR     R0                   ;SEL LINE NUMBER TO ZERO
013516  005001                           CLR     R1                   ;SET SEC. REG TO ZERO ALSO
013520  013702  001376                   MOV     DVSRA,R2             ;SET ADDRESS POINTER INTO R2
013524  110077  165642           1$:     MOVB    R0,@DVSRS            ;LOAD LINE NUMBER
013530  110177  165640                   MOVB    R1,@DVSRSH           ;LOAD SEC. REG.
013534  005004                           CLR     R4                   ;ZERO DATA PATTERN
013536  010412                   2$:     MOV     R4,(R2)              ;LOAD DATA INTO DV11 REGISTER
013540  011203                           MOV     (R2),R3              ;READ BACK THE DATA
013542  020403                           CMP     R4,R3                ;WAS WHAT WAS WRITTEN READ BACK??
013544  001401                           BEQ     64$                  ;BR IF DATA OK
013546  104002                           HLT     2                    ;SECONDARY REGISTER READ WRITE ERROR.
013550  104401                   64$:    SCOP1                        ;LOCK ON DATA,LINE,AND SEC REG? (SW09=1)
013552  005104                           COM     R4                   ;MAKE DATA ALL 1'S
013554  001370                           BNE     2$                   ;RETURN AND WRITE IT INTO THE REGISTER
013556  012704  000001                   MOV     #1,R4                ;GET READY FOR THE "FLOATING 1"
013562  012737  013570  001220           MOV     #3$,LOCK             ;SET IF SW09=1
013570  010412                   3$:     MOV     R4,(R2)              ;LOAD DATA
013572  011203                           MOV     (R2),R3              ;READ DATA
013574  020403                           CMP     R4,R3                ;DATA OK??
013576  001401                           BEQ     65$                  ;BR IF OK!
013600  104002                           HLT     2                    ;SECONDARY REGISTER READ/WRITE ERROR
013602  104401                   65$:    SCOP1                        ;SW09=1?
013604  000241                           CLC                          ;ZERO CPU CARRY
013606  006104                           ROL     R4                   ;FLOAT THE 1
013610  001367                           BNE     3$                   ;IF NOT DONE WRITE NEW PATTERN
013612  012704  177776                   MOV     #^C<1>,R4            ;GET READY FOR THE "FLOATING 0"
013616  012737  013624  001220           MOV     #4$,LOCK             ;SET IF SW09=1
013624  010412                   4$:     MOV     R4,(R2)              ;WRITE DATA
013626  011203                           MOV     (R2),R3              ;READ DATA
013630  020403                           CMP     R4,R3                ;DATA OK??
013632  001401                           BEQ     66$                  ;BR IF OK!
013634  104002                           HLT     2                    ;SECONDARY REGISTER DATA COMPARE ERROR
013636  104401                   66$:    SCOP1                        ;SW09=1??
013640  000261                           SEC                          ;SET CPU CARRY
013642  006104                           ROL     R4                   ;CHANGE DATA PATTERN
013644  103767                           BCS     4$                   ;IF NOT DONE ,GO BACK WITH NEW PATTERN
013646  005012                           CLR     (R2)                 ;ZERO THE REGISTER (ALL DONE WITH THIS ONE.
013650  005201                           INC     R1                   ;UPDATE THE SEC REG POINTER
013652  022701  000020                   CMP     #16.,R1              ;ALL SEC REGISTERS DONE?
013656  001322                           BNE     1$                   ; BR IF NO.
013660  005001                           CLR     R1                   ;ZERO SEC REG POINTER
013662  005200                           INC     R0                   ;UPDATE LINE NUMBER POINTER
013664  022700  000020                   CMP     #16.,R0              ;ALL LINES DONE?
013670  001315                           BNE     1$                   ;BR IF NO
```

```
013672  104400                          SCOPE                   ;SCOPE THIS TEST


                                        ;********************** TEST 55 ****************************
                                        ;*INDIVIDUAL LINE DUEL ADDRESS TESTS
                                        ;*THIS TEST VERIFIES THAT WRITING ONE SECONDARY
                                        ;*REGISTER FOR A SPECIFIC LINE DOES NOT ALTER
                                        ;*ANY OTHER SECONDARY REGISTER FOR THAT LINE.
                                        ;;*************************************************************

                                ;  TEST 55
                                ;--------------
013674  012737  000055  001226  TST55:  MOV     #55,TSTNO
013702  012737  014034  001216          MOV     #TST56,NEXT
013710  012737  013770  001220          MOV     #2$,LOCK
013716  005000                          CLR     R0              ;SELECT THE LINE NUMBER.
013720  005001                  65$:    CLR     R1              ;SET FOR SEC. REG. POINTER.
013722  005004                          CLR     R4              ;SET DATA
013724  013702  001376                  MOV     DVSRA,R2        ;SET ACCESS REGISTER.
013730  110077  165436                  MOVB    R0,@DVSRS       ;SELECT THE LINE NUMBER
013734  110177  165434          1$:     MOVB    R1,@DVSRSH      ;SELECT THE SEC. REG.
013740  010412                          MOV     R4,(R2)         ;WRITE SEC. REG.
013742  062704  010421                  ADD     #^B<0001000100010001>,R4
                                                                ;UPDATE DATA
013746  005201                          INC     R1              ;UPDATE SECONDARY REG. POINTER
013750  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. DONE?
013754  001367                          BNE     1$              ;BR IF NO
013756  005001                          CLR     R1              ;RESET SEC. REG. POINTER TO ZERO.
013760  005004                          CLR     R4              ;ZERO DATA COMPARE
013762  012737  013770  001220          MOV     #2$,LOCK        ;SET FOR LOCK.
013770  110177  165400          2$:     MOVB    R1,@DVSRSH      ;GET SEC. REG.
013774  011203                          MOV     (R2),R3         ;READ SEC. REG.
013776  020403                          CMP     R4,R3           ;R4=GOOD; R3=UNKNOWN
014000  001401                          BEQ     3$              ;BR IF ALL OK
014002  104002                          HLT     2               ;SECONDARY REGISTER ADDRESSING ERROR
014004  104401                  3$:     SCOP1                   ;LOCK ON REG. (SW09=1)
014006  062704  010421                  ADD     #^B<0001000100010001>,R4
014012  005201                          INC     R1              ;UPDATE SEC REG POINTER
014014  022701  000020                  CMP     #16.,R1         ;ALL 16 LINES TESTED YET?
014020  001363                          BNE     2$              ;BR IF NO
014022  005200                          INC     R0              ;UPDATE LINE NO POINTER
014024  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE??
014030  001333                          BNE     65$             ;BR IF NO
014032  104400                          SCOPE                   ;SCOPE THE TEST


                                        ;********************** TEST 56 ****************************
                                        ;*VERIFY NO LINE INTERACTION.
                                        ;*THIS TEST VERIFIES THAT WRITING THE SECONDARY
                                        ;*REGISTERS FOR ONE LINE DOES NOT INTERFEAR WITH
                                        ;*THE SECONDARY REGISTERS OF ANOTHER LINE.
                                        ;;*************************************************************

                                ;  TEST 56
                                ;--------------
014034  012737  000056  001226  TST56:  MOV     #56,TSTNO
```

```
014042  012737  000002  001222          MOV     #2,ICOUNT
014050  012737  014432  001216          MOV     #TST57,NEXT
014056  012737  014150  001220          MOV     #4$,LOCK
014064  005000                          CLR     R0              ;SELECT THE LINE NUMBER
014066  005001                          CLR     R1              ;R1 = SECONDARY REGISTER SELECT.
014070  005004                          CLR     R4              ;SET DATA TO BE WRITTEN.
014072  013702  001376                  MOV     DVSRA,R2        ;SET "SECONDARY REGISTER ACCESS" POINTER.
014076  110077  165270          1$:     MOVB    R0,@DVSRS       ;GET THE LINE NUMBER SELECTED.
014102  110177  165266          2$:     MOVB    R1,@DVSRSH      ;GET THE SECONDARY REGISTER.
014106  010412                          MOV     R4,(R2)         ;WRITE THE SECONDARY REG.
014110  005201                          INC     R1              ;UPDATE  SECONDARY REG. POINTER.
014112  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. DONE?
014116  001371                          BNE     2$              ;BR IF NO.
014120  005001                          CLR     R1              ;ZERO SEC.REG.POINTER
014122  062704  010421                  ADD     #^B<0001000100010001>,R4
                                                                ;UPDATE DATA POINTER.
014126  005200                          INC     R0              ;UPDATE LINE POINTER.
014130  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
014134  001360                          BNE     1$              ;BR IF NO.

014136  005000                          CLR     R0              ;SELECT LINE NUMBER
014140  005001                          CLR     R1              ;START SEC. REG. AT 0
014142  005004                          CLR     R4              ;ZERO DATA COMPARE.
014144  110077  165222          3$:     MOVB    R0,@DVSRS       ;SELECT LINE NUMBER.
014150  110177  165220          4$:     MOVB    R1,@DVSRSH      ;SELECT SEC.REG.
014154  011203                          MOV     (R2),R3         ;READ THE SEC. REG.
014156  020403                          CMP     R4,R3           ;WAS DATA CORRECT?
014160  001401                          BEQ     5$              ;BR IF OK
014162  104002                          HLT     2               ;SECONDARY REG. ADDRESSING ERROR.
014164  104401                  5$:     SCOP1                   ;LOCK ON REGISTER? (SW09=1)
014166  005201                          INC     R1              ;UPDATE SEC. REG. POINTER
014170  022701  000020                  CMP     #16.,R1         ;ALL SEC. REG. FOR THIS LINE DONE?
014174  001365                          BNE     4$              ;BR IF NO.
014176  062704  010421                  ADD     #^B<0001000100010001>,R4
                                                                ;UPDATE DATA
014202  005001                          CLR     R1              ;SET FOR SEC. REG.
014204  005200                          INC     R0              ;UPDATE LINE NUMBER POINTER.
014206  022700  000020                  CMP     #16.,R0         ;ALL LINES DONE?
014212  001354                          BNE     3$              ;BR IF NO

                                        ;*PART 2
                                        ;*FILL ALL RAMS WITH ALL 1'S AND THEN
                                        ;*CLEAR JUST ONE BIT AT A TIME VERIFYING
                                        ;*THAT ONLY THAT ONE BIT IS CLEAR AND THAT
                                        ;*ALL OTHER RAMS STILL CONTAIN ALL 1'S.
                                        ;*THERE SHOULD BE ONLY ONE BIT CLEARED AT
                                        ;*ONE TIME.

014214  005077  165164                  CLR     @RESV16         ;CLEAR "LOC FOUND FLAG".
014220  005037  001220                  CLR     LOCK            ;
014224  013702  001376          MAR17:  MOV     DVSRA,R2        ;LOAD POINTER TO DVSRA
014230  005077  165136                  CLR     @DVSRS          ;SET LINE AND SEC REG POINTER TO ZERO
014234  012712  177777          1$:     MOV     #^B<1111111111111111>,(R2)
                                                                ;PREPARE TO LOAD ALL 1'S INTO ALL RAM LOC.
014240  062777  170361  165124          ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
                                                                ;UPDATE LINE AND SEC REG POINTERS.
```

```
014246  001372                          BNE     1$                  ;BR IF NOT ALL DONE FILLING 1'S.
014250  005000                          CLR     R0                  ;ZERO LINE # POINTER
014252  005001                          CLR     R1                  ;ZERO  SEC REG # POINTER
014254  012704  177776          2$:     MOV     #^C<BIT0>,R4        ;SET INITAL DATA
014260  110077  165106          3$:     MOVB    R0,@DVSRS           ;LOAD LINE #
014264  110177  165104                  MOVB    R1,@DVSRSH          ;LOAD SEC REG #
014270  010412                          MOV     R4,(R2)             ;LOAD DATA
014272  005077  165074                  CLR     @DVSRS              ;ZERO POINTERS
014276  022712  177777          100$:   CMP     #^B<1111111111111111>,(R2)
                                                                    ;VERIFY ONLY ONE LOC. HAS ONE BIT CLEARED
014302  001417                          BEQ     6$                  ;BR IF LOC. OK
014304  012777  177777  165072          MOV     #-1,@RESV16         ;SET "LOC FOUND FLAG".
014312  011203                          MOV     (R2),R3             ;SAVE DATA
014314  120077  165052                  CMPB    R0,@DVSRS           ;IS THIS THE RIGHT LINE?
014320  001401                          BEQ     4$                  ;BR IF YES
014322  104002                          HLT     2                   ;WRONG LINE HAS CLEARED BIT!
014324  120177  165044          4$:     CMPB    R1,@DVSRSH          ;IS THIS THE RIGHT SEC REG?
014330  001401                          BEQ     5$                  ;BR IF YES
014332  104002                          HLT     2                   ;WRONG SEC REG HAS CLEARED BIT.
014334  020403                  5$:     CMP     R4,R3               ;IS THE ACTUAL DATA OK?
014336  001401                          BEQ     6$                  ;BR IF YES
014340  104002                          HLT     2                   ;ACTUAL DATA WAS WRONG.
014342  062777  170361  165022  6$:     ADD     #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,@DVSRS
014350  001352                          BNE     100$                ;BR IF NOT DONE.
014352  005777  165026                  TST     @RESV16             ;HAS A LOC BEEN FOUND?
014356  001001                          BNE     7$                  ;BR IF YES
014360  104000                          HLT     0                   ;NO LOC WAS FOUND WITH A ZERO BIT.
014362  005077  165016          7$:     CLR     @RESV16             ;CLEAR "LOC FOUND FLAG".
014366  000261                          SEC                         ;SHIFT IN A  1
014370  006104                          ROL     R4                  ;CHANGE DATA PATERN
014372  103732                          BCS     3$                  ;DO IT ALL OVER AGAIN
014374  110077  164772                  MOVB    R0,@DVSRS           ;LOAD LINE NO.
014400  110177  164770                  MOVB    R1,@DVSRSH          ;LOAD SEC REG.
014404  010412                          MOV     R4,(R2)             ;PUT RAM LOC BACK TO ALL 1'S.
014406  005201                          INC     R1                  ;UPDATE SEC REG #
014410  022701  000020                  CMP     #16.,R1             ;ALL SEC REG DONE?
014414  001317                          BNE     2$                  ;BR IF NO
014416  005001                          CLR     R1                  ;ZERO SEC REG POINTER
014420  005200                          INC     R0                  ;UPDATE  LINE POINTER
014422  022700  000020                  CMP     #16.,R0             ;ALL LINES DONE?
014426  001312                          BNE     2$                  ;BR IF NO
014430  104400                          SCOPE                       ;SCOPE  THIS TEST.



                                        ;************************ TEST 57 ******************************
                                        ;*MEMORY EXTENSION READ/WRITE TEST
                                        ;*VERIFY BITS 4 AND 5 OF EACH LINE
                                        ;*SECONDARY REGISTERS EXERCISED ARE:
                                        ;* 00   TX BUS ADDRESS (PRIMARY)
                                        ;* 02   TX BUS ADDRESS (SECONDARY)
                                        ;* 04   RX BUS ADDRESS
                                        ;* 10   TX TABLE BASE ADDRESS
                                        ;* 11   RX TABLE BASE ADDRESS
                                        ;*NOTE THAT ALL LINES (00-16) ARE EXERCISED.
                                        ;;**************************************************************
```

```


                                ;  TEST 57
                                ;--------------
014432  012737  000057  001226  TST57:  MOV     #57,TSTNO
014440  012737  014626  001216          MOV     #TST60,NEXT
014446  012737  014506  001220          MOV     #2$,LOCK
014454  005000                          CLR     R0                  ;R0=LINE NUMBER (START AT 0)
014456  013702  001370                  MOV     DVLCR,R2            ;SET R2 =BASE ADDRESS(DVLCR)
014462  012704  000060          1$:     MOV     #BIT5+BIT4,R4       ;R4=GOOD DATA  (BOTH EA BITS SET AT START)
014466  012705  000005                  MOV     #5,R5               ;R5 IS COUNTER OF SEC REGISTERS
014472  012737  000004  001246          MOV     #4,TEMP1            ;TEMP1 IS COUNTER OF COMB. OF EA BITS.
                                                                    ;EX: 11,10,01,00
014500  012737  014620  001256          MOV     #MEMEXT,TEMP5       ;TEMP5=SEC. REGISTER POINTER.
014506  010477  164650          2$:     MOV     R4,@DVSCR           ;LOAD DVSCR WITH EA BITS.
014512  110077  164654                  MOVB    R0,@DVSRS           ;SEL THE LINE NUMBER
014516  117701  164534                  MOVB    @TEMP5,R1           ;GET SEC REG.
014522  110177  164646                  MOVB    R1,@DVSRSH          ;SEL THE SEC. REGISTER
014526  005077  164644                  CLR     @DVSRA              ;HIT THE SEC.REG. ACCESS REGISTER.
014532  017703  164632                  MOV     @DVLCR,R3           ;SAVE THE DVLINE PARM. REG.
014536  042703  177717                  BIC     #^C<BIT5+BIT4>,R3
                                                                    ;CLEAR ALL BUT BITS 5 AND 4.
014542  020403                          CMP     R4,R3               ;ARE THE EA BITS GOOD
014544  001401                          BEQ     3$
014546  104004                          HLT     4
014550  104401                  3$:     SCOP1                       ;SW09=1?
014552  005237  001256                  INC     TEMP5               ;POP POINTER
014556  005305                          DEC     R5                  ;ALL SEC REG DONE?
014560  001352                          BNE     2$                  ;BR IF NO.
014562  012737  014620  001256          MOV     #MEMEXT,TEMP5       ;RESET POINTER
014570  012705  000005                  MOV     #5,R5               ;RESET COUNTER
014574  162704  000020                  SUB     #BIT4,R4            ;ADJUST FOR NEXT EA BIT PATTERN
014600  005337  001246                  DEC     TEMP1               ;ALL PATERNS DONE?
014604  001340                          BNE     2$
014606  005200                          INC     R0                  ;UPDATE TO NEXT LINE
014610  022700  000020                  CMP     #16.,R0             ;ALL LINES DONE
014614  001322                          BNE     1$                  ;BR IF NO.
014616  104400                          SCOPE
                                ;TABLE OF SECONDARY REGISTERS EXERCISERD....
014620     000                  MEMEXT: .BYTE   00
014621     002                          .BYTE   02
014622     004                          .BYTE   04
014623     010                          .BYTE   10
014624     011                          .BYTE   11
        014626                  .EVEN


                                        ;************************ TEST 60 ******************************
                                        ;*INITIALIZATION TESTS
                                        ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
                                        ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
                                        ;*BY A BUS RESET
                                        ;;**************************************************************

                                ;  TEST 60
                                ;--------------
```

```
014626  012737  000060  001226  TST60:  MOV     #60,TSTNO
014634  012737  015042  001216          MOV     #TST61,NEXT
014642  012737  000340  177776          MOV     #340,PS          ;LOCK OUT INTERUPTS.
014650  013703  001362                  MOV     DVSCR,R3         ;SET REGISTER POINTER FOR LOADING
014654  005077  164512                  CLR     @DVSRS           ;CLEAR LINE POINTER
014660  005077  164512                  CLR     @DVSRA           ;CLEAR ACCESS REG.
014664  012723  173777                  MOV     #^C<BIT11>,(R3)+
014670  012702  000007                  MOV     #7,R2            ;SET ALL BITS BUT MSTCLR
014674  012723  177777          1$:     MOV     #-1,(R3)+        ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
014700  005302                          DEC     R2               ;ALLREGISTERS LOADED?
014702  001374                          BNE     1$               ;BR IF NO
014704  000005                          RESET                    ;ISSUES A BUS INIT (RESET INSTR)
014706  005200                          INC     R0               ;FLASH THE CPU LIGHTS!!!
014710  013703  001362                  MOV     DVSCR,R3         ;SET REGISTER POINTER
014714  005005                          CLR     R5               ;SET "EXPECTED" FOR DVSCR
014716  011304                          MOV     (R3),R4          ;READ THE DVSCR REG.
014720  020504                          CMP     R5,R4            ;IS BIT8 ALONE SET?
014722  001401                          BEQ     2$               ;BR IF YES
014724  104003                          HLT     3                ;DVSCR HAS WRONG DATA>
014726  005723                  2$:     TST     (R3)+            ;POP POINTER TO DVRIC
014730  005005                          CLR     R5               ;SET EXPECTED TO ZERO
014732  011304                          MOV     (R3),R4          ;DVRIC (EXPECT ALL 0'S)
014734  001401                          BEQ     3$               ;BR IF OK
014736  104003                          HLT     3                ;DVRIC NO ALL 0'S
014740  005723                  3$:     TST     (R3)+            ;POP POINTER TO DVLCR REG
014742  011304                          MOV     (R3),R4          ;DVLCR (READ DVLCR INTO R4)
014744  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4
014750  020504                          CMP     R5,R4            ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
014752  001401                          BEQ     4$               ;DVLCR OK?
014754  104003                          HLT     3                ;DVLCR INCORECT (DISREGUARD BIT5,4,1,0)
014756  005723                  4$:     TST     (R3)+            ;POP POINTER TO DVSRS REG
014760  011304                          MOV     (R3),R4          ;DVSRS (EXPECT ALL 0'S)
014762  001401                          BEQ     5$               ;BR IF OK
014764  104003                          HLT     3                ;DVSRS REG NOT ALL ZEROS
014766  005723                  5$:     TST     (R3)+            ;POP POINTER TO DVSRA REG
014770  011304                          MOV     (R3),R4          ;DVSRA (EXPECT ALL 0'S)
014772  001401                          BEQ     6$               ;BR IF GOOD
014774  104003                          HLT     3                ;DVSRA NOT ALL 0'S
014776  005723                  6$:     TST     (R3)+            ;POP POINTER TO DVSFR
015000  011304                          MOV     (R3),R4          ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
015002  012705  177777                  MOV     #177777,R5       ;SET EXPECTED
015006  020504                          CMP     R5,R4            ;EXPECETD =FOUND?
015010  001401                          BEQ     7$               ;BR IF YES
015012  104003                          HLT     3                ;DVSFR NOT ALL 1'S
015014  005723                  7$:     TST     (R3)+            ;POP POINTER TO DVNSR REG
015016  005713                          TST     (R3)             ;DVNSR S/B PLUS (15=0)
015020  100001                          BPL     64$
015022  104000                          HLT     0
015024  005723                  64$:    TST     (R3)+            ;POP POINTER TO RESV16 REG
015026  011304                          MOV     (R3),R4          ;RESV16 (EXPECT ALL 0'S)
015030  005005                          CLR     R5               ;SET EXPECTED TO 0'S
015032  020504                          CMP     R5,R4            ;WELL DOES IT =1'S?
015034  001401                          BEQ     8$               ;BR IF OK
015036  104003                          HLT     3                ;RESV16 NOT ALL 0'S
015040  104400                  8$:     SCOPE                    ;SCOPE THIS TEST;
```

```
                                        ;*********************** TEST 61 ******************************
                                        ;*INITIALIZATION TESTS
                                        ;*SET ALL POSSIBLE BITS IN ALL THE PRIMARY REGISTERS
                                        ;*AND VERIFY THAT ALL THE BITS ARE CLEARED
                                        ;*BY A MASTER CLEAR
                                        ;;*************************************************************

                                ;  TEST 61
                                ;---------------
015042  012737  000061  001226  TST61:  MOV     #61,TSTNO
015050  012737  015260  001216          MOV     #TST62,NEXT
015056  012737  000340  177776          MOV     #340,PS          ;LOCK OUT INTERUPTS.
015064  013703  001362                  MOV     DVSCR,R3         ;SET REGISTER POINTER FOR LOADING
015070  005077  164276                  CLR     @DVSRS           ;CLEAR LINE POINTER
015074  005077  164276                  CLR     @DVSRA           ;CLEAR ACCESS REG.
015100  012723  173777                  MOV     #^C<BIT11>,(R3)+
015104  012702  000007                  MOV     #7,R2            ;SET ALL BITS BUT MSTCLR
015110  012723  177777          1$:     MOV     #-1,(R3)+        ;LOAD ALL OTHER REGISTERS WITH ALL 1'S
015114  005302                          DEC     R2               ;ALLREGISTERS LOADED?
015116  001374                          BNE     1$               ;BR IF NO
015120  052777  004000  164234          BIS     #MRESET,@DVSCR   ;ISSUE A "MASTER CLEAR"
015126  013703  001362                  MOV     DVSCR,R3         ;SET REGISTER POINTER
015132  005005                          CLR     R5               ;SET "EXPECTED" FOR DVSCR
015134  011304                          MOV     (R3),R4          ;READ THE DVSCR REG.
015136  020504                          CMP     R5,R4            ;IS BIT8 ALONE SET?
015140  001401                          BEQ     2$               ;BR IF YES
015142  104003                          HLT     3                ;DVSCR HAS WRONG DATA>
015144  005723                  2$:     TST     (R3)+            ;POP POINTER TO DVRIC
015146  005005                          CLR     R5               ;SET EXPECTED TO ZERO
015150  011304                          MOV     (R3),R4          ;DVRIC (EXPECT ALL 0'S)
015152  001401                          BEQ     3$               ;BR IF OK
015154  104003                          HLT     3                ;DVRIC NO ALL 0'S
015156  005723                  3$:     TST     (R3)+            ;POP POINTER TO DVLCR REG
015160  011304                          MOV     (R3),R4          ;DVLCR (READ DVLCR INTO R4)
015162  042704  000063                  BIC     #BIT5+BIT4+BIT1+BIT0,R4
015166  020504                          CMP     R5,R4            ;DISREGUARD BR TEST POINTS AND MEM EXT BITS.
015170  001401                          BEQ     4$               ;DVLCR OK?
015172  104003                          HLT     3                ;DVLCR INCORECT (DISREGUARD BIT5,4,1,0)
015174  005723                  4$:     TST     (R3)+            ;POP POINTER TO DVSRS REG
015176  011304                          MOV     (R3),R4          ;DVSRS (EXPECT ALL 0'S)
015200  001401                          BEQ     5$               ;BR IF OK
015202  104003                          HLT     3                ;DVSRS REG NOT ALL ZEROS
015204  005723                  5$:     TST     (R3)+            ;POP POINTER TO DVSRA REG
015206  011304                          MOV     (R3),R4          ;DVSRA (EXPECT ALL 0'S)
015210  001401                          BEQ     6$               ;BR IF GOOD
015212  104003                          HLT     3                ;DVSRA NOT ALL 0'S
015214  005723                  6$:     TST     (R3)+            ;POP POINTER TO DVSFR
015216  011304                          MOV     (R3),R4          ;DVSFR (EXPECT ALL 1'S (THATS RIGHT))
015220  012705  177777                  MOV     #177777,R5       ;SET EXPECTED
015224  020504                          CMP     R5,R4            ;EXPECETD =FOUND?
015226  001401                          BEQ     7$               ;BR IF YES
015230  104003                          HLT     3                ;DVSFR NOT ALL 1'S
015232  005723                  7$:     TST     (R3)+            ;POP POINTER TO DVNSR REG
015234  005713                          TST     (R3)             ;DVNSR S/B PLUS (15=0)
015236  100001                          BPL     64$
```

```
015240  104000                                  HLT     0
015242  005723                          64$:    TST     (R3)+           ;POP POINTER TO RESV16 REG
015244  011304                                  MOV     (R3),R4         ;RESV16 (EXPECT ALL 0'S)
015246  005005                                  CLR     R5              ;SET EXPECTED TO 0'S
015250  020504                                  CMP     R5,R4           ;WELL DOES IT =1'S?
015252  001401                                  BEQ     8$              ;BR IF OK
015254  104003                                  HLT     3               ;RESV16 NOT ALL 0'S
015256  104400                          8$:     SCOPE                   ;SCOPE THIS TEST;


                                                ;*********************** TEST 62 *****************************
                                                ;*ATTACK OF THE SPECIAL FUNCTIONS REGISTER.
                                                ;*BEGIN CHECK OF THE DVSFR.
                                                ;*SUMARY OF PROC. INSTRUCTIONS
                                                ;*
                                                ;*BIT14 BIT13 BIT12  INSTRUCTION
                                                ;*  0     0     0    BRANCH "A"
                                                ;*  0     0     1    ALU OPERATION
                                                ;*  0     1     0    RAM OPERATION
                                                ;*  0     1     1    DATA TRANSFER
                                                ;*  1     0     0    NPR OPERATION
                                                ;*  1     0     1    SET/CLEAR OPERATION
                                                ;*  1     1     0    BCC CALCULATION
                                                ;*  1     1     1    BRANCH "B"
                                                ;*
                                                ;:*************************************************************

                                                ;*********************** TEST 62 *****************************
                                                ;*VERIFY THAT "ROM STEP"
                                                ;*IS SELF-CLEARING AND THAT
                                                ;*THE DATA IN THE DVSFR
                                                ;*IS CHANGED WHEN THE ROM IS STEPPED.
                                                ;:*************************************************************

                                        ;  TEST 62
                                        ;--------------
015260  012737  000062  001226  TST62:  MOV     #62,TSTNO
015266  012737  015362  001216          MOV     #TST63,NEXT
015274  104412                          MSTCLR                  ;CLEAR ALL THE DV11
015276  012777  000010  164056          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
015304  012705  050000                  MOV     #S.C,R5 ;PUT INSTR INTO DVSFR
015310  010577  164064                  MOV     R5,@DVSFR       ;
015314  027705  164060                  CMP     @DVSFR,R5       ;WAS THE DVSFR REALLY LOADED?
015320  001401                          BEQ     1$              ;BR IF YES
015322  104000                          HLT                     ;BAD DVSFR
015324  042777  000010  164030  1$:     BIC     #BIT3,@DVSCR    ;CLEAR SOURCE SEL.
015332  104415                          ROMCLK                  ;ISSUE A ROM CLOCK
015334  000240                          NOP                     ;WAIST AN INTRUSTION TIME
015336  032777  000002  164016          BIT     #BIT1,@DVSCR    ;DID CLK BIT CLEAR BY IT SELF?
015344  001401                          BEQ     2$              ;BR IF CLK GONE
015346  104000                          HLT                     ;BIT 1 OF DVSCR (ROM CLK) NOT ZERO
015350  020577  164024          2$:     CMP     R5,@DVSFR       ;HAS DATA IN DVSFR CHANGED?
015354  001001                          BNE     3$              ;BR IF YES
015356  104000                          HLT                     ;DATA NOT CHANGED (DID CLK REALLY CLK??)
015360  104400                  3$:     SCOPE                   ;SCOPE THIS TEST.
```

```
                                        ;*********************** TEST 63 *****************************
                                        ;*BASIC TEST OF DVSFR
                                        ;*TEST THAT "BRANCH A" INSTRUCTION.
                                        ;*POINTS TESTED:
                                        ;*
                                        ;*BIT11  BIT10  BIT09  BIT08  BR "A"  BR "B"  FUNCTION
                                        ;*  0      0      0      1      L       H    PLUS 3 VOLTS
                                        ;*  0      1      0      1      L,H     H,H DVSCR08 (=0,=1)
                                        ;*  0      1      1      1      H       H    NPR SILO NOT AVAIL.
                                        ;*  1      1      1      1      H       H    SILO FULL
                                        ;*  0      1      1      0      H       H      NXM

                                        ;:*************************************************************

                                        ;  TEST 63
                                        ;--------------
015362  012737  000063  001226  TST63:  MOV     #63,TSTNO
015370  012737  015626  001216          MOV     #TST64,NEXT
015376  104412                          MSTCLR                  ;CLEAR DV11
015400  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER TO R0
015404  013703  001370                  MOV     DVLCR,R3        ;SET DVLCR POINTER TO R3
015410  012777  000410  163744          MOV     #BIT8+BIT3,@DVSCR
015416  012710  000400          1$:     MOV     #BIT8,(R0)      ;BR=A TEST +3
015422  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015424  012705  000002                  MOV     #BIT1,R5        ;SET EXPECTED RESULTS
015430  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
015432  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                ;CLEAR UNWANTED BITS.
015436  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
015440  001401                          BEQ     2$              ;BR IF OK
015442  104006                          HLT     6               ;BR POINT WRONG
015444  012710  002400          2$:     MOV     #BIT10+BIT8,(R0)
015450  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015452  012705  000002                  MOV     #BIT1,R5        ;SET EXPECTED RESULTS
015456  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
015460  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                ;CLEAR UNWANTED BITS.
015464  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
015466  001401                          BEQ     3$              ;BR IF OK
015470  104006                          HLT     6               ;BR POINT WRONG
015472  042777  000400  163662  3$:     BIC     #BIT8,@DVSCR    ;RESET BIT 8 TO =0
015500  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015502  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015506  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
015510  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                ;CLEAR UNWANTED BITS.
015514  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
015516  001401                          BEQ     4$              ;BR IF OK
015520  104006                          HLT     6               ;BR POINT WRONG
015522  012710  003400          4$:     MOV     #BIT10+BIT9+BIT8,(R0)
015526  011002                          MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015530  012705  000003                  MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015534  011304                          MOV     (R3),R4         ;READ DVLCR INTO R4
015536  042704  177774                  BIC     #^C<BIT1+BIT0>,R4
                                                                ;CLEAR UNWANTED BITS.
015542  020504                          CMP     R5,R4           ;EXPECTED = FOUND??
```

```
015544  001401                           BEQ     5$              ;BR IF OK
015546  104006                           HLT     6               ;BR POINT WRONG
015550  012710  007400           5$:     MOV     #BIT11+BIT10+BIT9+BIT8,(R0)
015554  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015556  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015562  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
015564  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
015570  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
015572  001401                           BEQ     6$              ;BR IF OK
015574  104006                           HLT     6               ;BR POINT WRONG
015576  012710  003000           6$:     MOV     #BIT10+BIT9,(R0)        ;NXM
015602  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015604  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015610  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
015612  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
015616  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
015620  001401                           BEQ     7$              ;BR IF OK
015622  104006                           HLT     6               ;BR POINT WRONG
015624  104400                   7$:     SCOPE                   ;SCOPE TEST

                                         ;*********************** TEST 64 *****************************
                                         ;*TEST OF BRANCH B"
                                         ;*TEST THAT POINT 16 (GROUND)
                                         ;*MAKES LCR BIT1=1 AND BIT0=1.
                                         ;;*************************************************************


                                 ;  TEST 64
                                 ;--------------
015626  012737  000064  001226   TST64:  MOV     #64,TSTNO
015634  012737  015704  001216           MOV     #TST65,NEXT
015642  104412                           MSTCLR                  ;CLEAR DV11
015644  012777  000010  163510           MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
015652  012777  077400  163520           MOV     #BRB+BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                 ;BR=B "GROUND"?
015660  017702  163514                   MOV     @DVSFR,R2       ;READ DVSFR INTO R2
015664  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015670  017704  163474                   MOV     @DVLCR,R4       ;READ REAL RESULTS
015674  020504                           CMP     R5,R4           ;SAME??
015676  001401                           BEQ     1$              ;
015700  104006                           HLT     6               ;BR TEST POINT WRONG
015702  104400                   1$:     SCOPE                   ;SCOPE THIS TEST

                                         ;*********************** TEST 65 *****************************
                                         ;*TEST OF BRANCH B
                                         ;*CHECKING "DEFAULT" STATES OF THE DV11 SIGNALS.
                                         ;*BIT11 BIT10   BIT09   BIT08   FUNCTION
                                         ;*  1     0       0       0      DATA NOT AVAIL.
                                         ;*  1     0       0       1      REQUEST BUS
                                         ;*  1     0       1       0      MEMORY PARITY ERROR
                                         ;*  1     1       1       1      WRITE INHIBIT
                                         ;;*************************************************************


                                 ;  TEST 65
                                 ;--------------
```

```
015704  012737  000065  001226   TST65:  MOV     #65,TSTNO
015712  012737  016072  001216           MOV     #TST66,NEXT
015720  104412                           MSTCLR
015722  013700  001400                   MOV     DVSFR,R0
015726  013703  001370                   MOV     DVLCR,R3
015732  012777  000010  163422           MOV     #BIT3,@DVSCR
015740  012710  074000           1$:     MOV     #BRB+BIT11,(R0)
015744  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015746  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
015752  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
015754  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
015760  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
015762  001401                           BEQ     2$              ;BR IF OK
015764  104006                           HLT     6               ;BR POINT WRONG
015766  012710  074400           2$:     MOV     #BRB+BIT11+BIT8,(R0)
015772  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
015774  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016000  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
016002  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
016006  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
016010  001401                           BEQ     3$              ;BR IF OK
016012  104006                           HLT     6               ;BR POINT WRONG
016014  012710  075000           3$:     MOV     #BRB+BIT11+BIT9,(R0)
016020  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016022  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016026  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
016030  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
016034  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
016036  001401                           BEQ     4$              ;BR IF OK
016040  104006                           HLT     6               ;BR POINT WRONG
016042  012710  077000           4$:     MOV     #BRB+BIT11+BIT10+BIT9,(R0)
016046  011002                           MOV     (R0),R2         ;READ DVSFR FOR PRINTOUT
016050  012705  000003                   MOV     #BIT1+BIT0,R5   ;SET EXPECTED RESULTS
016054  011304                           MOV     (R3),R4         ;READ DVLCR INTO R4
016056  042704  177774                   BIC     #^C<BIT1+BIT0>,R4
                                                                 ;CLEAR UNWANTED BITS.
016062  020504                           CMP     R5,R4           ;EXPECTED = FOUND??
016064  001401                           BEQ     5$              ;BR IF OK
016066  104006                           HLT     6               ;BR POINT WRONG
016070  104400                   5$:     SCOPE




                                         ;*********************** TEST 66 *****************************
                                         ;*BASIC TEST OF THE
                                         ;*"SET/CLEAR INSTRUCTION.
                                         ;*TEST THAT THE SET/CLEAR CAN DO:
                                         ;*CLEAR DVSCR 08
                                         ;*SET DVSCR10
                                         ;*SET RECEIVER INTERUPT (DVSCR07)
                                         ;;*************************************************************


                                 ;  TEST 66
```

```
                                        ;--------------
 016072  012737  000066  001226  TST66:  MOV     #66,TSTNO
 016100  012737  016264  001216          MOV     #TST67,NEXT
 016106  012737  016136  001220          MOV     #1$,LOCK
 016114  104412                          MSTCLR                  ;CLEAR DV11
 016116  052777  000400  163236          BIS     #BIT8,@DVSCR    ;SET BIT8.
 016124  052777  000010  163230          BIS     #BIT3,@DVSCR    ;SET SOURCE SEL.
 016132  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER ADDRESS IN R0
 016136  012710  050016          1$:     MOV     #S.C+BIT3+BIT2+BIT1,(R0)
 016142  012705  000010                  MOV     #BIT3,R5        ;DO SET/CLEAR -CLEAR BIT 8 OF DVSCR
 016146  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016150  104415                          ROMCLK                  ;CYCLE THE ROM
 016152  017704  163204                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
 016156  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016160  001401                          BEQ     64$             ;BR IF DVSCR OK
 016162  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016164  104401                  64$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016166  012737  016174  001220          MOV     #3$,LOCK        ;SET FOR RETURN IF SW09=1
 016174  052705  000200          3$:     BIS     #BIT7,R5        ;SET EXPECTED (SCR BIT 7=1)
 016200  012710  050013                  MOV     #S.C+BIT3+BIT1+BIT0,(R0)
 016204  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016206  104415                          ROMCLK                  ;CYCLE THE ROM
 016210  017704  163146                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
 016214  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016216  001401                          BEQ     65$             ;BR IF DVSCR OK
 016220  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016222  104401                  65$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016224  012737  016232  001220          MOV     #4$,LOCK
 016232  052705  002000          4$:     BIS     #BIT10,R5       ;ALTER EXPECTED ADDRESS
 016236  012710  050012                  MOV     #S.C+BIT3+BIT1,(R0)
                                                                 ;DO A SET/CLEAR SET DVSCR BIT 10
 016242  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016244  104415                          ROMCLK                  ;CYCLE THE ROM
 016246  017704  163110                  MOV     @DVSCR,R4       ;READ DVSCR INTO "FOUND LOC.
 016252  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016254  001401                          BEQ     66$             ;BR IF DVSCR OK
 016256  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016260  104401                  66$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016262  104400                          SCOPE                   ;SCOPE THIS TEST


                                         ;*********************** TEST 67 ******************************
                                         ;*BASIC TEST OF THE
                                         ;*"SET/CLEAR INSTRUCTION.
                                         ;*TEST THAT THE SET/CLEAR CAN:
                                         ;*
                                         ;*BIT 14  BIT12  BIT03  BIT02  BIT01  BIT00  FUNCTION
                                         ;*  1       1      1      0      0      0     SET RICR 15
                                         ;*  1       1      1      0      0      1     SET RICR 14
                                         ;*  1       1      1      1      0      0     SET RICR 13
                                         ;*  1       1      1      1      0      1     SET RICR 12
                                         ;*
                                         ;;**************************************************************


                                 ;  TEST 67
                                 ;--------------
```

```
 016264  012737  000067  001226  TST67:  MOV     #67,TSTNO
 016272  012737  016506  001216          MOV     #TST70,NEXT
 016300  104412                          MSTCLR                  ;CLEAR DV11
 016302  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER TO R0
 016306  012777  000010  163046          MOV     #BIT3,@DVSCR    ;SET SORCE SEL
 016314  012705  100000                  MOV     #BIT15,R5       ;SET EXPECTED RESULTS
 016320  012710  050010                  MOV     #S.C+BIT3,(R0)
                                                                 ;SET/CLEAR DVRICR 15
 016324  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016326  104415                          ROMCLK                  ;CYCLE THE ROM
 016330  017704  163032                  MOV     @DVRIC,R4       ;READ DVRIC INTO "FOUND" LOC.
 016334  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016336  001401                          BEQ     64$             ;BR IF DVSCR OK
 016340  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016342  104401                  64$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016344  104412                          MSTCLR                  ;CLEAR DV11
 016346  012777  000010  163006          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 016354  012705  040000                  MOV     #BIT14,R5       ;SET EXPECTED RESULTS
 016360  012710  050011                  MOV     #S.C+BIT3+BIT0,(R0)
                                                                 ;SET/CLEAR DVRICR 14
 016364  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016366  104415                          ROMCLK                  ;CYCLE THE ROM
 016370  017704  162772                  MOV     @DVRIC,R4       ;READ DVRIC INTO "FOUND" LOC.
 016374  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016376  001401                          BEQ     65$             ;BR IF DVSCR OK
 016400  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016402  104401                  65$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016404  104412                          MSTCLR                  ;CLEAR DV11
 016406  012777  000010  162746          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 016414  012705  020000                  MOV     #BIT13,R5       ;SET EXPECTED RESULTS
 016420  012710  050014                  MOV     #S.C+BIT3+BIT2,(R0)
                                                                 ;SET/CLEAR DVRICR 13
 016424  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016426  104415                          ROMCLK                  ;CYCLE THE ROM
 016430  017704  162732                  MOV     @DVRIC,R4       ;READ DVRIC INTO "FOUND" LOC.
 016434  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016436  001401                          BEQ     66$             ;BR IF DVSCR OK
 016440  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016442  104401                  66$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016444  104412                          MSTCLR                  ;CLEAR DV11
 016446  012777  000010  162706          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 016454  012705  010000                  MOV     #BIT12,R5       ;SET EXPECTED RESULTS
 016460  012710  050015                  MOV     #S.C+BIT3+BIT2+BIT0,(R0)
                                                                 ;SET/CLEAR DVRICR 12
 016464  011002                          MOV     (R0),R2         ;SAVE DVSFR CONTENTS FOR ERROR PRINT OUT IF NECESSARY
 016466  104415                          ROMCLK                  ;CYCLE THE ROM
 016470  017704  162672                  MOV     @DVRIC,R4       ;READ DVRIC INTO "FOUND" LOC.
 016474  020504                          CMP     R5,R4           ;WAS THE ROM INSTR EXECUTED?
 016476  001401                          BEQ     67$             ;BR IF DVSCR OK
 016500  104006                          HLT     6               ;ROM FAILED TO EXECUTE
 016502  104401                  67$:    SCOP1                   ;LOCK ON THIS SUB-TEST? SW09=1?
 016504  104400                          SCOPE                   ;SCOPE THIS TEST


                                         ;*********************** TEST 70 ******************************
                                         ;*BASIC TEST OF DVSFR.
```

```
                                        ;*TEST OF "SET/CLEAR" AND
                                        ;*"BRANCH A" AND "BRANCH B" FUNCTIONS.
                                        ;;*************************************************************

                                  ;  TEST 70
                                  ;--------------
016506  012737  000070  001226   TST70:  MOV     #70,TSTNO
016514  012737  017170  001216           MOV     #TST71,NEXT
016522  012737  016544  001220           MOV     #1$,LOCK
016530  104412                           MSTCLR                  ;CLEAR DV11
016532  012777  000010  162622           MOV     #BIT3,@DVSCR    ;SET SOURCE SELECT
016540  013700  001400                   MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR ALU BIT02
016544  004237  017134           1$:     JSR     R2,10$          ;GOTO SUBROUTINE.
016550  000024                           BIT4+BIT2               ;POINT SET/CLEARED
016552  006000                           BIT11+BIT10             ;BR TEST POINT
016554  000002                           BIT1                    ;EXPECTED RESULTS IN DVLCR
016556  004237  017134                   JSR     R2,10$          ;GOSUB
016562  000025                           BIT4+BIT2+BIT0          ;POINT SET/CLEARED
016564  006000                           BIT11+BIT10             ;BR TEST POINT
016566  000003                           BIT1+BIT0               ;EXPECTED RESUTS IN DVLCR
016570  104401                           SCOP1                   ;SW09=1??

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT00
016572  012737  016600  001220           MOV     #2$,LOCK        ;SET RETURN IF SW09=1
016600  004237  017134           2$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016604  000047                           BIT5+BIT2+BIT1+BIT0             ;POINT SET/CLEARED [SET]
016606  070000                           BRB                     ;BR TEST POINT
016610  000001                           BIT0                    ;DVLCR EXPECTED
016612  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
016616  000043                           BIT5+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
016620  070000                           BRB                     ;BR TEST POINT
016622  000003                           BIT1+BIT0               ;EXPECTED RESULTS
016624  104401                   65$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT01
016626  012737  016634  001220           MOV     #3$,LOCK        ;SET RETURN IF SW09=1
016634  004237  017134           3$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016640  000045                           BIT5+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
016642  070400                           BRB+BIT8                        ;BR TEST POINT
016644  000001                           BIT0                    ;DVLCR EXPECTED
016646  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
016652  000041                           BIT5+BIT0                       ;POINT SET/CLEARED [CLEARED]
016654  070400                           BRB+BIT8                        ;BR TEST POINT
016656  000003                           BIT1+BIT0               ;EXPECTED RESULTS
016660  104401                   67$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT02
016662  012737  016670  001220           MOV     #4$,LOCK        ;SET RETURN IF SW09=1
016670  004237  017134           4$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016674  000046                           BIT5+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
```

```
016676  071000                           BRB+BIT9                        ;BR TEST POINT
016700  000001                           BIT0                    ;DVLCR EXPECTED
016702  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
016706  000042                           BIT5+BIT1                       ;POINT SET/CLEARED [CLEARED]
016710  071000                           BRB+BIT9                        ;BR TEST POINT
016712  000003                           BIT1+BIT0               ;EXPECTED RESULTS
016714  104401                   69$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT03
016716  012737  016724  001220           MOV     #6$,LOCK        ;SET RETURN IF SW09=1
016724  004237  017134           6$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016730  000044                           BIT5+BIT2                       ;POINT SET/CLEARED [SET]
016732  071400                           BRB+BIT9+BIT8                   ;BR TEST POINT
016734  000001                           BIT0                    ;DVLCR EXPECTED
016736  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
016742  000040                           BIT5                    ;POINT SET/CLEARED [CLEARED]
016744  071400                           BRB+BIT9+BIT8                   ;BR TEST POINT
016746  000003                           BIT1+BIT0               ;EXPECTED RESULTS
016750  104401                   71$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT04
016752  012737  016760  001220           MOV     #7$,LOCK        ;SET RETURN IF SW09=1
016760  004237  017134           7$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
016764  000207                           BIT7+BIT2+BIT1+BIT0             ;POINT SET/CLEARED [SET]
016766  072000                           BRB+BIT10                       ;BR TEST POINT
016770  000001                           BIT0                    ;DVLCR EXPECTED
016772  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
016776  000203                           BIT7+BIT1+BIT0                  ;POINT SET/CLEARED [CLEARED]
017000  072000                           BRB+BIT10                       ;BR TEST POINT
017002  000003                           BIT1+BIT0               ;EXPECTED RESULTS
017004  104401                   73$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT05
017006  012737  017014  001220           MOV     #8$,LOCK        ;SET RETURN IF SW09=1
017014  004237  017134           8$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
017020  000205                           BIT7+BIT2+BIT0                  ;POINT SET/CLEARED [SET]
017022  072400                           BRB+BIT10+BIT8                  ;BR TEST POINT
017024  000001                           BIT0                    ;DVLCR EXPECTED
017026  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
017032  000201                           BIT7+BIT0                       ;POINT SET/CLEARED [CLEARED]
017034  072400                           BRB+BIT10+BIT8                  ;BR TEST POINT
017036  000003                           BIT1+BIT0               ;EXPECTED RESULTS
017040  104401                   75$:    SCOP1                   ;SWR 09=1?

                                        ;*TEST SET/CLEAR FUNCTION
                                        ;*FOR RAM OUTPUT BIT06
017042  012737  017050  001220           MOV     #9$,LOCK        ;SET RETURN IF SW09=1
017050  004237  017134           9$:     JSR     R2,10$  ;GOTO THE SUBROUTINE
017054  000206                           BIT7+BIT2+BIT1                  ;POINT SET/CLEARED [SET]
017056  073000                           BRB+BIT10+BIT9                  ;BR TEST POINT
017060  000001                           BIT0                    ;DVLCR EXPECTED
017062  004237  017134                   JSR     R2,10$          ;GOTO SUB ROUTINE
017066  000202                           BIT7+BIT1                       ;POINT SET/CLEARED [CLEARED]
```

```
017070  073000                           BRB+BIT10+BIT9               ;BR TEST POINT
017072  000003                           BIT1+BIT0              ;EXPECTED RESULTS
017074  104401                   77$:    SCOP1                  ;SWR 09=1?

                                         ;*TEST SET/CLEAR FUNCTION
                                         ;*FOR RAM OUTPUT BIT07
017076  012737  017104  001220           MOV     #101$,LOCK     ;SET RETURN IF SW09=1
017104  004237  017134           101$:   JSR     R2,10$  ;GOTO THE SUBROUTINE
017110  000204                           BIT7+BIT2                      ;POINT SET/CLEARED [SET]
017112  073400                           BRB+BIT10+BIT9+BIT8                      ;BR TEST POINT
017114  000001                           BIT0                   ;DVLCR EXPECTED
017116  004237  017134                   JSR     R2,10$         ;GOTO SUB ROUTINE
017122  000200                           BIT7                   ;POINT SET/CLEARED [CLEARED]
017124  073400                           BRB+BIT10+BIT9+BIT8                 ;BR TEST POINT
017126  000003                           BIT1+BIT0              ;EXPECTED RESULTS
017130  104401                   79$:    SCOP1                  ;SWR 09=1?
017132  104400                           SCOPE                  ;SCOPE THE TEST
017134  012710  050000           10$:    MOV     #S.C,(R0)      ;SET/CLEAR INSTR
017140  010201                           MOV     R2,R1          ;SAVE JSR PC ADDRESS
017142  052210                           BIS     (R2)+,(R0)     ;LOAD POINT SET/CLEARED
017144  104415                           ROMCLK                 ;CYCLE THE ROM
017146  012210                           MOV     (R2)+,(R0)     ;LOAD BR TEST POINT
017150  011003                           MOV     (R0),R3        ;READ DVSFR INTO R3
017152  012205                           MOV     (R2)+,R5       ;LOAD EXPECTED INTO R5
017154  017704  162210                   MOV     @DVLCR,R4      ;READ DVLCR INTO FOUND LOC.
017160  020504                           CMP     R5,R4          ;EXPECTED=FOUND?
017162  001401                           BEQ     11$            ;BR IF YES
017164  104005                           HLT     5              ;DVLCR WRONG BR RESULTS.
017166  000202                   11$:    RTS     R2             ;RETURN

                                         ;*********************** TEST 71 ****************************
                                         ;*TEST OF "RECEIVER CHARACTER SILO"
                                         ;*THRU THE USE OF THE DVSFR REG.
                                         ;*TEST THE FILLING THE SILO PRODUCES "SILO FULL"
                                         ;*ON EXACTLY THE 128 LOAD.
                                         ;*SET/CLEAR IS USED TO STUFF SILO AND BRANCH A IS USED TO TEST SILO.
                                         ;*SET/CLEAR "SILO IN" AND SET/CLEAR "SILO OUT" ARE EXERCISED TOO.
                                         ;;************************************************************


                                 ;  TEST 71
                                 ;--------------
017170  012737  000071  001226   TST71:  MOV     #71,TSTNO
017176  012737  017412  001216           MOV     #TST72,NEXT
017204  104412                           MSTCLR                 ;CLEAR DV11
017206  012700  000177                   MOV     #127.,R0       ;SET R0 TO 1 LESS THAN FULL SILO
017212  012777  000010  162142           MOV     #BIT3,@DVSCR   ;SET SOURCE SEL
017220  012705  000003                   MOV     #BIT1+BIT0,R5  ;SET EXPECTED RESULTS INTO R5
017224  012777  050021  162146   1$:     MOV     #S.C+BIT4+BIT0,@DVSFR
017232  104415                           ROMCLK                 ;S/C "SILO IN"
017234  012777  007400  162136           MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                ;BR=A "SILO FULL"?
017242  017702  162132                   MOV     @DVSFR,R2      ;SAVE CONTENTS OF DVSFR FOR ERROR PRINTOUT
017246  017704  162116                   MOV     @DVLCR,R4      ;READ DVLCR FOR RESULTS
017252  020504                           CMP     R5,R4          ;ARE BR TEST POINTS CORRECT?
017254  001401                           BEQ     64$            ;BR IF YES
017256  104006                           HLT     6              ;BR TEST POINTS WRONG (BIT1 OR 0)
```

```
017260  005300                   64$:    DEC     R0             ;IS SILO FULL-1 YET?
017262  001360                           BNE     1$             ;BR IF NOT 127 TIMES YET
017264  012777  050021  162106   2$:     MOV     #S.C+BIT4+BIT0,@DVSFR
                                                                ;S/C "SILO IN"
017272  104415                           ROMCLK                 ;
017274  000240                           NOP                    ;WAIST INSTRUCTION TIME
017276  012777  007400  162074           MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                ;BR=A "SILO FULL"?
017304  017702  162070                   MOV     @DVSFR,R2      ;SAVE DVSFR
017310  017704  162054                   MOV     @DVLCR,R4      ;READ BR TEST POINTS
017314  042705  000001                   BIC     #BIT0,R5       ;ALTER EXPECTED RESULTS
017320  020504                           CMP     R5,R4          ;BR TEST POINTS OK??
017322  001401                           BEQ     3$             ;BR IF YES
017324  104006                           HLT     6              ;BR TEST POINTS WRONG
017326  012777  050020  162044   3$:     MOV     #S.C+BIT4,@DVSFR
017334  104415                           ROMCLK                 ;S/C "SILO OUT"
017336  000240                           NOP                    ;WAIST INSTR TIME
017340  012777  007400  162032           MOV     #BIT11+BIT10+BIT9+BIT8,@DVSFR
                                                                ;BR=A "SILO FULL"?
017346  005002                           CLR     R2             ;DEALAY AT LEAST 32US
017350  032777  000001  162012   4$:     BIT     #BIT0,@DVLCR   ;IS SILO *NOT FULL*??
017356  001003                           BNE     5$             ;BR IF OK.
017360  062702  000001                   ADD     #1,R2          ;DELAY.......
017364  001371                           BNE     4$             ;GOTO 4$
017366  017702  162006           5$:     MOV     @DVSFR,R2      ;SAVE DVSFR
017372  017704  161772                   MOV     @DVLCR,R4      ;READ BR TEST POINTS
017376  052705  000001                   BIS     #BIT0,R5       ;SET EXPECTED RESULTS
017402  020504                           CMP     R5,R4          ;OK??
017404  001401                           BEQ     6$             ;YES
017406  104006                           HLT     6              ;SILO  STILL FULL.
017410  104400                   6$:     SCOPE                  ;SCOPE TEST

                                         ;*********************** TEST 72 ****************************
                                         ;*TEST THAT AFTER AN INIT
                                         ;*THAT "RECV CHARACTER WAITING"
                                         ;*IS FALSE (HIGH) AND THEN VERIFY
                                         ;*THAT WHEN "SILO IN" IS ASSERTED THAT
                                         ;*THAT "RCVED CHARACTER WAITING" IS TRUE (LOW)
                                         ;*AND MAKES "BRANCH A" TRUE.
                                         ;;************************************************************


                                 ;  TEST 72
                                 ;--------------
017412  012737  000072  001226   TST72:  MOV     #72,TSTNO
017420  012737  017616  001216           MOV     #TST73,NEXT
017426  104412                           MSTCLR                 ;CLEAR DV11
017430  012777  000010  161724           MOV     #BIT3,@DVSCR   ;SET SOURCE SEL
017436  012705  000003                   MOV     #BIT1+BIT0,R5  ;SET EXPECTED RESULTS
017442  012702  001400                   MOV     #BIT9+BIT8,R2  ;BR=A "RCVD CHAR WAITING"?
017446  010277  161726                   MOV     R2,@DVSFR      ;LOAD DV INSTR
017452  017704  161712                   MOV     @DVLCR,R4      ;READ TEST POINTS
017456  020504                           CMP     R5,R4          ;OK??
017460  001401                           BEQ     64$            ;YES
017462  104006                           HLT     6              ;TEST POINT RECV CHAR WAITING WRONG
017464  012702  050021           64$:    MOV     #S.C+BIT4+BIT0,R2
                                                                ;S/C "SILO IN"
```

```
017470  010277  161704                  MOV     R2,@DVSFR       ;LOAD INSTR
017474  104415                          ROMCLK                  ;CLOCK
017476  005004                          CLR     R4              ;PREPARE COUNTER
017500  012702  001400                  MOV     #BIT9+BIT8,R2   ;BR-A RCV CHAR WAITING
                                                                ;BR-A "RCVD CHAR WAITING"?
017504  010277  161670                  MOV     R2,@DVSFR       ;LOAD INSTR
017510  012705  000002                  MOV     #BIT1,R5        ;SET GOOD RESULTS
017514  032777  000001  161646  1$:     BIT     #BIT0,@DVLCR    ;TEST DV11 BR POINT
017522  001403                          BEQ     2$              ;BR IF OK
017524  062704  000001                  ADD     #1,R4           ;DELAY
017530  001371                          BNE     1$              ;GOTO 1$
017532  017704  161632          2$:     MOV     @DVLCR,R4       ;READ DV11 BR POINT
017536  020504                          CMP     R5,R4           ;
017540  001401                          BEQ     3$
017542  104006                          HLT     6               ;BR POINT RCV CHAR WAITING WRONG
                                        ;*TEST THAT SETTING DVSCR07
                                        ;*INHIBITS RCV CHAR WAITING FROM APPEARING
                                        ;*TRUE; AND THAT CLEARING
                                        ;*DVSCR07 MAKES IT APPEAR TRUE AGAIN.

017544  012705  000003          3$:     MOV     #BIT1+BIT0,R5   ;LOAD EXPECTED
017550  052777  001200  161604          BIS     #BIT9+BIT7,@DVSCR       ;SET RECV INTER
017556  017704  161606                  MOV     @DVLCR,R4       ;READ DV BR POINTS
017562  020504                          CMP     R5,R4           ;
017564  001401                          BEQ     4$              ;
017566  104006                          HLT     6               ;BR TEST POINTS WRONG
017570  042705  000001          4$:     BIC     #BIT0,R5        ;RESET EXPECTED RESULTS
017574  042777  000200  161560          BIC     #BIT7,@DVSCR    ;CLEAR RECV INT
017602  017704  161562                  MOV     @DVLCR,R4       ;READ BR POINTS
017606  020504                          CMP     R5,R4           ;
017610  001401                          BEQ     5$              ;
017612  104006                          HLT     6               ;BR TEST POINTS WRONG
017614  104400                  5$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*********************** TEST 73 ****************************
                                        ;*BASIC TEST OF THE "DATA TRANSFER INSTRUCTION"
                                        ;*BITS 07,06,05,04 OF DVSFR INDICATE THE SOURCE
                                        ;*BITS 03,02,01,00 OF DVSFR INDICATE THE DESTINATION.
                                        ;;*************************************************************

                                ;  TEST 73
                                ;---------------
017616  012737  000073  001226  TST73:  MOV     #73,TSTNO
017624  012737  020156  001216          MOV     #TST74,NEXT
017632  012737  017732  001220          MOV     #1$,LOCK
017640  104412                          MSTCLR                  ;CLEAR DV11
017642  012777  000010  161512          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
017650  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER INTO R0

                                        ;*TEST TO XFR SOURCE REGISTERS TO THE DVRIC
                                        ;*REGISTER VERIFYING THAT THE FOLLOWING REGISTERS
                                        ;*ARE CLEARED AND THAT THE XFR BUS IS CLEAR AFTER
                                        ;*A MSTCLR.
                                        ;*REGISTER      FUNCTION
                                        ;* 0000         GROUND
```

```
                                        ;* 0001         GROUND
                                        ;* 0010         GROUND
                                        ;* 0011         GROUND
                                        ;* 0100         GROUND
                                        ;* 0101         MASTER SCAN 0-3/0-3
                                        ;* 0110         ALU RESULT 8-11/0-3
                                        ;* 0111         ALU RESULT 5-7/0-2
                                        ;* 1000         LOW BYTE=B REG 8-15 ; HIGH BYTE=GRND
                                        ;* 1001         LO BYTE=NPR OUT ; HI BYTE=CDC REG
                                        ;* 1010         RAM OUTPUT 0-2/8-10
                                        ;* 1011         RAM OUTPUT
                                        ;* 1101         NPR INPUT REGISTER
                                        ;* 1110         BCC REGISTER
                                        ;* 1111         ALU RESULT REGISTER

017654  005005                          CLR     R5              ;SET EXPECTED TO 0
017656  012702  030000                  MOV     #XFR,R2         ;SET DATA XFR INSTR.
017662  052702  000006                  BIS     #BIT2+BIT1,R2   ;SET DESTINATION TO DVRIC REG.
017666  005003                          CLR     R3              ;ZERO SOURCE REG POINTER
017670  042702  000360          65$:    BIC     #BIT7+BIT6+BIT5+BIT4,R2
017674  050302                          BIS     R3,R2           ;SET SOURCE REGISTER
017676  010210                          MOV     R2,(R0) ;LOAD SFR WITH XFR INSTR
017700  104415                          ROMCLK                  ;EXECUTE INSTR
017702  017704  161460                  MOV     @DVRIC,R4       ;READ SOURCE REGISTER
017706  001401                          BEQ     66$             ;BR IF IT WAS ZERO
017710  104006                          HLT     6               ;SOURCE REGISTER IN SFR NOT ZERO
017712  062703  000020          66$:    ADD     #BIT4,R3        ;UPDATE SOURCE REGISTER
017716  022703  000300                  CMP     #300,R3         ;DON'T DO SILO REGISTER!!
017722  001773                          BEQ     66$             ;GET NEXT REG IF THIS IS SILO.
017724  032703  000360                  BIT     #BIT7+BIT6+BIT5+BIT4,R3
017730  001357                          BNE     65$             ;BR IF MORE TO DO.

                                        ;*TEST OF SET RAM OUTPUT BIT0
                                        ;*AND THE USE OF THE DATA XFER INSTR.
                                        ;*PLACE RAM BIT0 INTO THE DVRIC REG
017732  012705  000400          1$:     MOV     #BIT8,R5
017736  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
017742  052703  000246                  BIS     #BIT7+BIT5+BIT2+BIT1,R3
017746  004237  020076                  JSR     R2,10$          ;S= RAM OUTPUT 0-2, D= DVRIC
017752  000047                          BIT5+BIT2+BIT1+BIT0
017754  000043                          BIT5+BIT1+BIT0

                                        ;*TEST TO SET RAM OUTPUT DATA BIT3
                                        ;*AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                        ;*TO PLACE BIT3 INTO THE DVRIC REGISTER.
017756  012737  017764  001220          MOV     #3$,LOCK        ;SET RETURN IF SW09=1
017764  012705  000010          3$:     MOV     #BIT3,R5        ;SET EXPECTED DATA
017770  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
017774  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
020000  004237  020076                  JSR     R2,10$          ;S= RAM OUTPUT, D=DVRIC
020004  000044                          BIT5+BIT2
020006  000040                          BIT5

                                        ;*TEST TO SET RAM OUTPUT DATA BIT4
                                        ;*AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                        ;*TO PLACE BIT4 INTO THE DVRIC REGISTER.
```

```
020010  012737  020016  001220          MOV     #4$,LOCK         ;SET RETURN IF SW09=1
020016  012705  000020          4$:     MOV     #BIT4,R5         ;SET EXPECTED DATA
020022  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
020026  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
020032  004237  020076                  JSR     R2,10$           ;S= RAM OUTPUT, D=DVRIC
020036  000207                          BIT7+BIT2+BIT1+BIT0
020040  000203                          BIT7+BIT1+BIT0

                                        ;*TEST TO SET RAM OUTPUT DATA BIT7
                                        ;*AND THE USE OF THE "DATA TRANSFER" INSTRUCTION
                                        ;*TO PLACE BIT7 INTO THE DVRIC REGISTER.
020042  012737  020050  001220          MOV     #6$,LOCK         ;SET RETURN IF SW09=1
020050  012705  000200          6$:     MOV     #BIT7,R5         ;SET EXPECTED DATA
020054  012703  030000                  MOV     #XFR,R3 ;-DATA XFER-
020060  052703  000266                  BIS     #BIT7+BIT5+BIT4+BIT2+BIT1,R3
020064  004237  020076                  JSR     R2,10$           ;S= RAM OUTPUT, D=DVRIC
020070  000204                          BIT7+BIT2
020072  000200                          BIT7
020074  104400                          SCOPE
020076  012710  050000          10$:    MOV     #S.C,(R0) ;SET CLEAR INSTR
020102  010201                          MOV     R2,R1            ;JSR PC TO R1
020104  052210                          BIS     (R2)+,(R0)       ;LOAD SET/CLEAR POINT
020106  104415                          ROMCLK                   ;EXECUTE
020110  010310                          MOV     R3,(R0)          ;LOAD XFER
020112  104415                          ROMCLK                   ;EXECUTE
020114  017704  161246                  MOV     @DVRIC,R4        ;READ RESULTS
020120  020504                          CMP     R5,R4            ;OK??
020122  001401                          BEQ     11$              ;YES
020124  104005                          HLT     5                ;XFER FAILED
020126  012710  050000          11$:    MOV     #S.C,(R0) ;SET/CLEAR
020132  052210                          BIS     (R2)+,(R0)       ;POINT
020134  104415                          ROMCLK                   ;EXECUTE
020136  010310                          MOV     R3,(R0)          ;XFER
020140  104415                          ROMCLK                   ;EXECUTE
020142  005005                          CLR     R5               ;SET EXPECTED RESULTS
020144  017704  161216                  MOV     @DVRIC,R4        ;READ REAL RESULTS
020150  001401                          BEQ     12$              ;BR IF RESULT=0
020152  104005                          HLT     5                ;DVRICR NOT =0
020154  000202                  12$:    RTS     R2               ;EXIT SUB


                                        ;************************ TEST 74 *****************************
                                        ;*BASIC TEST OF THE "ALU OPERATION" INSTRUCTION.
                                        ;*FIRST PART:ISSUE AN INIT AND MOVE
                                        ;*THE ALU RESULT REGISTER TO THE DVRIC
                                        ;*REGISTER VERIFING THAT IT IS ZERO.
                                        ;*SECOND PART: DO A FUNCTION "F=A"
                                        ;*THEN MOV "F" TO THE DVRIC REGISTER VERIFING IT TO
                                        ;*BE ZERO
                                        ;*THIRD PART: DO A FUNCTION "F=A+B", MOVING
                                        ;*"F" TO DVRIC AND MAKING SURE IT IS ZERO.
                                        ;*THUS THE FOLLOWING HAS BEEN TESTED:
                                        ;*ALU RESULT,"A" REG,AND "B" REG ALL ZEROED ON INIT.
                                        ;:**************************************************************


                                ;  TEST 74
```

```
                                ;---------------
020156  012737  000074  001226  TST74:  MOV     #74,TSTNO
020164  012737  020276  001216          MOV     #TST75,NEXT
020172  104412                          MSTCLR                   ;RESET DV11
020174  012777  000010  161160          MOV     #BIT3,@DVSCR     ;SET SOURCE SELECT
020202  013700  001400                  MOV     DVSFR,R0         ;SET DVSFR POINTER IN R0
020206  012702  030366                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
020212  010210                          MOV     R2,(R0)          ;XFR "ALU RESULT REG." TO DVNSR
020214  104415                          ROMCLK                   ;CLOCK INSTRUCTION
020216  005005                          CLR     R5               ;ZERO "EXPECTED" LOC
020220  017704  161142                  MOV     @DVRIC,R4        ;READ "ALU RESULT"
020224  001401                          BEQ     1$               ;S/B=0
020226  104006                          HLT     6                ;ALU RESULT NOT=0 ON INIT
020230  012710  010037          1$:     MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,(R0)
020234  104415                          ROMCLK                   ;DO "ALU F=A"
020236  010210                          MOV     R2,(R0)          ;XFR "ALU RESULT" TO DVNSR
020240  104415                          ROMCLK                   ;CLOCK INSTR
020242  017704  161120                  MOV     @DVRIC,R4        ;READ RESULT
020246  001401                          BEQ     2$               ;S/B=0
020250  104006                          HLT     6                ;"A" REG NOT=0 ON INIT
020252  012710  010026          2$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
020256  104415                          ROMCLK                   ;ALU F=A+B
020260  010210                          MOV     R2,(R0)          ;XFR TO RIC
020262  104415                          ROMCLK                   ;CLOCK INSTR.
020264  017704  161076                  MOV     @DVRIC,R4        ;READ RESULT
020270  001401                          BEQ     3$               ;S/B=0
020272  104006                          HLT     6                ;"B" REG NOT=0 ON INIT
020274  104400                  3$:     SCOPE                    ;SCOPE TEST

                                        ;************************ TEST 75 *****************************
                                        ;*TEST OF ALU OPERATIONS.
                                        ;*TEST OF ALL ALU OPERATIONS USED BY DV11.
                                        ;*FUNCTIONS TESTED:(NOTE THAT "F" IS ALU RESULT)
                                        ;*DVSFR BITS:
                                        ;*BIT12 BIT05 BIT04 BIT03 BIT02 BIT01 BIT00 FUNCTION
                                        ;*  1     0     1     1     1     0     0    F=-1
                                        ;*  1     0     0     1     1     0     0    F=0
                                        ;*  1     0     1     1     1     1     1    F=A
                                        ;*  1     0     0     0     1     0     1    F=B
                                        ;*  1     1     1     1     1     1     1    F=A+1
                                        ;*  1     0     1     0     1     1     0    F=A+B
                                        ;:**************************************************************

                                ;  TEST 75
                                ;---------------
020276  012737  000075  001226  TST75:  MOV     #75,TSTNO
020304  012737  021206  001216          MOV     #TST76,NEXT
                                        ;*FUNCTION TESTED
                                        ;*F=-1,RIC_F
                                        ;*
020312  012737  020320  001220          MOV     #10$,LOCK        ;SET FOR SW09
020320  104412                  10$:    MSTCLR                   ;CLEAR DV11
020322  012777  000010  161032          MOV     #BIT3,@DVSCR     ;SET SOURCE SEL
020330  013700  001400                  MOV     DVSFR,R0         ;SET DVSFR POINTER IN R0
020334  012702  030366                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1,R2
020340  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
```

```
Ø2Ø344  1Ø4415                           ROMCLK                  ;DO "F=-1"
Ø2Ø346  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR "ALU RESULT TO DVRIC"
Ø2Ø35Ø  1Ø4415                           ROMCLK                  ;
Ø2Ø352  Ø177Ø4  161Ø1Ø                   MOV     @DVRIC,R4       ;READ RESULTS
Ø2Ø356  Ø127Ø5  177777                   MOV     #-1,R5          ;SET EXPECTED
Ø2Ø362  Ø2Ø5Ø4                           CMP     R5,R4           ;DID F=-1 WORK?
Ø2Ø364  ØØ14Ø1                           BEQ     63$             ;BR IF YES
Ø2Ø366  1Ø4ØØ6                           HLT     6               ;F=-1 APPEARED TO FAIL
                                         ;*TEST THAT DVRIC (NOW THAT ITS ALL 1'S)
                                         ;*CAN BE CLEARED BY A MSTCLR.
                                         ;*
Ø2Ø37Ø  1Ø44Ø1                   63$:    SCOP1                   ;SWØ9=1?
Ø2Ø372  Ø12737  Ø2Ø4ØØ  ØØ122Ø           MOV     #11$,LOCK       ;SET RETURN IF SWØ9=1
Ø2Ø4ØØ  1Ø4412                   11$:    MSTCLR                  ;CLEAR DV11
Ø2Ø4Ø2  ØØ5ØØ5                           CLR     R5              ;SET EXPECTED RESULTS
Ø2Ø4Ø4  Ø177Ø4  16Ø756                   MOV     @DVRIC,R4       ;READ DVRIC
Ø2Ø41Ø  ØØ14Ø1                           BEQ     64$             ;BR IF=Ø
Ø2Ø412  1Ø4ØØ6                           HLT     6               ;DVRIC REG. NOT CLEARED ON INIT
                                         ;*NEXT SET OF FUNCTIONS:
                                         ;*F=Ø,RIC_F
                                         ;*
Ø2Ø414  Ø12777  ØØØØ1Ø  16Ø74Ø   64$:    MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
Ø2Ø422  Ø1271Ø  Ø1ØØ34                   MOV     #ALU+BIT4+BIT3+BIT2,(RØ)
Ø2Ø426  1Ø4415                           ROMCLK                  ;"F=-1"
Ø2Ø43Ø  Ø1271Ø  Ø1ØØ14                   MOV     #ALU+BIT3+BIT2,(RØ)
Ø2Ø434  1Ø4415                           ROMCLK                  ;"F=Ø"
Ø2Ø436  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR "F" TO DVRIC
Ø2Ø44Ø  1Ø4415                           ROMCLK                  ;
Ø2Ø442  ØØ5ØØ5                           CLR     R5              ;SET EXPECTED
Ø2Ø444  Ø177Ø4  16Ø716                   MOV     @DVRIC,R4       ;READ RESULTS
Ø2Ø45Ø  ØØ14Ø1                           BEQ     65$             ;BR IF=Ø
Ø2Ø452  1Ø4ØØ6                           HLT     6               ;"F=Ø" APPEARED TO FAIL
                                         ;*NEXT SET OF FUNCTIONS:
                                         ;*F=A+1,RIC_F,A_F,F=Ø,F=A
                                         ;*
Ø2Ø454  1Ø44Ø1                   65$:    SCOP1                   ;SWØ9=1?
Ø2Ø456  Ø12737  Ø2Ø464  ØØ122Ø           MOV     #12$,LOCK       ;SET RETURN
Ø2Ø464  Ø127Ø5  ØØØØØ1           12$:    MOV     #1,R5           ;SET GOOD RESULTS
Ø2Ø47Ø  Ø1271Ø  Ø1ØØ77           1$:     MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BITØ,(RØ)
Ø2Ø474  Ø52777  ØØØØØ2  16Ø66Ø           BIS     #BIT1,@DVSCR    ;ISSUE ROM CLOCK
Ø2Ø5Ø2  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR "F" TO DVRIC
Ø2Ø5Ø4  Ø52777  ØØØØØ2  16Ø65Ø           BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
Ø2Ø512  Ø177Ø4  16Ø65Ø                   MOV     @DVRIC,R4       ;READ RESULTS
Ø2Ø516  Ø2Ø5Ø4                           CMP     R5,R4           ;FUNCTION WORK?
Ø2Ø52Ø  ØØ14Ø1                           BEQ     66$             ;YES
Ø2Ø522  1Ø4ØØ6                           HLT     6               ;F=A+1 APPEARED TO FAIL
Ø2Ø524  Ø1271Ø  Ø3Ø361           66$:    MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BITØ,(RØ)
Ø2Ø53Ø  Ø52777  ØØØØØ2  16Ø624           BIS     #BIT1,@DVSCR    ;ISSUE ROM CLK
Ø2Ø536  Ø1271Ø  Ø1ØØ14                   MOV     #ALU+BIT3+BIT2,(RØ)
Ø2Ø542  Ø52777  ØØØØØ2  16Ø612           BIS     #BIT1,@DVSCR    ;ROM CLK
Ø2Ø55Ø  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR RIC_F
Ø2Ø552  Ø52777  ØØØØØ2  16Ø6Ø2           BIS     #BIT1,@DVSCR    ;ROM CLK
Ø2Ø56Ø  Ø177Ø4  16Ø6Ø2                   MOV     @DVRIC,R4       ;READ RESULT
Ø2Ø564  ØØ14Ø1                           BEQ     67$             ;BR IF GOOD
Ø2Ø566  1Ø4ØØØ                           HLT     Ø               ;F=Ø FAILED
Ø2Ø57Ø  Ø1271Ø  Ø1ØØ37           67$:    MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BITØ,(RØ)
```

```
Ø2Ø574  Ø52777  ØØØØØ2  16Ø56Ø           BIS     #BIT1,@DVSCR    ;CLOCK "F=A"
Ø2Ø6Ø2  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR RIC_F
Ø2Ø6Ø4  Ø52777  ØØØØØ2  16Ø55Ø           BIS     #BIT1,@DVSCR    ;ROM CLK
Ø2Ø612  Ø177Ø4  16Ø55Ø                   MOV     @DVRIC,R4       ;READ RESULT
Ø2Ø616  Ø2Ø5Ø4                           CMP     R5,R4           ;BR IF OK
Ø2Ø62Ø  ØØ14Ø1                           BEQ     68$             ;
Ø2Ø622  1Ø4ØØ6                           HLT     6               ;F=A FAILED
Ø2Ø624  ØØ52Ø5                   68$:    INC     R5              ;UPDATE DATA
Ø2Ø626  ØØ132Ø                           BNE     1$              ;BR IF NOT ALL DONE.
Ø2Ø63Ø  1Ø44Ø1                           SCOP1                   ;SWØ9=1?
                                         ;*NEXT SET OF FUNCTIONS:
                                         ;*F=A+1,A_F,B_F,F=A+B,RIC_F
                                         ;*
Ø2Ø632  Ø12737  Ø2Ø64Ø  ØØ122Ø           MOV     #13$,LOCK       ;SET RETURN
Ø2Ø64Ø  1Ø4412                   13$:    MSTCLR                  ;RESET DV11
Ø2Ø642  Ø12777  ØØØØ1Ø  16Ø512           MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
Ø2Ø65Ø  Ø127Ø5  ØØØØØ2                   MOV     #2,R5           ;SET EXPECTED RESULTS
Ø2Ø654  Ø137Ø1  ØØ1366                   MOV     DVRIC,R1        ;SET POINTER
Ø2Ø66Ø  Ø137Ø3  ØØ1362                   MOV     DVSCR,R3        ;SET POINTER
Ø2Ø664  Ø1271Ø  Ø1ØØ77                   MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BITØ,(RØ)
Ø2Ø67Ø  1Ø4415                           ROMCLK                  ;F=A+1
Ø2Ø672  Ø1271Ø  Ø3Ø361                   MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BITØ,(RØ)
Ø2Ø676  1Ø4415                           ROMCLK                  ;XFR A_ALU RESULT
Ø2Ø7ØØ  Ø1271Ø  Ø3Ø362           2$:     MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(RØ)
Ø2Ø7Ø4  Ø52713  ØØØØØ2                   BIS     #BIT1,(R3)      ;XFR B_ALU RESULT
Ø2Ø71Ø  Ø1271Ø  Ø1ØØ26                   MOV     #ALU+BIT4+BIT2+BIT1,(RØ)
Ø2Ø714  Ø52713  ØØØØØ2                   BIS     #BIT1,(R3)      ;CLOCK F=A+B
Ø2Ø72Ø  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR RIC_F
Ø2Ø722  Ø52713  ØØØØØ2                   BIS     #BIT1,(R3)      ;
Ø2Ø726  Ø111Ø4                           MOV     (R1),R4         ;READ DVRIC
Ø2Ø73Ø  Ø2Ø5Ø4                           CMP     R5,R4           ;FUNCTION WORK?
Ø2Ø732  ØØ14Ø1                           BEQ     69$             ;BR IF YES
Ø2Ø734  1Ø4ØØ6                           HLT     6               ;F=A+B APPEARED TO FAIL
Ø2Ø736  ØØ52Ø5                   69$:    INC     R5              ;INC DATA POINTER
Ø2Ø74Ø  Ø227Ø5  ØØØØØ2                   CMP     #2,R5           ;ALL DONE?
Ø2Ø744  ØØ1355                           BNE     2$              ;BR IF NO
Ø2Ø746  1Ø4412                           MSTCLR                  ;RESET DV11
Ø2Ø75Ø  Ø177Ø4  16Ø412                   MOV     @DVRIC,R4       ;DVRIC ZERO ON INIT?
Ø2Ø754  ØØ14Ø1                           BEQ     7Ø$             ;BR IF YES
Ø2Ø756  1Ø4ØØØ                           HLT     Ø               ;S/B=Ø
Ø2Ø76Ø  1Ø44Ø1                   7Ø$:    SCOP1                   ;SWØ9=1?
                                         ;*NEXT SET OF FUNCTIONS:
                                         ;*F=-1,B_F,F=Ø,F=B
                                         ;*
Ø2Ø762  Ø12737  Ø2Ø77Ø  ØØ122Ø           MOV     #14$,LOCK       ;SET RETURN
Ø2Ø77Ø  1Ø4412                   14$:    MSTCLR                  ;RESET DV11
Ø2Ø772  Ø12777  ØØØØ1Ø  16Ø362           MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
Ø21ØØØ  Ø1271Ø  Ø1ØØ34                   MOV     #ALU+BIT4+BIT3+BIT2,(RØ)
Ø21ØØ4  1Ø4415                           ROMCLK                  ;F=-1
Ø21ØØ6  Ø1271Ø  Ø3Ø362                   MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(RØ)
Ø21Ø12  1Ø4415                           ROMCLK                  ;XFR B_F
Ø21Ø14  Ø1271Ø  Ø1ØØ14                   MOV     #ALU+BIT3+BIT2,(RØ) ;
Ø21Ø2Ø  1Ø4415                           ROMCLK                  ;F=Ø
Ø21Ø22  Ø1271Ø  Ø1ØØØ5                   MOV     #ALU+BIT2+BITØ,(RØ) ;
Ø21Ø26  1Ø4415                           ROMCLK                  ;F=B
Ø21Ø3Ø  Ø1Ø21Ø                           MOV     R2,(RØ)         ;XFR RIC_F
```

```
021032  104415                          ROMCLK                          ;CLOCK
021034  017704  160326                  MOV     @DVRIC,R4               ;READ RESULTS
021040  012705  177777                  MOV     #-1,R5                  ;SET EXPECTED
021044  020504                          CMP     R5,R4                   ;DID F=B WORK?
021046  001401                          BEQ     71$                     ;BR IF YES
021050  104006                          HLT     6                       ;F=B FAILED
021052  104401                  71$:    SCOP1                           ;SW09=1?
                                        ;*NEXT SET OF FUNCTIONS:
                                        ;*CATCH "SET/CLEAR" THAT WAS MISSED.
                                        ;*F=-1,S/C[ALU01=0],RIC_F
                                        ;*
021054  012737  021062  001220          MOV     #15$,LOCK               ;SET RETURN
021062  104412                  15$:    MSTCLR                          ;RESET DV11
021064  012777  000010  160270          MOV     #BIT3,@DVSCR            ;SET SOURCE SELECT
021072  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
021076  104415                          ROMCLK                          ;F=-1
021100  012710  050026                  MOV     #S.C+BIT4+BIT2+BIT1,(R0)
021104  104415                          ROMCLK                          ;S/C "CLEAR ALU01"
021106  010210                          MOV     R2,(R0)                 ;XFR RIC_F
021110  104415                          ROMCLK                          ;
021112  011104                          MOV     (R1),R4                 ;READ DVRIC
021114  012705  177775                  MOV     #177775,R5              ;SET EXPECTED
021120  020504                          CMP     R5,R4                   ;DID S/C ALU01 WORK?
021122  001401                          BEQ     72$                     ;BR IF YES
021124  104006                          HLT     6                       ;S/C ALU01 FAILED.
021126  104401                  72$:    SCOP1                           ;SW09=1?
                                        ;*NEXT SET OF FUNCTIONS:
                                        ;*CATCH ANOTHER "SET/CLEAR" THAT WAS MISSED.
                                        ;*F=-1,S/C[ALU HIGH BYTE=0],RIC_F
                                        ;*
021130  012737  021136  001220          MOV     #16$,LOCK               ;SET RETURN
021136  104412                  16$:    MSTCLR                          ;RESET DV11
021140  012777  000010  160214          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
021146  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
021152  104415                          ROMCLK                          ;F=-1
021154  012710  050027                  MOV     #S.C+BIT4+BIT2+BIT1+BIT0,(R0)
021160  104415                          ROMCLK                          ;S/C "CLEAR ALU HIGH BYTE"
021162  010210                          MOV     R2,(R0)                 ;XFR RIC_F
021164  104415                          ROMCLK                          ;
021166  011104                          MOV     (R1),R4                 ;READ RIC
021170  012705  000377                  MOV     #377,R5                 ;SET EXPECTED
021174  020504                          CMP     R5,R4                   ;DID S/C WORK?
021176  001401                          BEQ     73$                     ;BR IF YES
021200  104006                          HLT     6                       ;S/C ALU HIGH BYTE FAILED.
021202  104401                  73$:    SCOP1                           ;SW09=1?
021204  104400                          SCOPE                           ;SCOPE TEST


                                        ;*********************** TEST 76 *****************************
                                        ;*MASTER SCANNER TEST.
                                        ;*VERIFY FIRST THAT THE MASTER SCANNER
                                        ;*IS CLEARED BY INIT.
                                        ;*VERIFY SECONDLY THAT THE MASTER SCANNER
                                        ;*CAN BE INCREMENTED FROM 0 THRU 17 BACK TO 0.
                                        ;;************************************************************

```

```
                                ;  TEST 76
                                ;---------------
021206  012737  000076  001226  TST76:  MOV     #76,TSTNO
021214  012737  021334  001216          MOV     #TST77,NEXT
021222  104412                          MSTCLR                          ;RESET DV11
021224  012777  000010  160130          MOV     #BIT3,@DVSCR            ;SET SOURCE SEL
021232  013700  001400                  MOV     DVSFR,R0                ;SET DVSFR POINTER
021236  012702  030126                  MOV     #XFR+BIT6+BIT4+BIT2+BIT1,R2
021242  010210                          MOV     R2,(R0)                 ;XFR "MASTER SCAN 0-3" TO DVRIC
021244  104415                          ROMCLK                          ;
021246  005005                          CLR     R5                      ;SET EXPECTED
021250  017704  160112                  MOV     @DVRIC,R4               ;READ RESULTS
021254  001401                          BEQ     1$                      ;BR IF=0
021256  104006                          HLT     6                       ;MSCAN NOT=0 ON INIT
021260  005205                  1$:     INC     R5                      ;UPDATE POINTER
021262  012703  000025                  MOV     #25,R3                  ;SET COUNT TO 25
021266  012710  050102          2$:     MOV     #S.C+BIT6+BIT1,(R0) ;S/C "ADVANCE MSCAN"
021272  104415                          ROMCLK                          ;CLOCK
021274  012710  050102                  MOV     #S.C+BIT6+BIT1,(R0) ;S/C "ADVANCE MSCAN"
021300  104415                          ROMCLK                          ;CLOCK
021302  010210                          MOV     R2,(R0)                 ;XFR RIC_MSCAN
021304  104415                          ROMCLK                          ;CLOCK
021306  017704  160054                  MOV     @DVRIC,R4               ;READ RESULTS
021312  020504                          CMP     R5,R4                   ;MSCAN INCREMENTED?
021314  001401                          BEQ     3$                      ;BR IF YES
021316  104006                          HLT     6                       ;MSCAN WRONG
021320  005205                  3$:     INC     R5                      ;UPDATE
021322  042705  177760                  BIC     #^C<17>,R5              ;CLEAN
021326  005303                          DEC     R3                      ;COUNT DONE?
021330  001356                          BNE     2$                      ;BR IF NO
021332  104400                          SCOPE                           ;SCOPE

                                        ;*********************** TEST 77 *****************************
                                        ;*BASIC TESTS OF THE "RAM OPERATION" INSTRUCTION.
                                        ;*VERIFY THE READ PORTION OF THE RAM OPERATION.
                                        ;*LOAD ALL SECONDARY REGISTERS OF ALL LINES
                                        ;*WITH DIFFERENT NUMBERS AND VERIFY THAT THE RAM OPERATION
                                        ;*CAN READ THE CORRECT SEC. REG. INTO THE DVRIC REG.
                                        ;;************************************************************

                                ;  TEST 77
                                ;---------------
021334  012737  000077  001226  TST77:  MOV     #77,TSTNO
021342  012737  021560  001216          MOV     #TST100,NEXT
021350  104412                          MSTCLR                          ;RESET DV11
021352  005000                          CLR     R0                      ;
021354  005001                          CLR     R1                      ;
021356  013702  001372                  MOV     DVSRS,R2                ;
021362  013703  001374                  MOV     DVSRSH,R3               ;
021366  013705  001376                  MOV     DVSRA,R5                ;
021372  012704  000001                  MOV     #1,R4                   ;
021376  110012                  1$:     MOVB    R0,(R2)                 ;LOAD LINE NUMBER
021400  110113                          MOVB    R1,(R3)                 ;LOAD SEC. REG. POINTER
021402  010415                          MOV     R4,(R5)                 ;LOAD DATA
021404  122024                          CMPB    (R0)+,(R4)+             ;UPDATE LINE AND DATA
021406  022700  000020                  CMP     #16.,R0                 ;ALL LINES DONE?
```

```
021412  001371                            BNE     1$              ;BR IF NO
021414  005000                            CLR     RØ              ;ZERO LINE POINTER
021416  005201                            INC     R1              ;UPDATE SEC REG POINTER.
021420  022701  000020                    CMP     #16.,R1         ;ALL SEC REG POINTERS DONE?
021424  001364                            BNE     1$              ;BR IF NO
021426  005077  157740            2$:     CLR     @DVSRS          ;ZERO POINTERS
021432  012705  000001                    MOV     #1,R5           ;SET GOOD DATA
021436  012777  000010  157716            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
021444  012703  020000                    MOV     #RAM,R3         ;LOAD RAM INSTR.
021450  013700  001400                    MOV     DVSFR,RØ        ;SET POINTER
021454  012702  030266                    MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1,R2
021460  010310                    3$:     MOV     R3,(RØ)         ;DO RAM READ
021462  104415                            ROMCLK                  ;EXECUTE
021464  010210                            MOV     R2,(RØ)         ;XFR RIC_RAM OUTPUT
021466  104415                            ROMCLK                  ;CLOCK
021470  017704  157672                    MOV     @DVRIC,R4       ;READ RESULT
021474  020504                            CMP     R5,R4           ;GOOD?
021476  001401                            BEQ     4$              ;BR IF YES
021500  104006                            HLT     6               ;RAM READ FAILED.
021502  062705  000020            4$:     ADD     #20,R5          ;UPDATE DATA
021506  005203                            INC     R3              ;UPDATE POINTER
021510  122703  000020                    CMPB    #16.,R3         ;ALL DONE
021514  001361                            BNE     3$              ;BR IF NO
021516  042705  177760                    BIC     #^C<17>,R5      ;CLEAR JUNK
021522  012703  020000                    MOV     #RAM,R3         ;SET RAM INSTR
021526  010046                            MOV     RØ,-(SP)        ;SAVE RØ ON STACK
021530  004537  031342                    PERFORM ,SETSCAN        ;UPDATE MSCANNER
021534  000001                            1                       ;
021536  012600                            MOV     (SP)+,RØ        ;RESTORE RØ
021540  005205                            INC     R5              ;UPDATE DATA
021542  012710  030124                    MOV     #XFR+BIT6+BIT4+BIT2,(RØ) ;XFR RAR_MSCAN Ø-3
021546  104415                            ROMCLK                  ;EXECUTE
021550  022705  000020                    CMP     #16.,R5         ;ALL DONE?
021554  001341                            BNE     3$              ;BR IF NO
021556  104400                            SCOPE                   ;SCOPE TEST.

                                          ;*********************** TEST 100 ******************************
                                          ;*TEST OF BRANCH A TEST POINTS
                                          ;*THAT WERE PREVIOUSLY SKIPPED BECAUSE
                                          ;OF THE SIGNALS NEEDED TO TEST THE
                                          ;*FUNCTIONS:
                                          ;*BIT11 BIT1Ø   BITØ9   BITØ8   FUNCTION
                                          ;*  Ø     Ø       Ø       Ø      ALU 15=1,Ø
                                          ;*  1     Ø       Ø       Ø      ALU 13-  =1,Ø
                                          ;*  1     Ø       Ø       Ø            -12=1,Ø
                                          ;*  1     Ø       1       Ø      ALU ØØ=1,Ø
                                          ;*  1     Ø       1       1      ALU Ø1=1,Ø
                                          ;*  1     1       Ø       Ø      ALU Ø2=1,Ø
                                          ;*  1     1       Ø       1      ALU Ø3=1,Ø
                                          ;*  1     1       1       Ø      ALU Ø4=1,Ø
                                          ;****************************************************************


                                  ;  TEST 100
                                  ;--------------
021560  012737  000100  001226    TST100: MOV     #100,TSTNO
021566  012737  023162  001216            MOV     #TST101,NEXT
```

```
021574  005077  157572                    CLR     @DVSRS
                                          ;*BRANCH "A" TEST OF ALU 15
                                          ;*
021600  104412                            MSTCLR                  ;RESET DV11
021602  012777  000010  157552            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
021610  012777  100000  157560            MOV     #BIT15,@DVSRA   ;LOAD DATA
021616  012777  020000  157554            MOV     #RAM,@DVSFR     ;DO A "RAM READ"
021624  104415                            ROMCLK                  ;EXECUTE
021626  012777  030261  157544            MOV     #XFR+BIT7+BIT5+BIT4+BITØ,@DVSFR
021634  104415                            ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
021636  012777  010037  157534            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BITØ,@DVSFR
021644  104415                            ROMCLK                  ;F=A
021646  012777  000000  157524            MOV     #ØØØØ,@DVSFR    ;LOAD BRANCH POINT
021654  017704  157510                    MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
021660  042704  177774                    BIC     #^C<BIT1+BITØ>,R4 ;CLEAR JUNK
021664  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED
021670  020504                            CMP     R5,R4           ;BR POINTS CORRECT?
021672  001401                            BEQ     64$             ;BR IF YES
021674  104006                            HLT     6               ;BRANCH POINTS WRONG
021676  104412                    64$:    MSTCLR
021700  012777  000010  157454            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
021706  012777  000000  157464            MOV     #ØØØØ,@DVSFR    ;RESET DV11
021714  017704  157450                    MOV     @DVLCR,R4       ;LOAD BRANCH, POINT TEST
021720  042704  177774                    BIC     #^C<BIT1+BITØ>,R4 ;READ BR POINTS
021724  012705  000003                    MOV     #BIT1+BITØ,R5   ;CLEAR JUNK
021730  020504                            CMP     R5,R4           ;SET EXPECTED
021732  001401                            BEQ     65$             ;BR POINT OK?
021734  104006                            HLT     6               ;BR IF YES
021736                            65$:                            ;BR POINTS WRONG
                                          ;*BRANCH "A" TEST OF ALU 13-
                                          ;*
021736  104412                            MSTCLR                  ;RESET DV11
021740  012777  000010  157414            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
021746  012777  020000  157422            MOV     #BIT13,@DVSRA   ;LOAD DATA
021754  012777  020000  157416            MOV     #RAM,@DVSFR     ;DO A "RAM READ"
021762  104415                            ROMCLK                  ;EXECUTE
021764  012777  030261  157406            MOV     #XFR+BIT7+BIT5+BIT4+BITØ,@DVSFR
021772  104415                            ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
021774  012777  010037  157376            MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BITØ,@DVSFR
022002  104415                            ROMCLK                  ;F=A
022004  012777  004000  157366            MOV     #BIT11,@DVSFR   ;LOAD BRANCH POINT
022012  017704  157352                    MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022016  042704  177774                    BIC     #^C<BIT1+BITØ>,R4 ;CLEAR JUNK
022022  012705  000002                    MOV     #BIT1,R5        ;SET EXPECTED
022026  020504                            CMP     R5,R4           ;BR POINTS CORRECT?
022030  001401                            BEQ     66$             ;BR IF YES
022032  104006                            HLT     6               ;BRANCH POINTS WRONG
022034  104412                    66$:    MSTCLR
022036  012777  000010  157316            MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022044  012777  004000  157326            MOV     #BIT11,@DVSFR   ;RESET DV11
022052  017704  157312                    MOV     @DVLCR,R4       ;LOAD BRANCH, POINT TEST
022056  042704  177774                    BIC     #^C<BIT1+BITØ>,R4 ;READ BR POINTS
022062  012705  000003                    MOV     #BIT1+BITØ,R5   ;CLEAR JUNK
022066  020504                            CMP     R5,R4           ;SET EXPECTED
022070  001401                            BEQ     67$             ;BR POINT OK?
022072  104006                            HLT     6               ;BR IF YES
```

```
022074                            67$:                            ;BR POINTS WRONG
                                        ;*BRANCH "A" TEST OF ALU  -12
                                        ;*
022074  104412                          MSTCLR                    ;RESET DV11
022076  012777  000010  157256          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL
022104  012777  010000  157264          MOV     #BIT12,@DVSRA     ;LOAD DATA
022112  012777  020000  157260          MOV     #RAM,@DVSFR       ;DO A "RAM READ"
022120  104415                          ROMCLK                    ;EXECUTE
022122  012777  030261  157250          MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022130  104415                          ROMCLK                    ;XFR RAM OUTPUT TO "A" REG
022132  012777  010037  157240          MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022140  104415                          ROMCLK                    ;F=A
022142  012777  004000  157230          MOV     #BIT11,@DVSFR     ;LOAD BRANCH POINT
022150  017704  157214                  MOV     @DVLCR,R4         ;READ BRANCH TEST POINT
022154  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022160  012705  000002                  MOV     #BIT1,R5          ;SET EXPECTED
022164  020504                          CMP     R5,R4             ;BR POINTS CORRECT?
022166  001401                          BEQ     68$               ;BR IF YES
022170  104006                          HLT     6                 ;BRANCH POINTS WRONG
022172  104412                    68$:  MSTCLR
022174  012777  000010  157160          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL,
022202  012777  004000  157170          MOV     #BIT11,@DVSFR     ;RESET DV11
022210  017704  157154                  MOV     @DVLCR,R4         ;LOAD BRANCH, POINT TEST
022214  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022220  012705  000003                  MOV     #BIT1+BIT0,R5     ;CLEAR JUNK
022224  020504                          CMP     R5,R4             ;SET EXPECTED
022226  001401                          BEQ     69$               ;BR POINT OK?
022230  104006                          HLT     6                 ;BR IF YES
022232                            69$:                            ;BR POINTS WRONG
                                        ;*BRANCH "A" TEST OF ALU 00
                                        ;*
022232  104412                          MSTCLR                    ;RESET DV11
022234  012777  000010  157120          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL
022242  012777  000001  157126          MOV     #BIT0,@DVSRA      ;LOAD DATA
022250  012777  020000  157122          MOV     #RAM,@DVSFR       ;DO A "RAM READ"
022256  104415                          ROMCLK                    ;EXECUTE
022260  012777  030261  157112          MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022266  104415                          ROMCLK                    ;XFR RAM OUTPUT TO "A" REG
022270  012777  010037  157102          MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022276  104415                          ROMCLK                    ;F=A
022300  012777  005000  157072          MOV     #BIT11+BIT9,@DVSFR      ;LOAD BRANCH POINT
022306  017704  157056                  MOV     @DVLCR,R4         ;READ BRANCH TEST POINT
022312  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022316  012705  000002                  MOV     #BIT1,R5          ;SET EXPECTED
022322  020504                          CMP     R5,R4             ;BR POINTS CORRECT?
022324  001401                          BEQ     70$               ;BR IF YES
022326  104006                          HLT     6                 ;BRANCH POINTS WRONG
022330  104412                    70$:  MSTCLR
022332  012777  000010  157022          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL,
022340  012777  005000  157032          MOV     #BIT11+BIT9,@DVSFR      ;RESET DV11
022346  017704  157016                  MOV     @DVLCR,R4         ;LOAD BRANCH, POINT TEST
022352  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022356  012705  000003                  MOV     #BIT1+BIT0,R5     ;CLEAR JUNK
022362  020504                          CMP     R5,R4             ;SET EXPECTED
022364  001401                          BEQ     71$               ;BR POINT OK?
022366  104006                          HLT     6                 ;BR IF YES
```

```
022370                            71$:                            ;BR POINTS WRONG
                                        ;*BRANCH "A" TEST OF ALU 01
                                        ;*
022370  104412                          MSTCLR                    ;RESET DV11
022372  012777  000010  156762          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL
022400  012777  000002  156770          MOV     #BIT1,@DVSRA      ;LOAD DATA
022406  012777  020000  156764          MOV     #RAM,@DVSFR       ;DO A "RAM READ"
022414  104415                          ROMCLK                    ;EXECUTE
022416  012777  030261  156754          MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022424  104415                          ROMCLK                    ;XFR RAM OUTPUT TO "A" REG
022426  012777  010037  156744          MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022434  104415                          ROMCLK                    ;F=A
022436  012777  005400  156734          MOV     #BIT11+BIT9+BIT8,@DVSFR ;LOAD BRANCH POINT
022444  017704  156720                  MOV     @DVLCR,R4         ;READ BRANCH TEST POINT
022450  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022454  012705  000002                  MOV     #BIT1,R5          ;SET EXPECTED
022460  020504                          CMP     R5,R4             ;BR POINTS CORRECT?
022462  001401                          BEQ     72$               ;BR IF YES
022464  104006                          HLT     6                 ;BRANCH POINTS WRONG
022466  104412                    72$:  MSTCLR
022470  012777  000010  156664          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL,
022476  012777  005400  156674          MOV     #BIT11+BIT9+BIT8,@DVSFR ;RESET DV11
022504  017704  156660                  MOV     @DVLCR,R4         ;LOAD BRANCH, POINT TEST
022510  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022514  012705  000003                  MOV     #BIT1+BIT0,R5     ;CLEAR JUNK
022520  020504                          CMP     R5,R4             ;SET EXPECTED
022522  001401                          BEQ     73$               ;BR POINT OK?
022524  104006                          HLT     6                 ;BR IF YES
022526                            73$:                            ;BR POINTS WRONG
                                        ;*BRANCH "A" TEST OF ALU 02
                                        ;*
022526  104412                          MSTCLR                    ;RESET DV11
022530  012777  000010  156624          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL
022536  012777  000004  156632          MOV     #BIT2,@DVSRA      ;LOAD DATA
022544  012777  020000  156626          MOV     #RAM,@DVSFR       ;DO A "RAM READ"
022552  104415                          ROMCLK                    ;EXECUTE
022554  012777  030261  156616          MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022562  104415                          ROMCLK                    ;XFR RAM OUTPUT TO "A" REG
022564  012777  010037  156606          MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022572  104415                          ROMCLK                    ;F=A
022574  012777  006000  156576          MOV     #BIT11+BIT10,@DVSFR     ;LOAD BRANCH POINT
022602  017704  156562                  MOV     @DVLCR,R4         ;READ BRANCH TEST POINT
022606  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022612  012705  000002                  MOV     #BIT1,R5          ;SET EXPECTED
022616  020504                          CMP     R5,R4             ;BR POINTS CORRECT?
022620  001401                          BEQ     74$               ;BR IF YES
022622  104006                          HLT     6                 ;BRANCH POINTS WRONG
022624  104412                    74$:  MSTCLR
022626  012777  000010  156526          MOV     #BIT3,@DVSCR      ;SET SOURCE SEL,
022634  012777  006000  156536          MOV     #BIT11+BIT10,@DVSFR     ;RESET DV11
022642  017704  156522                  MOV     @DVLCR,R4         ;LOAD BRANCH, POINT TEST
022646  042704  177774                  BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
022652  012705  000003                  MOV     #BIT1+BIT0,R5     ;CLEAR JUNK
022656  020504                          CMP     R5,R4             ;SET EXPECTED
022660  001401                          BEQ     75$               ;BR POINT OK?
022662  104006                          HLT     6                 ;BR IF YES
```

```
022664                              75$:                            ;BR POINTS WRONG
                                             ;*BRANCH "A" TEST OF ALU 03
                                             ;*
022664  104412                               MSTCLR                  ;RESET DV11
022666  012777  000010  156466               MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
022674  012777  000010  156474               MOV     #BIT3,@DVSRA    ;LOAD DATA
022702  012777  020000  156470               MOV     #RAM,@DVSFR     ;DO A "RAM READ"
022710  104415                               ROMCLK                  ;EXECUTE
022712  012777  030261  156460               MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
022720  104415                               ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
022722  012777  010037  156450               MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
022730  104415                               ROMCLK                  ;F=A
022732  012777  006400  156440               MOV     #BIT11+BIT10+BIT8,@DVSFR        ;LOAD BRANCH POINT
022740  017704  156424                       MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
022744  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
022750  012705  000002                       MOV     #BIT1,R5        ;SET EXPECTED
022754  020504                               CMP     R5,R4           ;BR POINTS CORRECT?
022756  001401                               BEQ     76$             ;BR IF YES
022760  104006                               HLT     6               ;BRANCH POINTS WRONG
022762  104412                      76$:     MSTCLR
022764  012777  000010  156370               MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
022772  012777  006400  156400               MOV     #BIT11+BIT10+BIT8,@DVSFR        ;RESET DV11
023000  017704  156364                       MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
023004  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
023010  012705  000003                       MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
023014  020504                               CMP     R5,R4           ;SET EXPECTED
023016  001401                               BEQ     77$             ;BR POINT OK?
023020  104006                               HLT     6               ;BR IF YES
023022                              77$:                            ;BR POINTS WRONG
                                             ;*BRANCH "A" TEST OF ALU 04
                                             ;*
023022  104412                               MSTCLR                  ;RESET DV11
023024  012777  000010  156330               MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
023032  012777  000020  156336               MOV     #BIT4,@DVSRA    ;LOAD DATA
023040  012777  020000  156332               MOV     #RAM,@DVSFR     ;DO A "RAM READ"
023046  104415                               ROMCLK                  ;EXECUTE
023050  012777  030261  156322               MOV     #XFR+BIT7+BIT5+BIT4+BIT0,@DVSFR
023056  104415                               ROMCLK                  ;XFR RAM OUTPUT TO "A" REG
023060  012777  010037  156312               MOV     #ALU+BIT4+BIT3+BIT2+BIT1+BIT0,@DVSFR
023066  104415                               ROMCLK                  ;F=A
023070  012777  007000  156302               MOV     #BIT11+BIT10+BIT9,@DVSFR        ;LOAD BRANCH POINT
023076  017704  156266                       MOV     @DVLCR,R4       ;READ BRANCH TEST POINT
023102  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023106  012705  000002                       MOV     #BIT1,R5        ;SET EXPECTED
023112  020504                               CMP     R5,R4           ;BR POINTS CORRECT?
023114  001401                               BEQ     78$             ;BR IF YES
023116  104006                               HLT     6               ;BRANCH POINTS WRONG
023120  104412                      78$:     MSTCLR
023122  012777  000010  156232               MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
023130  012777  007000  156242               MOV     #BIT11+BIT10+BIT9,@DVSFR        ;RESET DV11
023136  017704  156226                       MOV     @DVLCR,R4       ;LOAD BRANCH. POINT TEST
023142  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;READ BR POINTS
023146  012705  000003                       MOV     #BIT1+BIT0,R5   ;CLEAR JUNK
023152  020504                               CMP     R5,R4           ;SET EXPECTED
023154  001401                               BEQ     79$             ;BR POINT OK?
023156  104006                               HLT     6               ;BR IF YES
```

```
023160                              79$:                            ;BR POINTS WRONG
023160  104400                               SCOPE


                                             ;************************ TEST 101 ********************************
                                             ;*TEST OF BRANCH "B" "RAM OUTPUT 0-14=0".
                                             ;*TEST TO A RAM READ AND "FLOAT" A "1" FROM
                                             ;*RAM 0 TO 14 ; EXPECTING "RAM 014=0" TO BE FALSE.
                                             ;*THEN THE "1" IS SHIFTED INTO BIT15 AND
                                             ;*"RAM 0-14=0" SHOULD BE FALSE.
                                             ;*THIS ALSO TEST "BRB" [RAM OUTPUT BIT15] TRUE.
                                             ;;*****************************************************************

                                    ;  TEST 101
                                    ;--------------
023162  012737  000101  001226      TST101: MOV     #101,TSTNO
023170  012737  023362  001216               MOV     #TST102,NEXT
023176  104412                               MSTCLR                  ;RESET DV11
023200  012703  000001                       MOV     #1,R3           ;SET DATA
023204  012702  076000                       MOV     #BRB+BIT11+BIT10,R2 ;BRB "RAM OUTPUT 0-14=0"?
023210  012777  000010  156144               MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
023216  010277  156156                       MOV     R2,@DVSFR       ;LOAD BRB TEST
023222  017704  156142                       MOV     @DVLCR,R4       ;READ TEST POINTS
023226  012705  000001                       MOV     #BIT0,R5        ;SET EXPECTED
023232  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023236  020504                               CMP     R5,R4           ;TRUE?
023240  001401                               BEQ     .+4             ;BR IF YES
023242  104006                               HLT     6               ;RAM OUTPUT 0-14 NOT TRUE AFTER INIT
023244  012705  000003                       MOV     #BIT1+BIT0,R5   ;SET EXPECTED
023250  010377  156122              1$:      MOV     R3,@DVSRA       ;LOAD DATA
023254  012777  020000  156116               MOV     #RAM,@DVSFR     ;ISSUE RAM READ INSTR
023262  104415                               ROMCLK                  ;
023264  010277  156110                       MOV     R2,@DVSFR       ;BRB TEST
023270  017704  156074                       MOV     @DVLCR,R4       ;READ BR TEST POINTS
023274  042704  177774                       BIC     #^C<BIT1+BIT0>,R4 ;CLEAR JUNK
023300  005703                               TST     R3              ;IS 15=1 IN DATA?
023302  100002                               BPL     2$              ;BR IF NO
023304  042705  000002                       BIC     #BIT1,R5        ;IF 15=1 - 0-14=TRUE
023310  020504                      2$:      CMP     R5,R4           ;BRB TEST POINT OK?
023312  001401                               BEQ     3$              ;BR IF YES
023314  104006                               HLT     6               ;BAD TEST POINT
023316  000241                      3$:      CLC                     ;CLEAR CARRY
023320  006103                               ROL     R3              ;MOV BIT TO NEXT RAM POSITION
023322  001352                               BNE     1$              ;HAVE ALL BITS BEEN TESTED?
023324  012705  000001                       MOV     #BIT0,R5        ;SET EXPECTED RESULTS
023330  012777  075400  156042               MOV     #BRB+BIT11+BIT9+BIT8,@DVSFR
023336  017704  156026                       MOV     @DVLCR,R4       ;BRB "RAM OUTPUT 15H"
023342  042704  177774                       BIC     #^C<BIT1+BIT0>,R4
023346  020504                               CMP     R5,R4           ;BRANCH RESULTS OK?
023350  001403                               BEQ     4$              ;BR IF YES
023352  017702  156022                       MOV     @DVSFR,R2       ;SAVE DVSFR FOR TYPEOUT
023356  104006                               HLT     6               ;"RAM OUTPUT 15H" S/B TRUE
023360  104400                      4$:      SCOPE                   ;SCOPE TESTS



                                             ;************************ TEST 102 ********************************
                                             ;*TEST OF THE RAM WRITE OPERATION.
                                             ;*WRITE ALL SECONDARY REGISTERS FOR ALL LINES
                                             ;*WITH DIFFERENT DATA BY USING THE ROM
                                             ;*AND VERIFY THE DATA BY THE UNIBUS.
                                             ;!*************************************************************

                                     ;  TEST 102
                                     ;---------------
023362  012737  000102  001226      TST102: MOV     #102,TSTNO
023370  012737  023776  001216              MOV     #TST103,NEXT
023376  104412                              MSTCLR                  ;CLEAR ALL DV11 REGISTERS
023400  012777  000010  155754              MOV     #BIT3,@DVSCR    ;SET SORCE SEC
023406  013700  001400                      MOV     DVSFR,R0        ;SET DVSFR POINTER IN R0
023412  012702  010077                      MOV     #ALU+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0,R2
                                                                    ;FUNCTION "F=A+1
023416  012703  020760                      MOV     #RAM+BIT8+BIT7+BIT6+BIT5+BIT4,R3
                                                                    ;RAM WRITE FROM ALU RESULT.
023422  012704  030361                      MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,R4
                                                                    ;MOVE ALU RESULT TO "A" REG.
023426  005005                              CLR     R5              ;CLEAR LINE NUMBER COUNTER
023430  005001                      1$:     CLR     R1              ;ZERO SEC REG POINTER
023432  110577  155734              2$:     MOVB    R5,@DVSRS       ;LOAD LINE
023436  110177  155732                      MOVB    R1,@DVSRSH      ;LOAD SEC REG.
023442  012777  177777  155726              MOV     #-1,@DVSRA      ;SET "FOOT PRINT"
023450  042703  000017                      BIC     #17,R3          ;CLEAR LINER
023454  050103                              BIS     R1,R3           ;SET LINE
023456  010310                              MOV     R3,(R0)         ;DO "RAM WRITE"
023460  104415                              ROMCLK                  ;
023462  012777  077000  155710              MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023470  010546                              MOV     R5,-(SP)        ;SAVERS
023472  010446                              MOV     R4,-(SP)        ;SAVE R4
023474  010246                              MOV     R2,-(SP)        ;SAVE R2
023476  012705  000001                      MOV     #BIT0,R5        ;EXPECTED
023502  017704  155662                      MOV     @DVLCR,R4       ;READ BR. RESULT
023506  042704  177774                      BIC     #^C<BIT1+BIT0>,R4 ;STRIP JUNK
023512  020504                              CMP     R5,R4           ;WRITE INHIBIT TRUE?
023514  001401                              BEQ     .+4             ;
023516  104006                              HLT     6               ;WRITE INHIBIT FAILED
023520  022777  177777  155650              CMP     #-1,@DVSRA      ;WAS WRITE
023526  001401                              BEQ     .+4             ;REALLY
023530  104000                              HLT     0               ;INHIBTED?
023532  012602                              MOV     (SP)+,R2        ;RESTORE
023534  012604                              MOV     (SP)+,R4        ;REGISTERS
023536  012605                              MOV     (SP)+,R5        ;
023540  012710  020000                      MOV     #RAM,(R0)       ;PLACE RAM INSTR IN SFR
023544  050110                              BIS     R1,(R0)         ;PLACE SEC REG POINTER IN SFR
023546  104415                              ROMCLK                  ;EXECUTE INSTR.
023550  010210                              MOV     R2,(R0)         ;"F=A+1"
023552  104415                              ROMCLK                  ;EXECUTE
023554  042703  000017                      BIC     #17,R3          ;CLEAR SEC REG POINTER
023560  050103                              BIS     R1,R3           ;SET SEC REG POINTER
023562  010310                              MOV     R3,(R0)         ;RAM WRITE FROM ALU RESULT
023564  104415                              ROMCLK                  ;EXECUTE
023566  012777  077000  155604              MOV     #BRB+BIT11+BIT10+BIT9,@DVSFR
023574  010546                              MOV     R5,-(SP)        ;TEST WRITE
023576  010446                              MOV     R4,-(SP)        ;INHIBIT
```

```
023600  010246                              MOV     R2,-(SP)        ;FALSE!
023602  017702  155572                      MOV     @DVSFR,R2
023606  012705  000003                      MOV     #BIT1+BIT0,R5   ;EXPECTED
023612  017704  155552                      MOV     @DVLCR,R4       ;READ RESULTS
023616  042704  177774                      BIC     #^C<BIT1+BIT0>,R4 ;STRIP JUNK
023622  020504                              CMP     R5,R4           ;GOOD?
023624  001401                              BEQ     .+4             ;
023626  104006                              HLT     6               ;WRITE INHIBIT S/B FALSE
023630  012602                              MOV     (SP)+,R2        ;RESTORE
023632  012604                              MOV     (SP)+,R4        ;REGISTERS
023634  012605                              MOV     (SP)+,R5        ;
023636  010410                              MOV     R4,(R0)         ;MOVE ALU RESULT TO A REG (UPDATED DATA)
023640  104415                              ROMCLK                  ;EXECUTE
023642  005201                              INC     R1              ;UPDATE SEC REG POINTER
023644  022701  000020                      CMP     #16.,R1         ;ALL REGISTERS DONE?
023650  001270                              BNE     2$              ;BR IF NO
023652  010046                              MOV     R0,-(SP)        ;SAVE R0
023654  004537  031342                      PERFORM ,SETSCAN        ;UPDATE MSCAN
023660  000001                              1                       ;(LINE NO. UPDATE)
023662  012600                              MOV     (SP)+,R0        ;RESTORE R0
023664  012710  030124                      MOV     #XFR+BIT6+BIT4+BIT2,(R0)
023670  104415                              ROMCLK                  ;XFR MSCAN TO RAR 0-3 CLOCK
023672  005205                              INC     R5              ;UPDATE LINE NUMBER COUNTER.
023674  022705  000020                      CMP     #16.,R5         ;ALL LINES DONE?
023700  001253                              BNE     1$              ;BR IF NO
023702  005000                      3$:     CLR     R0              ;CLEAR SEC REG POINTER
023704  005037  001246                      CLR     TEMP1           ;CLEAR LINE NUMBER POINTER
023710  013702  001372                      MOV     DVSRS,R2        ;SET SRC POINTER (LINE SEL)
023714  013703  001374                      MOV     DVSRSH,R3       ;SET HIGH BYTE POINTER (SEC REG SEL)
023720  013701  001376                      MOV     DVSRA,R1        ;SET SRA POINTER (ACCESS REG)
023724  012705  000001                      MOV     #1,R5           ;SET EXPECTED DATA
023730  110013                      4$:     MOVB    R0,(R3)         ;LOAD SEC REG SEL
023732  113712  001246                      MOVB    TEMP1,(R2)      ;LOAD LINE NO.
023736  011104                              MOV     (R1),R4         ;READ RAM RESULT
023740  020504                              CMP     R5,R4           ;WAS RAM "WRITTEN" OCCRECTLY?
023742  001401                              BEQ     5$              ;BR IF RAM DATA OK
023744  104006                              HLT     6               ;RAM WAS "WRITTEN" INCORRECTLY
023746  122025                      5$:     CMPB    (R0)+,(R5)+     ;UPDATE SEC REG POINTER AND DATA EXPECTED
023750  022700  000020                      CMP     #16.,R0         ;ALL SEC REGISTERS DONE?
023754  001365                              BNE     4$              ;BR IF NO
023756  005000                              CLR     R0              ;ZERO SEC REG POINTER
023760  005237  001246                      INC     TEMP1           ;UPDATE LINE NUMBER POINTER
023764  023727  001246  000020              CMP     TEMP1,#16.      ;ALL LINES DONE?
023772  001356                              BNE     4$              ;BR IF NO
023774  104400                              SCOPE                   ;SCOPE TEST

                                             ;*********************** TEST 103 *****************************
                                             ;*BASIC TEST FOR THE "BCC OPERATION"
                                             ;*POLYNOMIAL SELECTION TABLE:
                                             ;*RAM OUTPUT BIT04 BIT03        POLY
                                             ;*            0     0           LRC 8
                                             ;*            0     1           CRC 16
                                             ;*            1     1           CRC CCITT
                                             ;!*************************************************************


                                             ;*********************** TEST 103 *****************************
```

```
                                        ;*TEST OF LRC 8.
                                        ;*PATTERNS ARE:
                                        ;*A REG  B REG  BCC (EXPECTED)
                                        ;*  0'S    0'S   0'S
                                        ;*  0'S    1'S   1'S
                                        ;*  1'S    0'S   1'S
                                        ;*  1'S    1'S   0'S
                                        ;;*****************************************************************

                                        ;  TEST 103
                                        ;--------------
 023776  012737  000103  001226  TST103: MOV     #103,TSTNO
 024004  012737  024252  001216          MOV     #TST104,NEXT
 024012  104412                          MSTCLR                  ;RESET DV11
 024014  012777  000010  155340          MOV     #BIT3,@DVSCR    ;SET SORCE SEL
 024022  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
 024026  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
 024032  012710  060000                  MOV     #BCC,(R0)       ;DO A "BCC" INSTRUCTION USING "LRC8"
 024036  104415                          ROMCLK                  ;EXECUTE
 024040  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
 024042  104415                          ROMCLK                  ;EXECUTE
 024044  005005                          CLR     R5              ;SET EXPECTED TO 0
 024046  017704  155314                  MOV     @DVRIC,R4       ;READ RESULTS OF BCC REG.
 024052  001401                          BEQ     1$              ;BR IF =0
 024054  104006                          HLT     6               ;BCC REG NOT =0
 024056  012710  010034          1$:     MOV     #ALU+BIT4+BIT3+BIT2,(R0)
 024062  104415                          ROMCLK                  ;ALU OPR "F=-1"
 024064  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
 024070  104415                          ROMCLK                  ;XFR ALU RESULT TO B REGISTER
 024072  012710  060000                  MOV     #BCC,(R0)       ;DO A "BCC" OPR
 024076  104415                          ROMCLK                  ;EXECUTE
 024100  010210                          MOV     R2,(R0)         ;XFR BCC REG TO DVRIC
 024102  104415                          ROMCLK                  ;EXECUTE
 024104  012705  000377                  MOV     #377,R5         ;SET EXPECTED RESULTS =ALL 1'S (LOW BYTE)
 024110  017704  155252                  MOV     @DVRIC,R4       ;READ RESULTS
 024114  020504                          CMP     R5,R4           ;DID BCC OPR WORK
 024116  001401                          BEQ     2$              ;BR IF OK
 024120  104006                          HLT     6               ;EITHER "BCC" OPR FAILED OR BIT(S) DROPPED IN LOW BYTE O
 024122  104412                  2$:     MSTCLR                  ;RESET INTERAL REG (BCC,ALU,B)
 024124  012777  000010  155230          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
 024132  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
 024136  104415                          ROMCLK                  ;ALU "F=-1"
 024140  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
 024144  104415                          ROMCLK                  ;XFR ALU RESULT TO "A" REGISTER
 024146  012710  060000                  MOV     #BCC,(R0)       ;BCC OPERATION
 024152  104415                          ROMCLK                  ;EXECUTE
 024154  010210                          MOV     R2,(R0)         ;XFR BCC TO DVRIC
 024156  104415                          ROMCLK                  ;EXECUTE
 024160  017704  155202                  MOV     @DVRIC,R4       ;READ RESULTS
 024164  020504                          CMP     R5,R4           ;BCC REG S/B =377
 024166  001401                          BEQ     3$              ;BR IF OK
 024170  104006                          HLT     6               ;BCC REG NOT =377
                                                                 ;READ OF DATA FROM "A" REG LEG FAILED
 024172  104412                  3$:     MSTCLR                  ;CLEAR INTERNAL REGISTERS
 024174  012777  000010  155160          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 024202  012710  010034                  MOV     #ALU+BIT4+BIT3+BIT2,(R0)
```

```
 024206  104415                          ROMCLK                  ;ALU "F=-1"
 024210  012710  030361                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT0,(R0)
 024214  104415                          ROMCLK                  ;XFR ALU RESULT TO "A" REG
 024216  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
 024222  104415                          ROMCLK                  ;XFR ALU RESULT TO "B" REG
 024224  012710  060000                  MOV     #BCC,(R0)       ;SO BCC OPR
 024230  104415                          ROMCLK                  ;
 024232  010210                          MOV     R2,(R0)         ;XFR BCC TO DVRIC
 024234  104415                          ROMCLK                  ;ROMCLK
 024236  005005                          CLR     R5              ;EXPECT 0'S
 024240  017704  155122                  MOV     @DVRIC,R4       ;READ BCC RESULT
 024244  001401                          BEQ     4$              ;BR IF =0
 024246  104006                          HLT     6               ;BCC "LRC" XOR TEST FAILED
 024250  104400                  4$:     SCOPE                   ;SCOPE TEST

                                        ;*********************** TEST 104 ****************************
                                        ;*TEST OF POLYNOMIAL "CRC 16"
                                        ;*TEST THAT BITS 9-13 OF THE "B" REG APPEAR
                                        ;*IN BITS 1-5 OF THE BCC REG.
                                        ;;*****************************************************************

                                        ;  TEST 104
                                        ;--------------
 024252  012737  000104  001226  TST104: MOV     #104,TSTNO
 024260  012737  024444  001216          MOV     #TST105,NEXT
 024266  104412                          MSTCLR                  ;RESET DV11
 024270  012777  000010  155064          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
 024276  013700  001400                  MOV     DVSFR,R0        ;SET DVSFR POINTER
 024302  012777  001000  155066          MOV     #BIT9,@DVSRA    ;LOAD "DATA"
 024310  012710  020000                  MOV     #RAM,(R0)       ;RAM READ
 024314  104415                          ROMCLK                  ;EXECUTE
 024316  012710  030261                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
 024322  104415                          ROMCLK                  ;XFR RAM OUTPUT TO "A" REG.
 024324  012710  030262                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
 024330  104415                          ROMCLK                  ;XFR RAM OUTPUT TO "B" REG.
 024332  012777  000010  155036          MOV     #BIT3,@DVSRA    ;SELECT POLYNOMIAL "CRC16"
 024340  012710  020000                  MOV     #RAM,(R0)       ;READ DATA
 024344  104415                          ROMCLK                  ;
 024346  012705  001000          1$:     MOV     #BIT9,R5        ;SET EXPECTED RESULTS
 024352  012710  060000          2$:     MOV     #BCC,(R0)       ;BCC OPR "CRC16"
 024356  104415                          ROMCLK                  ;EXECUTE
 024360  012710  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,(R0)
 024364  104415                          ROMCLK                  ;XFR BCC REG TO DVRIC
 024366  017704  154774                  MOV     @DVRIC,R4       ;READ RESULTS
 024372  012703  000010                  MOV     #8.,R3          ;PREPARE
 024376  000241                  3$:     CLC                     ;TO
 024400  006104                          ROL     R4              ;POSITION
 024402  005303                          DEC     R3              ;RESULT
 024404  001374                          BNE     3$              ;OF BCC OPERATION
 024406  020504                          CMP     R5,R4           ;DID CRC16 WORK?
 024410  001401                          BEQ     4$              ;BR IF YES
 024412  104006                          HLT     6               ;INCORRECT BCC RESULTS
 024414  012710  010026          4$:     MOV     #ALU+BIT4+BIT2+BIT1,(R0)
 024420  104415                          ROMCLK                  ;ALU "F=A+B"
 024422  012710  030362                  MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT1,(R0)
 024426  104415                          ROMCLK                  ;XFR ALU RESULT TO "B" REG
```

```
024430  062705  001000                  ADD     #BIT9,R5        ;UPDATE DATA COMPARE
024434  032705  040000                  BIT     #BIT14,R5       ;ALL DATA DONE?
024440  001744                          BEQ     2$              ;BR IF NO
024442  104400                          SCOPE                   ;SCOPE TEST



                                        ;*********************** TEST 105 ****************************
                                        ;*TEST OF THE BCC OPERATION USING
                                        ;*USING CRC16 FOR THE POLYNOMIAL
                                        ;*SPECIFIC DATA PATTERNS ARE USED TO
                                        ;*ISOLATE FAULTS AS SOON AS POSSIBLE
                                        ;;*************************************************************

                                ;  TEST 105
                                ;--------------
024444  012737  000105  001226  TST105: MOV     #105,TSTNO
024452  012737  026100  001216          MOV     #TST106,NEXT
024460  012737  120001  031646          MOV     #CRC16,XPOLY    ;SET POLYNOMIAL
024466  013700  001400                  MOV     DVSFR,R0        ;SET REGISTER POINTER
024472  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
                                                                ;SET FOR XFER FROM BCC TO DVRIC
024476  012737  024504  001220          MOV     #1$,LOCK        ;SET IF SW09=1
024504  104412                  1$:     MSTCLR                  ;ISSUE A MSTCLR
024506  012777  000010  154646          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
024514  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
024520  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
024524  000010                          8.                      ;SHIFTS PERFORMED
024526  000000                          0                       ;DATA CHAR
024530  000000                          0                       ;PREVIOUS BCC
024532  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
024536  000000                          0                       ;DATA TO THE A REGISTER
024540  000000                          0                       ;DATA TO THE B REGISTER
024542  000010                          10                      ;POLY SELT. (RAMOUTPUT)
024544  012710  060000                  MOV     #BCC,(R0)
024550  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
024552  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
024554  104415                          ROMCLK                  ;
024556  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
024562  017704  154600                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
024566  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
024570  001401                          BEQ     64$             ;BR IF OK!
024572  104006                          HLT     6               ;BCC CALCULATION ERROR
024574  104401                  64$:    SCOP1                   ;SW09=1?
024576  012737  024604  001220          MOV     #2$,LOCK        ;SET IF SW09=1
024604  104412                  2$:     MSTCLR                  ;ISSUE A MSTCLR
024606  012777  000010  154546          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
024614  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
024620  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
024624  000010                          8.                      ;SHIFTS PERFORMED
024626  000252                          ^B<10101010>                    ;DATA CHAR
024630  040000                          BIT14                   ;PREVIOUS BCC
024632  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
024636  000252                          ^B<10101010>                    ;DATA TO THE A REGISTER
024640  040000                          BIT14                   ;DATA TO THE B REGISTER
024642  000010                          10                      ;POLY SELT. (RAMOUTPUT)
```

```
024644  012710  060000                  MOV     #BCC,(R0)
024650  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
024652  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
024654  104415                          ROMCLK                  ;
024656  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
024662  017704  154500                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
024666  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
024670  001401                          BEQ     65$             ;BR IF OK!
024672  104006                          HLT     6               ;BCC CALCULATION ERROR
024674  104401                  65$:    SCOP1                   ;SW09=1?
024676  012737  024704  001220          MOV     #3$,LOCK        ;SET IF SW09=1
024704  104412                  3$:     MSTCLR                  ;ISSUE A MSTCLR
024706  012777  000010  154446          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
024714  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
024720  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
024724  000010                          8.                      ;SHIFTS PERFORMED
024726  000125                          ^B<01010101>                    ;DATA CHAR
024730  000000                          0                       ;PREVIOUS BCC
024732  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
024736  000125                          ^B<01010101>                    ;DATA TO THE A REGISTER
024740  000000                          0                       ;DATA TO THE B REGISTER
024742  000010                          10                      ;POLY SELT. (RAMOUTPUT)
024744  012710  060000                  MOV     #BCC,(R0)
024750  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
024752  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
024754  104415                          ROMCLK                  ;
024756  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
024762  017704  154400                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
024766  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
024770  001401                          BEQ     66$             ;BR IF OK!
024772  104006                          HLT     6               ;BCC CALCULATION ERROR
024774  104401                  66$:    SCOP1                   ;SW09=1?
024776  012737  025004  001220          MOV     #4$,LOCK        ;SET IF SW09=1
025004  104412                  4$:     MSTCLR                  ;ISSUE A MSTCLR
025006  012777  000010  154346          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025014  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025020  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025024  000010                          8.                      ;SHIFTS PERFORMED
025026  000377                          ^B<11111111>                    ;DATA CHAR
025030  000000                          0                       ;PREVIOUS BCC
025032  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025036  000377                          ^B<11111111>                    ;DATA TO THE A REGISTER
025040  000000                          0                       ;DATA TO THE B REGISTER
025042  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025044  012710  060000                  MOV     #BCC,(R0)
025050  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025052  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025054  104415                          ROMCLK                  ;
025056  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025062  017704  154300                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025066  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025070  001401                          BEQ     67$             ;BR IF OK!
025072  104006                          HLT     6               ;BCC CALCULATION ERROR
025074  104401                  67$:    SCOP1                   ;SW09=1?
025076  012737  025104  001220          MOV     #5$,LOCK        ;SET IF SW09=1
025104  104412                  5$:     MSTCLR                  ;ISSUE A MSTCLR
```

```
025106  012777  000010  154246          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025114  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025120  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025124  000010                          8.                      ;SHIFTS PERFORMED
025126  000001                          "B<00000001>                    ;DATA CHAR
025130  000000                          0                       ;PREVIOUS BCC
025132  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025136  000001                          "B<00000001>                    ;DATA TO THE A REGISTER
025140  000000                          0                       ;DATA TO THE B REGISTER
025142  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025144  012710  060000                  MOV     #BCC,(R0)
025150  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025152  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025154  104415                          ROMCLK                  ;
025156  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025162  017704  154200                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025166  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025170  001401                          BEQ     68$             ;BR IF OK!
025172  104006                          HLT     6               ;BCC CALCULATION ERROR
025174  104401                  68$:    SCOP1                   ;SW09=1?
025176  012737  025204  001220          MOV     #6$,LOCK        ;SET IF SW09=1
025204  104412                  6$:     MSTCLR                  ;ISSUE A MSTCLR
025206  012777  000010  154146          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025214  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025220  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025224  000010                          8.                      ;SHIFTS PERFORMED
025226  000004                          "B<00000100>                    ;DATA CHAR
025230  000000                          0                       ;PREVIOUS BCC
025232  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025236  000004                          "B<00000100>                    ;DATA TO THE A REGISTER
025240  000000                          0                       ;DATA TO THE B REGISTER
025242  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025244  012710  060000                  MOV     #BCC,(R0)
025250  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025252  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025254  104415                          ROMCLK                  ;
025256  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025262  017704  154100                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025266  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025270  001401                          BEQ     69$             ;BR IF OK!
025272  104006                          HLT     6               ;BCC CALCULATION ERROR
025274  104401                  69$:    SCOP1                   ;SW09=1?
025276  012737  025304  001220          MOV     #7$,LOCK        ;SET IF SW09=1
025304  104412                  7$:     MSTCLR                  ;ISSUE A MSTCLR
025306  012777  000010  154046          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025314  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025320  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025324  000010                          8.                      ;SHIFTS PERFORMED
025326  000020                          "B<00010000>                    ;DATA CHAR
025330  000000                          0                       ;PREVIOUS BCC
025332  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025336  000020                          "B<00010000>                    ;DATA TO THE A REGISTER
025340  000000                          0                       ;DATA TO THE B REGISTER
025342  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025344  012710  060000                  MOV     #BCC,(R0)
025350  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
```

```
025352  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025354  104415                          ROMCLK                  ;
025356  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025362  017704  154000                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025366  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025370  001401                          BEQ     70$             ;BR IF OK!
025372  104006                          HLT     6               ;BCC CALCULATION ERROR
025374  104401                  70$:    SCOP1                   ;SW09=1?
025376  012737  025404  001220          MOV     #8$,LOCK        ;SET IF SW09=1
025404  104412                  8$:     MSTCLR                  ;ISSUE A MSTCLR
025406  012777  000010  153746          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025414  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025420  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025424  000010                          8.                      ;SHIFTS PERFORMED
025426  000100                          "B<01000000>                    ;DATA CHAR
025430  000000                          0                       ;PREVIOUS BCC
025432  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025436  000100                          "B<01000000>                    ;DATA TO THE A REGISTER
025440  000000                          0                       ;DATA TO THE B REGISTER
025442  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025444  012710  060000                  MOV     #BCC,(R0)
025450  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025452  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025454  104415                          ROMCLK                  ;
025456  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025462  017704  153700                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025466  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025470  001401                          BEQ     71$             ;BR IF OK!
025472  104006                          HLT     6               ;BCC CALCULATION ERROR
025474  104401                  71$:    SCOP1                   ;SW09=1?
025476  012737  025504  001220          MOV     #9$,LOCK        ;SET IF SW09=1
025504  104412                  9$:     MSTCLR                  ;ISSUE A MSTCLR
025506  012777  000010  153646          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025514  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025520  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025524  000010                          8.                      ;SHIFTS PERFORMED
025526  000006                          "B<00000110>                    ;DATA CHAR
025530  000000                          0                       ;PREVIOUS BCC
025532  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025536  000006                          "B<00000110>                    ;DATA TO THE A REGISTER
025540  000000                          0                       ;DATA TO THE B REGISTER
025542  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025544  012710  060000                  MOV     #BCC,(R0)
025550  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025552  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025554  104415                          ROMCLK                  ;
025556  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025562  017704  153600                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025566  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025570  001401                          BEQ     72$             ;BR IF OK!
025572  104006                          HLT     6               ;BCC CALCULATION ERROR
025574  104401                  72$:    SCOP1                   ;SW09=1?
025576  012737  025604  001220          MOV     #100$,LOCK      ;SET IF SW09=1
025604  104412                  100$:   MSTCLR                  ;ISSUE A MSTCLR
025606  012777  000010  153546          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025614  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
```

```
025620  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025624  000010                          8.                      ;SHIFTS PERFORMED
025626  000120                          ^B<01010000>                    ;DATA CHAR
025630  000000                          0                       ;PREVIOUS BCC
025632  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025636  000120                          ^B<01010000>                    ;DATA TO THE A REGISTER
025640  000000                          0                       ;DATA TO THE B REGISTER
025642  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025644  012710  060000                  MOV     #BCC,(R0)
025650  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025652  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025654  104415                          ROMCLK                  ;
025656  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025662  017704  153500                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025666  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025670  001401                          BEQ     73$             ;BR IF OK!
025672  104006                          HLT     6               ;BCC CALCULATION ERROR
025674  104401                  73$:    SCOP1                   ;SW09=1?
025676  012737  025704  001220          MOV     #101$,LOCK      ;SET IF SW09=1
025704  104412                  101$:   MSTCLR                  ;ISSUE A MSTCLR
025706  012777  000010  153446          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
025714  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
025720  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
025724  000010                          8.                      ;SHIFTS PERFORMED
025726  000320                          ^B<11010000>                    ;DATA CHAR
025730  000000                          0                       ;PREVIOUS BCC
025732  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
025736  000320                          ^B<11010000>                    ;DATA TO THE A REGISTER
025740  000000                          0                       ;DATA TO THE B REGISTER
025742  000010                          10                      ;POLY SELT. (RAMOUTPUT)
025744  012710  060000                  MOV     #BCC,(R0)
025750  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
025752  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
025754  104415                          ROMCLK                  ;
025756  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
025762  017704  153400                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
025766  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
025770  001401                          BEQ     74$             ;BR IF OK!
025772  104006                          HLT     6               ;BCC CALCULATION ERROR
025774  104401                  74$:    SCOP1                   ;SW09=1?
025776  012737  026004  001220          MOV     #102$,LOCK      ;SET IF SW09=1
026004  104412                  102$:   MSTCLR                  ;ISSUE A MSTCLR
026006  012777  000010  153346          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
026014  005037  031652                  CLR     CALBCC          ;ZERO CALCULATED BCC (SOFTWARE)
026020  004537  031474                  JSR     R5,SIMBCC       ;GO AND CALCULATE A NEW BCC
026024  000010                          8.                      ;SHIFTS PERFORMED
026026  000300                          ^B<11000000>                    ;DATA CHAR
026030  000000                          0                       ;PREVIOUS BCC
026032  004337  031420                  JSR     R3,L.DATA       ;GO AND LOAD DATA INTO THE DV11
026036  000300                          ^B<11000000>                    ;DATA TO THE A REGISTER
026040  000000                          0                       ;DATA TO THE B REGISTER
026042  000010                          10                      ;POLY SELT. (RAMOUTPUT)
026044  012710  060000                  MOV     #BCC,(R0)
026050  104415                          ROMCLK                  ;DO A BCC CALCULATION (HRDW)
026052  010210                          MOV     R2,(R0)         ;DO A DATAXFR FROM BCC REG TO DVRIC
026054  104415                          ROMCLK                  ;
```

```
026056  013705  031652                  MOV     CALBCC,R5       ;PUT SOFTWARE BCC INTO EXPECTED
026062  017704  153300                  MOV     @DVRIC,R4       ;PUT HRDW BCC INTO RECEIVERD
026066  020504                          CMP     R5,R4           ;DOES EXPECTED = FOUND??
026070  001401                          BEQ     75$             ;BR IF OK!
026072  104006                          HLT     6               ;BCC CALCULATION ERROR
026074  104401                  75$:    SCOP1                   ;SW09=1?
026076  104400                          SCOPE



                                        ;*********************** TEST 106 ******************************
                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE LRC8 .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;**************************************************************

                                ;  TEST 106
                                ;--------------
026100  012737  000106  001226  TST106: MOV     #106,TSTNO
026106  012737  026256  001216          MOV     #TST107,NEXT
026114  012737  000200  031646          MOV     #LRC8,XPOLY     ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
026122  012702  030346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
026126  013700  001400                  MOV     DVSFR,R0        ;LOAD DATA XFER FROM BCC TO DVRIC
026132  005001                          CLR     R1              ;SET DATA POINTER TO 0
026134  005037  031652                  CLR     CALBCC          ;ZERO SOFTWARE BCC
026140  104412                  1$:     MSTCLR                  ;CLEAR THE DV11
026142  012777  000010  153212          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL
026150  010137  026202                  MOV     R1,2$           ;LOAD SOFTWARE CHAR
026154  010137  026212                  MOV     R1,4$           ;LOAD HRDW CHAR
026160  013737  031652  026204          MOV     CALBCC,3$       ;PLACE PREVIOUS BCC FOR SOFTWARE
026166  013737  031652  026214          MOV     CALBCC,5$       ;PLACE PREVIOUS BCC FOR HRDW
026174  004537  031474                  JSR     R5,SIMBCC       ;HAVE SOFTWARE GET THE RIGHT BCC
026200  000010                          8.                      ;EIGHT SHIFTS
026202  000001                  2$:     .BLKW 1                 ;DATA
026204  000001                  3$:     .BLKW 1                 ;PREVIOS BCC
026206  004337  031420                  JSR     R3,L.DATA       ;LOAD DV11 REGISTERS
026212  000001                  4$:     .BLKW 1                 ;TO BE PLACED INTO THE "A" REG
026214  000001                  5$:     .BLKW 1                 ;TO BE PLACED INTO THE "B" REG
026216  000000                          0                       ;TO BE LEFT IN THE RAM OUTPUT REG
026220  012710  060000                  MOV     #BCC,(R0)
026224  104415                          ROMCLK                  ;DO A BCC OPERATION
026226  010210                          MOV     R2,(R0)         ;DO A DATA XFER OPR.
026230  104415                          ROMCLK                  ;
026232  013705  031652                  MOV     CALBCC,R5       ;GET GOOD BCC
026236  017704  153124                  MOV     @DVRIC,R4       ;GET ???? BCC
026242  020504                          CMP     R5,R4           ;ARE THEY THE SAME?
026244  001401                          BEQ     6$              ;BR IF YES
026246  104006                          HLT     6               ;BCC ERROR
026250  105201                  6$:     INCB    R1              ;UPDATE DATA CHAR
026252  001332                          BNE     1$              ;BR IF NOT ALL DATA DONE.
026254  104400                          SCOPE                   ;SCOPE THIS TEST



                                        ;*********************** TEST 107 ******************************
                                        ;*TEST TO RUN A BINARY COUNT (000-377)PATTERN
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE CRC16 .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;***********************************************************

                                ;  TEST 1Ø7
                                ;--------------
Ø26256  Ø12737  ØØØ1Ø7  ØØ1226  TST1Ø7: MOV     #1Ø7,TSTNO
Ø26264  Ø12737  Ø26434  ØØ1216          MOV     #TST11Ø,NEXT
Ø26272  Ø12737  12ØØØ1  Ø31646          MOV     #CRC16,XPOLY     ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
Ø263ØØ  Ø127Ø2  Ø3Ø346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
Ø263Ø4  Ø137ØØ  ØØ14ØØ                  MOV     DVSFR,RØ         ;LOAD DATA XFER FROM BCC TO DVRIC
Ø2631Ø  ØØ5ØØ1                          CLR     R1               ;SET DATA POINTER TO Ø
Ø26312  ØØ5Ø37  Ø31652                  CLR     CALBCC           ;ZERO SOFTWARE BCC
Ø26316  1Ø4412                  1$:     MSTCLR                   ;CLEAR THE DV11
Ø2632Ø  Ø12777  ØØØØ1Ø  153Ø34          MOV     #BIT3,@DVSCR     ;SET SOURCE SEL
Ø26326  Ø1Ø137  Ø2636Ø                  MOV     R1,2$            ;LOAD SOFTWARE CHAR
Ø26332  Ø1Ø137  Ø2637Ø                  MOV     R1,4$            ;LOAD HRDW CHAR
Ø26336  Ø13737  Ø31652  Ø26362          MOV     CALBCC,3$        ;PLACE PREVIOUS BCC FOR SOFTWARE
Ø26344  Ø13737  Ø31652  Ø26372          MOV     CALBCC,5$        ;PLACE PREVIOUS BCC FOR HRDW
Ø26352  ØØ4537  Ø31474                  JSR     R5,SIMBCC        ;HAVE SOFTWARE GET THE RIGHT BCC
Ø26356  ØØØØ1Ø                          8.                       ;EIGHT SHIFTS
Ø2636Ø  ØØØØØ1                  2$:     .BLKW 1                  ;DATA
Ø26362  ØØØØØ1                  3$:     .BLKW 1                  ;PREVIOS BCC
Ø26364  ØØ4337  Ø3142Ø                  JSR     R3,L.DATA        ;LOAD DV11 REGISTERS
Ø2637Ø  ØØØØØ1                  4$:     .BLKW 1                  ;TO BE PLACED INTO THE "A" REG
Ø26372  ØØØØØ1                  5$:     .BLKW 1                  ;TO BE PLACED INTO THE "B" REG
Ø26374  ØØØØ1Ø                          1Ø                       ;TO BE LEFT IN THE RAM OUTPUT REG
Ø26376  Ø1271Ø  Ø6ØØØØ                  MOV     #BCC,(RØ)
Ø264Ø2  1Ø4415                          ROMCLK                   ;DO A BCC OPERATION
Ø264Ø4  Ø1Ø21Ø                          MOV     R2,(RØ)          ;DO A DATA XFER OPR.
Ø264Ø6  1Ø4415                          ROMCLK                   ;
Ø2641Ø  Ø137Ø5  Ø31652                  MOV     CALBCC,R5        ;GET GOOD BCC
Ø26414  Ø177Ø4  152746                  MOV     @DVRIC,R4        ;GET ???? BCC
Ø2642Ø  Ø2Ø5Ø4                          CMP     R5,R4            ;ARE THEY THE SAME?
Ø26422  ØØ14Ø1                          BEQ     6$               ;BR IF YES
Ø26424  1Ø4ØØ6                          HLT     6                ;BCC ERROR
Ø26426  1Ø52Ø1                  6$:     INCB    R1               ;UPDATE DATA CHAR
Ø2643Ø  ØØ1332                          BNE     1$               ;BR IF NOT ALL DATA DONE.
Ø26432  1Ø44ØØ                          SCOPE                    ;SCOPE THIS TEST


                                        ;************************ TEST 11Ø *****************************
                                        ;*TEST TO RUN A BINARY COUNT (ØØØ-377)PATTERN
                                        ;*THROUGH THE BCC GENERATION LOGIC.
                                        ;*THE POLYNOMIAL USED WILL BE CRC.CCITT .
                                        ;*THE BCC REGISTER WILL BE BUILT UP AFTER
                                        ;*EACH CHARACTER -*NOT ZEROED OUT*-
                                        ;;***********************************************************

                                ;  TEST 11Ø
                                ;--------------
Ø26434  Ø12737  ØØØ11Ø  ØØ1226  TST11Ø: MOV     #11Ø,TSTNO
Ø26442  Ø12737  Ø26612  ØØ1216          MOV     #TST111,NEXT
Ø2645Ø  Ø12737  1Ø2Ø1Ø  Ø31646          MOV     #CRC.CCITT,XPOLY         ;LOAD POLYNOMIAL FOR SOFTWARE CAL.
```

```
Ø26456  Ø127Ø2  Ø3Ø346                  MOV     #XFR+BIT7+BIT6+BIT5+BIT2+BIT1,R2
Ø26462  Ø137ØØ  ØØ14ØØ                  MOV     DVSFR,RØ         ;LOAD DATA XFER FROM BCC TO DVRIC
Ø26466  ØØ5ØØ1                          CLR     R1               ;SET DATA POINTER TO Ø
Ø2647Ø  ØØ5Ø37  Ø31652                  CLR     CALBCC           ;ZERO SOFTWARE BCC
Ø26474  1Ø4412                  1$:     MSTCLR                   ;CLEAR THE DV11
Ø26476  Ø12777  ØØØØ1Ø  152656          MOV     #BIT3,@DVSCR     ;SET SOURCE SEL
Ø265Ø4  Ø1Ø137  Ø26536                  MOV     R1,2$            ;LOAD SOFTWARE CHAR
Ø2651Ø  Ø1Ø137  Ø26546                  MOV     R1,4$            ;LOAD HRDW CHAR
Ø26514  Ø13737  Ø31652  Ø2654Ø          MOV     CALBCC,3$        ;PLACE PREVIOUS BCC FOR SOFTWARE
Ø26522  Ø13737  Ø31652  Ø2655Ø          MOV     CALBCC,5$        ;PLACE PREVIOUS BCC FOR HRDW
Ø2653Ø  ØØ4537  Ø31474                  JSR     R5,SIMBCC        ;HAVE SOFTWARE GET THE RIGHT BCC
Ø26534  ØØØØ1Ø                          8.                       ;EIGHT SHIFTS
Ø26536  ØØØØØ1                  2$:     .BLKW 1                  ;DATA
Ø2654Ø  ØØØØØ1                  3$:     .BLKW 1                  ;PREVIOS BCC
Ø26542  ØØ4337  Ø3142Ø                  JSR     R3,L.DATA        ;LOAD DV11 REGISTERS
Ø26546  ØØØØØ1                  4$:     .BLKW 1                  ;TO BE PLACED INTO THE "A" REG
Ø2655Ø  ØØØØØ1                  5$:     .BLKW 1                  ;TO BE PLACED INTO THE "B" REG
Ø26552  ØØØØ3Ø                          3Ø                       ;TO BE LEFT IN THE RAM OUTPUT REG
Ø26554  Ø1271Ø  Ø6ØØØØ                  MOV     #BCC,(RØ)
Ø2656Ø  1Ø4415                          ROMCLK                   ;DO A BCC OPERATION
Ø26562  Ø1Ø21Ø                          MOV     R2,(RØ)          ;DO A DATA XFER OPR.
Ø26564  1Ø4415                          ROMCLK                   ;
Ø26566  Ø137Ø5  Ø31652                  MOV     CALBCC,R5        ;GET GOOD BCC
Ø26572  Ø177Ø4  15257Ø                  MOV     @DVRIC,R4        ;GET ???? BCC
Ø26576  Ø2Ø5Ø4                          CMP     R5,R4            ;ARE THEY THE SAME?
Ø266ØØ  ØØ14Ø1                          BEQ     6$               ;BR IF YES
Ø266Ø2  1Ø4ØØ6                          HLT     6                ;BCC ERROR
Ø266Ø4  1Ø52Ø1                  6$:     INCB    R1               ;UPDATE DATA CHAR
Ø266Ø6  ØØ1332                          BNE     1$               ;BR IF NOT ALL DATA DONE.
Ø2661Ø  1Ø44ØØ                          SCOPE                    ;SCOPE THIS TEST
```

```

                                          ;*********************** TEST 111 ****************************
                                          ;*TEST THAT SETTING BIT9!BIT7 AND BIT9!BIT6
                                          ;*RECV IE AND RECV INTR PRODUCE AN INTERUPT ON VECTOR "A"
                                          ;:*************************************************************

                                 ;  TEST 111
                                 ;--------------
026612  012737  000111  001226   TST111: MOV     #111,TSTNO
026620  012737  026762  001216           MOV     #TST112,NEXT
026626  012737  000340  177776           MOV     #340,PS            ;LOCK OUT CPU INTERUPTS
026634  104412                           MSTCLR                     ;ISSUE DVRESET
026636  004537  031654                   JSR     R5,SETVEC          ;GO SET VECTOR "A" AND "B"
026642  031676                           NO.ATRAP                   ;"A"
026644  031702                           NO.BTRAP                   ;"B"
026646     340     340                   .BYTE   340,340            ;PRIO. AT 7
026650  005037  177776                   CLR     PS                 ;ZERO CPU PRIO.
026654  012777  001200  152500           MOV     #BIT9!BIT7,@DVSCR       ;SET AN INTERUPT RELATIVE BIT.
026662  000240                           NOP                        ;WAIST TIME
026664  012777  001100  152470           MOV     #BIT9!BIT6,@DVSCR       ;SET THE ALTERNATE RELATIVE BIT.
026672  000240                           NOP                        ;WAIST
026674  005077  152462                   CLR     @DVSCR             ;ZERO REG
026700  004537  031654                   JSR     R5,SETVEC          ;GO RESET VECTORS "A" AND "B"
026704  026744                           2$                         ;"A"
026706  026756                           3$                         ;"B"
026710     340     340                   .BYTE   340,340            ;PRIO. AT 7
026712  052777  001300  152442           BIS     #BIT9!BIT7!BIT9!BIT6,@DVSCR
026720  000240                           NOP                        ;SET BOTH INTERUPT RELATIVE BITS.
026722  104007                           HLT     7                  ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
026724  004537  031654           1$:     JSR     R5,SETVEC          ;RESET VECTORS
026730  031676                           NO.ATRAP                   ;"A"
026732  031702                           NO.BTRAP                   ;"B"
026734     340     340                   .BYTE   340,340
026736  005077  152420                   CLR     @DVSCR             ;DSABLE DV11
026742  104400                           SCOPE                      ;SCOPE THIS TEST.
026744  012706  001200           2$:     MOV     #STACK,SP          ;RESET STACK
026750  005077  152406                   CLR     @DVSCR             ;DSABLE DV11
026754  000763                           BR      1$                 ;RETURN
026756  104012                   3$:     HLT     12                 ;VECTOR HERE WAS WRONG SIDE
026760  000771                           BR      2$

                                          ;*********************** TEST 112 ****************************
                                          ;*TEST THAT SETTING BIT12 AND BIT10
                                          ;*STORE IE AND NPR STAT OVFLOW PRODUCE AN INTERUPT ON VECTOR "B"
                                          ;:*************************************************************

                                 ;  TEST 112
                                 ;--------------
026762  012737  000112  001226   TST112: MOV     #112,TSTNO
026770  012737  027132  001216           MOV     #TST113,NEXT
026776  012737  000340  177776           MOV     #340,PS            ;LOCK OUT CPU INTERUPTS
027004  104412                           MSTCLR                     ;ISSUE DVRESET
027006  004537  031654                   JSR     R5,SETVEC          ;GO SET VECTOR "A" AND "B"
027012  031676                           NO.ATRAP                   ;"A"
027014  031702                           NO.BTRAP                   ;"B"
027016     340     340                   .BYTE   340,340            ;PRIO. AT 7
```

```
027020  005037  177776                   CLR     PS                 ;ZERO CPU PRIO.
027024  012777  010000  152330           MOV     #BIT12,@DVSCR      ;SET AN INTERUPT RELATIVE BIT.
027032  000240                           NOP                        ;WAIST TIME
027034  012777  002000  152320           MOV     #BIT10,@DVSCR      ;SET THE ALTERNATE RELATIVE BIT.
027042  000240                           NOP                        ;WAIST
027044  005077  152312                   CLR     @DVSCR             ;ZERO REG
027050  004537  031654                   JSR     R5,SETVEC          ;GO RESET VECTORS "A" AND "B"
027054  027126                           3$                         ;"A"
027056  027114                           2$                         ;"B"
027060     340     340                   .BYTE   340,340            ;PRIO. AT 7
027062  052777  012000  152272           BIS     #BIT12!BIT10,@DVSCR
027070  000240                           NOP                        ;SET BOTH INTERUPT RELATIVE BITS.
027072  104010                           HLT     10                 ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
027074  004537  031654           1$:     JSR     R5,SETVEC          ;RESET VECTORS
027100  031676                           NO.ATRAP                   ;"A"
027102  031702                           NO.BTRAP                   ;"B"
027104     340     340                   .BYTE   340,340
027106  005077  152250                   CLR     @DVSCR             ;DSABLE DV11
027112  104400                           SCOPE                      ;SCOPE THIS TEST.
027114  012706  001200           2$:     MOV     #STACK,SP          ;RESET STACK
027120  005077  152236                   CLR     @DVSCR             ;DSABLE DV11
027124  000763                           BR      1$                 ;RETURN
027126  104011                   3$:     HLT     11                 ;VECTOR HERE WAS WRONG SIDE
027130  000771                           BR      2$

                                          ;*********************** TEST 113 ****************************
                                          ;*TEST THAT SETTING BIT15!BIT9 AND BIT13!BIT9
                                          ;*NPR STAT INTR AND NPR STAT IE PRODUCE AN INTERUPT ON VECTOR "B"
                                          ;:*************************************************************

                                 ;  TEST 113
                                 ;--------------
027132  012737  000113  001226   TST113: MOV     #113,TSTNO
027140  012737  027310  001216           MOV     #TST114,NEXT
027146  012737  000340  177776           MOV     #340,PS            ;LOCK OUT CPU INTERUPTS
027154  104412                           MSTCLR                     ;ISSUE DVRESET
027156  004537  031654                   JSR     R5,SETVEC          ;GO SET VECTOR "A" AND "B"
027162  031676                           NO.ATRAP                   ;"A"
027164  031702                           NO.BTRAP                   ;"B"
027166     340     340                   .BYTE   340,340            ;PRIO. AT 7
027170  005037  177776                   CLR     PS                 ;ZERO CPU PRIO.
027174  012777  101000  152160           MOV     #BIT15!BIT9,@DVSCR      ;SET AN INTERUPT RELATIVE BIT.
027202  000240                           NOP                        ;WAIST TIME
027204  012777  021000  152150           MOV     #BIT13!BIT9,@DVSCR      ;SET THE ALTERNATE RELATIVE BIT.
027212  000240                           NOP                        ;WAIST
027214  005077  152142                   CLR     @DVSCR             ;ZERO REG
027220  012777  001000  152134           MOV     #BIT9,@DVSCR       ;SET SYS MAINT ENABLE
027226  004537  031654                   JSR     R5,SETVEC          ;GO RESET VECTORS "A" AND "B"
027232  027304                           3$                         ;"A"
027234  027272                           2$                         ;"B"
027236     340     340                   .BYTE   340,340            ;PRIO. AT 7
027240  052777  121000  152114           BIS     #BIT15!BIT9!BIT13!BIT9,@DVSCR
027246  000240                           NOP                        ;SET BOTH INTERUPT RELATIVE BITS.
027250  104010                           HLT     10                 ;SETTING OF THESE BITS FAILED TO PRODUCE AN INTERUPT
027252  004537  031654           1$:     JSR     R5,SETVEC          ;RESET VECTORS
027256  031676                           NO.ATRAP                   ;"A"
```

```
027260  031702                          NO.BTRAP                ;"B"
027262     340     340                  .BYTE   340,340
027264  005077  152072                  CLR     @DVSCR          ;DSABLE DV11
027270  104400                          SCOPE                   ;SCOPE THIS TEST.
027272  012706  001200          2$:     MOV     #STACK,SP       ;RESET STACK
027276  005077  152060                  CLR     @DVSCR          ;DSABLE DV11
027302  000763                          BR      1$              ;RETURN
027304  104011                  3$:     HLT     11              ;VECTOR HERE WAS WRONG SIDE
027306  000771                          BR      2$


                                        ;*********************** TEST 114 ******************************
                                        ;*TEST TO VERIFY THAT VECTOR "A"
                                        ;*OCCURES BEFOR VECTOR "B" EVEN
                                        ;*WHEN VECTOR "B" IS ENABLED BEFORE
                                        ;*VECTOR "A".
                                        ;;*************************************************************

                                ;  TEST 114
                                ;--------------
027310  012737  000114  001226  TST114: MOV     #114,TSTNO
027316  012737  027450  001216          MOV     #TST115,NEXT
027324  012737  000340  177776          MOV     #340,PS         ;SET CPU TO PRIO 7
027332  104412                          MSTCLR                  ;RESET DV11
027334  005003                          CLR     R3              ;SET COUNTER TO 0
027336  012700  000002                  MOV     #2,R0           ;SET TO GET 2 INTERRUPTS
027342  004537  031654                  JSR     R5,SETVEC       ;SET DV11 VECTORS
027346  027416                          2$                      ;RECEIVER INTERRUPTS TO 2$
027350  027440                          3$                      ;TRANSMITTER INTERRUPTS TO 3$
027352     340     340                  .BYTE   340,340         ;SET PRIORITY TO 7 ON INTERRUPT
027354  012777  001000  152000          MOV     #BIT9,@DVSCR    ;SET SYSTEM MAINT
027362  052777  120000  151772          BIS     #BIT15+BIT13,@DVSCR
027370  052777  001300  151764          BIS     #BIT9+BIT7+BIT6,@DVSCR
027376  005037  177776                  CLR     PS              ;SET B INTERRUPT ;A INTERRUPT ;CLEAR CPU STATUS
027402  105203                          INCB    R3              ;HANG WAITING FOR INTERRUPTS
027404  001376                          BNE     .-2             ;
027406  005700                          TST     R0              ;DID TWO INTERRUPTS OCCUR?
027410  001401                          BEQ     1$              ;BR IF YES
027412  104000                          HLT     0               ;EITHER NOT ENOUGH(2) OR TOO MANY (72) INTERRUPTS
027414  104400                  1$:     SCOPE                   ;SCOPE TEST
027416  005300                  2$:     DEC     R0              ;RXISR
027420  022700  000001                  CMP     #1,R0           ;IF RX GOT HERE 1ST R0=1 S/B=1
027424  001401                          BEQ     64$             ;BR IF OK
027426  104011                          HLT     11              ;RX NOT 1ST
027430  042777  000300  151724  64$:    BIC     #BIT7+BIT6,@DVSCR ;DISQUALIFY INTERRUPT REQST
027436  000002                          RTI                     ;RETURN
027440  005300                  3$:     DEC     R0              ;TX ISR
027442  001401                          BEQ     4$              ;IF SCOND INTERRUPT R0 SHOULD NOW =0
027444  104012                          HLT     12              ;TX INTERRUPT OUT OF ORDER
027446  000002                  4$:     RTI                     ;RETURN


                                        ;*********************** TEST 115 ******************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 7 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;;*************************************************************
```

```

                                ;  TEST 115
                                ;--------------
027450  012737  000115  001226  TST115: MOV     #115,TSTNO
027456  012737  027540  001216          MOV     #TST116,NEXT
027464  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
027472  104412                          MSTCLR                  ;CLEAN DV11
027474  004537  031654                  JSR     R5,SETVEC       ;PREPARE VECTORS
027500  031676                          NO.ATRAP                        ;"A"
027502  031702                          NO.BTRAP                ;"B"
027504     340     340                  .BYTE   340,340         ;PRIO. AT 7
027506  012777  001300  151646          MOV     #BIT9+BIT7+BIT6,@DVSCR
027514  012737  000340  177776          MOV     #340,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027522  000240                          NOP                     ;WAIST
027524  005077  151632          1$:     CLR     @DVSCR          ;DSABLE DV11
027530  104400                          SCOPE                   ;SCOPE THIS TEST
027532  012716  027524          2$:     MOV     #1$,(SP)        ;SET FOR RETURN
027536  000002                          RTI                     ;


                                        ;*********************** TEST 116 ******************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 6 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;;*************************************************************

                                ;  TEST 116
                                ;--------------
027540  012737  000116  001226  TST116: MOV     #116,TSTNO
027546  012737  027630  001216          MOV     #TST117,NEXT
027554  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
027562  104412                          MSTCLR                  ;CLEAN DV11
027564  004537  031654                  JSR     R5,SETVEC       ;PREPARE VECTORS
027570  031676                          NO.ATRAP                        ;"A"
027572  031702                          NO.BTRAP                ;"B"
027574     340     340                  .BYTE   340,340         ;PRIO. AT 7
027576  012777  001300  151556          MOV     #BIT9+BIT7+BIT6,@DVSCR
027604  012737  000300  177776          MOV     #300,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027612  000240                          NOP                     ;WAIST
027614  005077  151542          1$:     CLR     @DVSCR          ;DSABLE DV11
027620  104400                          SCOPE                   ;SCOPE THIS TEST
027622  012716  027614          2$:     MOV     #1$,(SP)        ;SET FOR RETURN
027626  000002                          RTI                     ;


                                        ;*********************** TEST 117 ******************************
                                        ;*PRIORITY INTERUPT TESTS.
                                        ;*SET PS TO PRIORITY 5 AND VERIFY
                                        ;*THAT THE DV11 DOESN'T INTERUPT.
                                        ;;*************************************************************

                                ;  TEST 117
                                ;--------------
027630  012737  000117  001226  TST117: MOV     #117,TSTNO
027636  012737  027720  001216          MOV     #TST120,NEXT
027644  012737  000340  177776          MOV     #340,PS         ;LOCK OUT INTERUPTS
```

```
027652  104412                                  MSTCLR                      ;CLEAN DV11
027654  004537  031654                          JSR     R5,SETVEC           ;PREPARE VECTORS
027660  031676                                  NO.ATRAP                        ;"A"
027662  031702                                  NO.BTRAP                    ;"B"
027664     340     340                          .BYTE   340,340             ;PRIO. AT 7
027666  012777  001300  151466                  MOV     #BIT9+BIT7+BIT6,@DVSCR
027674  012737  000240  177776                  MOV     #240,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027702  000240                                  NOP                         ;WAIST
027704  005077  151452          1$:             CLR     @DVSCR              ;DSABLE DV11
027710  104400                                  SCOPE                       ;SCOPE THIS TEST
027712  012716  027704          2$:             MOV     #1$,(SP)            ;SET FOR RETURN
027716  000002                                  RTI                         ;


                                                ;*********************** TEST 120 ****************************
                                                ;*PRIORITY INTERUPT TESTS.
                                                ;*SET PS TO PRIORITY 4 AND VERIFY
                                                ;*THAT THE DV11 DOES INTERUPT.
                                                ;;*************************************************************

                                        ;  TEST 120
                                        ;--------------
027720  012737  000120  001226  TST120: MOV     #120,TSTNO
027726  012737  030012  001216          MOV     #TST121,NEXT
027734  012737  000340  177776          MOV     #340,PS             ;LOCK OUT INTERUPTS
027742  104412                          MSTCLR                      ;CLEAN DV11
027744  004537  031654                  JSR     R5,SETVEC           ;PREPARE VECTORS
027750  030004                          2$                          ;"A"
027752  031702                          NO.BTRAP                    ;"B"
027754     340     340                  .BYTE   340,340             ;PRIO. AT 7
027756  012777  001300  151376          MOV     #BIT9+BIT7+BIT6,@DVSCR
027764  012737  000200  177776          MOV     #200,PS ;SET CPU PRIO AND ENABLE VECT. "A"
027772  000240                          NOP                         ;WAIST
027774  104007                          HLT     7                   ;DV FAILED TO INTERUPT
027776  005077  151360          1$:     CLR     @DVSCR              ;DSABLE DV11
030002  104400                          SCOPE                       ;SCOPE THIS TEST
030004  012716  027776          2$:     MOV     #1$,(SP)            ;SET FOR RETURN
030010  000002                          RTI                         ;


                                        ;*********************** TEST 121 ****************************
                                        ;*TEST THAT BIT15(NPR STATUS INTR) WILL
                                        ;*SET WHEN AN ENTRY IS MADE INTO THE
                                        ;*NPR STATUS REGISTER.
                                        ;*ALSO VERIFY THAT READING THE DVNSR CLEARS DVSCR BIT15.
                                        ;;*************************************************************

                                        ;  TEST 121
                                        ;--------------
030012  012737  000121  001226  TST121: MOV     #121,TSTNO
030020  012737  030114  001216          MOV     #TST122,NEXT
030026  104412                          MSTCLR                      ;RESET DV11
030030  005777  151326                  TST     @DVSCR              ;MAKE SURE 15 =0
030034  100001                          BPL     64$                 ;BR IF OK
030036  104000                          HLT     0                   ;15 S/B=0
030040  012777  000010  151314  64$:    MOV     #BIT3,@DVSCR        ;SET SOURCE SEC
```

```
030046  012777  030367  151324          MOV     #XFR+BIT7+BIT6+BIT5+BIT4+BIT2+BIT1+BIT0,@DVSFR
030054  005000                          CLR     R0                  ;XFR ALU RESULT TO DVNSR
030056  104415                          ROMCLK                      ;EXECUTE
030060  005777  151276          1$:     TST     @DVSCR              ;15=1?
030064  100404                          BMI     2$                  ;BR IF YES
030066  104414                          DELAY                       ;WASTE TIME
030070  005200                          INC     R0                  ;DO DELAY AND WAIT
030072  001372                          BNE     1$                  ;
030074  104000                          HLT     0                   ;15 NEVER SET
030076  005777  151300          2$:     TST     @DVNSR              ;READ NSR
030102  005777  151254                  TST     @DVSCR              ;DID 15 CLEAR IN SCR?
030106  100001                          BPL     3$                  ;BR IF YES
030110  104000                          HLT     0                   ;DVSCR IS NOT CLEARED BY READING NSR
030112  104400                  3$:     SCOPE                       ;SCOPE TEST


                                        ;*********************** TEST 122 ****************************
                                        ;*TEST TO WRITE PATTERNS THROUGH
                                        ;*THE NPR STATUS REGISTER.
                                        ;*BITS WRITTEN: 11,10,09,08,03,02,01,00
                                        ;*(WHEN BIT 15 OF DVSCR IS SET SO SHOULD BIT 15 OF DVNSR)
                                        ;;*************************************************************

                                        ;  TEST 122
                                        ;--------------
030114  012737  000122  001226  TST122: MOV     #122,TSTNO
030122  012737  030314  001216          MOV     #TST123,NEXT
030130  012737  030172  001220          MOV     #1$,LOCK
030136  104412                          MSTCLR                      ;RESET DV11
030140  012777  000010  151214          MOV     #BIT3,@DVSCR        ;SET SOURCE SEL
030146  005000                          CLR     R0                  ;ZERO LINE NUMBER POINTER
030150  005037  001250                  CLR     TEMP2               ;ZERO DATA
030154  013703  001402                  MOV     DVNSR,R3            ;SET POINTER
030160  013702  001400                  MOV     DVSFR,R2            ;
030164  012737  100000  001246          MOV     #BIT15,TEMP1        ;SET CORRESPONDING BIT TO BE FOUND IN NSR.
030172  012712  030267          1$:     MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT1+BIT0,(R2)
030176  104415                          ROMCLK                      ;XFR RAM OUTPUT TO NSR
030200  005777  151156                  TST     @DVSCR              ;WAIT FOR
030204  100375                          BPL     .-4                 ;
030206  013705  001246                  MOV     TEMP1,R5            ;GET GOOD 15=1 DATA
030212  011304                          MOV     (R3),R4             ;READ NSR
030214  020504                          CMP     R5,R4               ;OK?
030216  001401                          BEQ     2$                  ;BR IF GOOD
030220  104013                          HLT     13                  ;DVNSR HAS WRONG DATA
030222  104401                  2$:     SCOP1                       ;LOCK ON DATA?
030224  004537  031342                  PERFORM ,SETSCAN            ;UPDATE MSCANNER
030230  000001                          1                           ;BY ONE
030232  005237  001246                  INC     TEMP1               ;UPDATE DATA
030236  032700  000020                  BIT     #BIT4,R0            ;ALL DONE?
030242  001753                          BEQ     1$                  ;BR IF NO.
030244  042737  077777  001246          BIC     #^C<BIT15>,TEMP1 ;CLEAR ALL BUT ENTRY PRESENT
030252  005000                          CLR     R0                  ;ZERO LINE POINTER
030254  005001                          CLR     R1                  ;ZERO MSCANNER. POINTER
030256  005237  001250                  INC     TEMP2               ;UPDATE HIGH BYTE DATA
030262  153737  001250  001247          BISB    TEMP2,TEMP1+1       ;PLACE IN GOOD LOCATION
030270  013712  001250                  MOV     TEMP2,(R2)          ;RAM ADDRESS
```

```
030274  052712  020000                  BIS     #RAM,(R2)       ;RAM READ
030300  104415                          ROMCLK                  ;
030302  032737  000020  001250          BIT     #BIT4,TEMP2     ;ALL DONE?
030310  001730                          BEQ     1$              ;BR IF NO
030312  104400                          SCOPE                   ;SCOPE TEST.

                                        ;*********  FIRST PLANNED ATTEMPT *********
                                        ;*********       TO EXECUTE NPR.   *********
                                        ;*-----------------------------------------

                                        ;********************** TEST 123 *****************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                        ;*PLACED INTO IT AND READ INTO THE DV11 AND XFERED
                                        ;*INTO THE DVRIC REGISTER.
                                        ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                        ;;***************************************************************

                                ;  TEST 123
                                ;--------------
030314  012737  000123  001226  TST123: MOV     #123,TSTNO
030322  012737  030454  001216          MOV     #TST124,NEXT
030330  012737  030412  001220          MOV     #1$,LOCK
030336  104412                          MSTCLR                  ;RESET DV11
030340  012777  000010  151014          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
030346  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
030352  013702  001400                  MOV     DVSFR,R2        ;
030356  012777  032756  151012          MOV     #NPRLOC,@DVSRA
030364  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
030370  104415                          ROMCLK                  ;EXECUTE
030372  012712  030263                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
030376  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
030400  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
030404  005005                          CLR     R5              ;CLEAR GOOD
030406  005077  150764                  CLR     @DVSRA          ;CLEAR DATA PORT
030412  012712  040000          1$:     MOV     #NPR,(R2)       ;DO THE NPR
030416  104415                          ROMCLK                  ;***NOW***
030420  012712  030226                  MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
030424  104415                          ROMCLK                  ;XFR NPR OUTPUT TO DVRIC
030426  011304                          MOV     (R3),R4         ;READ RIC
030430  013705  032756                  MOV     NPRLOC,R5 ;READ GOOD DATA
030434  020504                          CMP     R5,R4           ;OK?
030436  001401                          BEQ     2$              ;BR IF YES
030440  104013                          HLT     13              ;NPR FUNCTION FAILED
030442  104401                  2$:     SCOP1                   ;LOOP?
030444  105237  032756                  INCB    NPRLOC ;UPDATE DATA
030450  001360                          BNE     1$              ;BR IF MORE
030452  104400                          SCOPE                   ;SCOPE TEST



                                        ;********************** TEST 124 *****************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN "READ" FROM CORE LOCATION
```

```
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                        ;*PLACED INTO IT AND READ INTO THE DV11 AND XFERED
                                        ;*INTO THE DVRIC REGISTER.
                                        ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                        ;;***************************************************************

                                ;  TEST 124
                                ;--------------
030454  012737  000124  001226  TST124: MOV     #124,TSTNO
030462  012737  030616  001216          MOV     #TST125,NEXT
030470  012737  030552  001220          MOV     #1$,LOCK
030476  104412                          MSTCLR                  ;RESET DV11
030500  012777  000010  150654          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
030506  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
030512  013702  001400                  MOV     DVSFR,R2        ;
030516  012777  032757  150652          MOV     #NPRLOC+1,@DVSRA
030524  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
030530  104415                          ROMCLK                  ;EXECUTE
030532  012712  030263                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
030536  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
030540  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
030544  005005                          CLR     R5              ;CLEAR GOOD
030546  005077  150624                  CLR     @DVSRA          ;CLEAR DATA PORT
030552  012712  040000          1$:     MOV     #NPR,(R2)       ;DO THE NPR
030556  104415                          ROMCLK                  ;***NOW***
030560  012712  030226                  MOV     #XFR+BIT7+BIT4+BIT2+BIT1,(R2)
030564  104415                          ROMCLK                  ;XFR NPR OUTPUT TO DVRIC
030566  011304                          MOV     (R3),R4         ;READ RIC
030570  013705  032756                  MOV     NPRLOC,R5       ;GET GOOD DATA
030574  000305                          SWAB    R5              ;
030576  020504                          CMP     R5,R4           ;OK?
030600  001401                          BEQ     2$              ;BR IF YES
030602  104013                          HLT     13              ;NPR FUNCTION FAILED
030604  104401                  2$:     SCOP1                   ;LOOP?
030606  105237  032757                  INCB    NPRLOC+1 ;UPDATE DATA
030612  001357                          BNE     1$              ;BR IF MORE
030614  104400                          SCOPE                   ;SCOPE TEST



                                        ;********************** TEST 125 *****************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN "WRITTEN" INTO CORE LOCATION
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOCATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                        ;*WRITTEN INTO IT BY THE DV11 NPR LOGIC.
                                        ;*NOTE: THIS TEST USES AN EVEN ADDRESS FOR THE NPR OPERATION
                                        ;;***************************************************************

                                ;  TEST 125
                                ;--------------
030616  012737  000125  001226  TST125: MOV     #125,TSTNO
030624  012737  031000  001216          MOV     #TST126,NEXT
030632  012737  030714  001220          MOV     #1$,LOCK
030640  104412                          MSTCLR                  ;RESET DV11
```

```
030642  012777  000010  150512          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
030650  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
030654  013702  001400                  MOV     DVSFR,R2        ;
030660  012777  032756  150510          MOV     #NPRLOC,@DVSRA
030666  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
030672  104415                          ROMCLK                  ;EXECUTE
030674  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
030700  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
030702  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
030706  005005                          CLR     R5              ;CLEAR GOOD
030710  005077  150462                  CLR     @DVSRA          ;CLEAR DATA PORT
030714  005037  032756          1$:     CLR     NPRLOC          ;CLEAR NPR LOC
030720  012712  020000                  MOV     #RAM,(R2)       ;DO RAM READ
030724  104415                          ROMCLK                  ;
030726  012712  030265                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
030732  104415                          ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
030734  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
030740  104415                          ROMCLK                  ;***NOW***
030742  017705  150430                  MOV     @DVSRA,R5       ;READ GOOD DATA
030746  013704  032756                  MOV     NPRLOC,R4 ;READ ???? DATA
030752  020504                          CMP     R5,R4           ;OK?
030754  001401                          BEQ     2$              ;BR IF YES
030756  104013                          HLT     13              ;NPR FUNCTION FAILED
030760  104401                  2$:     SCOP1                   ;LOOP?
030762  005277  150410                  INC     @DVSRA          ;UPDATE DATA
030766  032777  000400  150402          BIT     #BIT8,@DVSRA    ;ALL DONE?
030774  001747                          BEQ     1$              ;BR IF NO
030776  104400                          SCOPE                   ;SCOPE TEST



                                        ;************************ TEST 126 ******************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN "WRITTEN" INTO CORE LOCATION
                                        ;*VIA THE NPR LOGIC.
                                        ;*LOACATION "NPRLOC" WILL HAVE A BINARY COUNT PATTERN
                                        ;*WRITTEN INTO IT BY THE DV11 NPR LOGIC.
                                        ;*NOTE: THIS TEST USES AN ODD ADDRESS FOR THE NPR OPERATION.
                                        ;;***************************************************************

                                ;  TEST 126
                                ;--------------
031000  012737  000126  001226  TST126: MOV     #126,TSTNO
031006  012737  031164  001216          MOV     #TST127,NEXT
031014  012737  031076  001220          MOV     #1$,LOCK
031022  104412                          MSTCLR                  ;RESET DV11
031024  012777  000010  150330          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
031032  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
031036  013702  001400                  MOV     DVSFR,R2        ;
031042  012777  032757  150326          MOV     #NPRLOC+1,@DVSRA
031050  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
031054  104415                          ROMCLK                  ;EXECUTE
031056  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
031062  104415                          ROMCLK                  ;DO XFR TO NPR ADD.
031064  005037  032756                  CLR     NPRLOC          ;CLEAR NPR LOCATION
031070  005005                          CLR     R5              ;CLEAR GOOD
```

```
031072  005077  150300                  CLR     @DVSRA          ;CLEAR DATA PORT
031076  005037  032756          1$:     CLR     NPRLOC          ;CLEAR NPR LOC
031102  012712  020000                  MOV     #RAM,(R2)       ;DO RAM READ
031106  104415                          ROMCLK                  ;
031110  012712  030265                  MOV     #XFR+BIT7+BIT5+BIT4+BIT2+BIT0,(R2)
031114  104415                          ROMCLK                  ;XFR RAM OUTPUT TO NPR INPUT DATA
031116  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
031122  104415                          ROMCLK                  ;***NOW***
031124  017705  150246                  MOV     @DVSRA,R5       ;READ GOOD DATA
031130  013704  032756                  MOV     NPRLOC,R4       ;GET DATA
031134  000304                          SWAB    R4              ;
031136  020504                          CMP     R5,R4           ;OK?
031140  001401                          BEQ     2$              ;BR IF YES
031142  104013                          HLT     13              ;NPR FUNCTION FAILED
031144  104401                  2$:     SCOP1                   ;LOOP?
031146  005277  150224                  INC     @DVSRA          ;UPDATE DATA
031152  032777  000400  150216          BIT     #BIT8,@DVSRA    ;ALL DONE?
031160  001746                          BEQ     1$              ;BR IF NO
031162  104400                          SCOPE                   ;SCOPE TEST



                                        ;************************ TEST 127 ******************************
                                        ;*BASIC TEST OF THE NPR OPERATION INSTRUCTION.
                                        ;*TEST THAT THE DV11 CAN DO AN NPR
                                        ;*TO A NON-EXISTANT MEMORY.
                                        ;*TEST THAT BRANCH "A" -NXM H- IS SET AFTER
                                        ;*THE NPR. THEN DO A "SET/CLEAR" CLEAR NXM
                                        ;*AND VERIFY THAT IT IS CLEARED.
                                        ;;***************************************************************

                                ;  TEST 127
                                ;--------------
031164  012737  000127  001226  TST127: MOV     #127,TSTNO
031172  012737  002436  001216          MOV     #.EOP,NEXT
031200  104412                          MSTCLR                  ;RESET DV11
031202  012777  000010  150152          MOV     #BIT3,@DVSCR    ;SET SOURCE SEL.
031210  013703  001366                  MOV     DVRIC,R3        ;SET POINTERS
031214  013702  001400                  MOV     DVSFR,R2        ;
031220  052777  000060  150134          BIS     #BIT5+BIT4,@DVSCR       ;SET EA BITS.
031226  012777  177320  150142          MOV     #177320,@DVSRA  ;LOAD "NXM".
031234  012712  020000                  MOV     #RAM,(R2)       ;RAM READ
031240  104415                          ROMCLK                  ;EXECUTE
031242  012712  031663                  MOV     #XFR+BIT9+BIT8+BIT7+BIT5+BIT4+BIT1+BIT0,(R2)
031246  104415                          ROMCLK                  ;EXECUTE
031250  012712  040000                  MOV     #NPR,(R2)       ;DO THE NPR
031254  104415                          ROMCLK                  ;***NOW***
031256  012712  003000                  MOV     #BIT10+BIT9,(R2)        ;BRANCH "A" NXM
031262  017704  150102                  MOV     @DVLCR,R4       ;READ BRANCH TEST POINTS.
031266  010405                          MOV     R4,R5           ;GET IMAGE
031270  042705  000001                  BIC     #BIT0,R5        ;BR "A" TRUE
031274  052705  000002                  BIS     #BIT1,R5        ;BR B FALSE
031300  020504                          CMP     R5,R4           ;NXM TRUE?
031302  001401                          BEQ     .+4             ;BR IF YES
031304  104006                          HLT     6               ;BR "A" OF NXM FAILED!
031306  012712  050017                  MOV     #S.C+BIT3+BIT2+BIT1+BIT0,(R2)
```

```
                                                              ;SET/CLEAR  CLEAR NXM.
031312  104415                          ROMCLK                ;EXECUTE.
031314  012712  003000                  MOV     #BIT10+BIT9,(R2)       ;BRANCH "A" NXM
031320  017704  150044                  MOV     @DVLCR,R4     ;READ BRANCH TEST POINTS.
031324  010405                          MOV     R4,R5         ;GET IMAGE
031326  052705  000003                  BIS     #BIT1+BIT0,R5 ;BR A FALSE  BR B FALSE
031332  020504                          CMP     R5,R4         ;NXM TRUE?
031334  001401                          BEQ     .+4           ;BR IF NO
031336  104006                          HLT     6             ;BR "A" OF NXM FAILED TO CLEAR!
031340  104400                          SCOPE                 ;SCOPE TEST
```

```
031342                          SETSCAN:
031342  010346                          MOV     R3,-(SP)
031344  052777  000010  150010          BIS     #BIT3,@DVSCR
031352  012503                          MOV     (R5)+,R3
031354  001414                          BEQ     2$
031356  012777  050102  150014  1$:     MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
031364  104415                          ROMCLK
031366  005201                          INC     R1
031370  012777  050102  150002          MOV     #BIT14+BIT12+BIT6+BIT1,@DVSFR
031376  104415                          ROMCLK
031400  005201                          INC     R1
031402  005303                          DEC     R3
031404  001364                          BNE     1$
031406  012603                  2$:     MOV     (SP)+,R3
031410  010100                          MOV     R1,R0
031412  000241                          CLC
031414  006000                          ROR     R0
031416  000205                          EXIT

                                        ;SUBROUTINE TO LOAD DATA INTO "A" AND "B" REG.
                                        ;THE FIRST ARGUMENT WILL LOAD THE "A" REGISTER;
                                        ;THE SECOND ARGUMENT WILL LOAD THE "B" REGISTER;
                                        ;AND THE THIRD ARGUMENT WILL SPECIFY THE POLYNOMIAL USED.

031420  012377  147752          L.DATA: MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "A" REG
031424  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ INSTR.
031430  104415                          ROMCLK                  ;EXECUTE
031432  012710  030261                  MOV     #XFR+BIT7+BIT5+BIT4+BIT0,(R0)
031436  104415                          ROMCLK                  ;DATAXFR FROM RAM OUTPUT TO "A" REGISTER
031440  012377  147732                  MOV     (R3)+,@DVSRA    ;LOAD DATA TO BE PLACED INTO THE "B" REG.
031444  012710  020000                  MOV     #RAM,(R0)       ;DO A ROM READ
031450  104415                          ROMCLK                  ;
031452  012710  030262                  MOV     #XFR+BIT7+BIT5+BIT4+BIT1,(R0)
031456  104415                          ROMCLK                  ;DO A DATA XFER FROM RAM OUTPUT TO THE "B" REG.
031460  012377  147712                  MOV     (R3)+,@DVSRA    ;PLACE DATA TO REMAIN IN THE RAMOUTPUT REG
031464  012710  020000                  MOV     #RAM,(R0)       ;DO A RAM READ TO SPECIFY POLYN.
031470  104415                          ROMCLK                  ;READ
031472  000203                          RTS     R3              ;LEAVE HERE


031474  010046                  SIMBCC: MOV     R0,-(SP)
031476  010146                          MOV     R1,-(SP)
031500  010246                          MOV     R2,-(SP)
031502  012537  001246                  MOV     (R5)+,TEMP1
031506  012537  001250                  MOV     (R5)+,TEMP2
031512  012537  001252                  MOV     (R5)+,TEMP3
031516  005037  031650          1$:     CLR     BCCFBK
031522  013700  001252                  MOV     TEMP3,R0
031526  006037  001250                  ROR     TEMP2
031532  005500                          ADC     R0
031534  032700  000001                  BIT     #BIT0,R0
031540  001402                          BEQ     2$
031542  005137  031650                  COM     BCCFBK
031546  013700  031646          2$:     MOV     XPOLY,R0
031552  005100                          COM     R0
```

```
031554  040037  031650                          BIC     R0,BCCFBK
031560  000241                                  CLC
031562  006037  001252                          ROR     TEMP3
031566  013700  031650                          MOV     BCCFBK,R0
031572  013701  001252                          MOV     TEMP3,R1
031576  010102                                  MOV     R1,R2
031600  040100                                  BIC     R1,R0
031602  043702  031650                          BIC     BCCFBK,R2
031606  050200                                  BIS     R2,R0
031610  043737  031646  001252                  BIC     XPOLY,TEMP3
031616  050037  001252                          BIS     R0,TEMP3
031622  005337  001246                          DEC     TEMP1
031626  001333                                  BNE     1$
031630  013737  001252  031652                  MOV     TEMP3,CALBCC
031636  012602                                  MOV     (SP)+,R2
031640  012601                                  MOV     (SP)+,R1
031642  012600                                  MOV     (SP)+,R0
031644  000205                                  RTS     R5
031646  000000                          XPOLY:  0
031650  000000                          BCCFBK: 0
031652  000000                          CALBCC: 0
        000200                          LRC8=200
        120001                          CRC16=120001
        102010                          CRC.CCITT=102010


031654                                  SETVEC:
031654  012577  147472                          MOV     (R5)+,@DVRVEC
031660  012577  147472                          MOV     (R5)+,@DVTVEC
031664  112577  147464                          MOVB    (R5)+,@DVRLVL
031670  112577  147464                          MOVB    (R5)+,@DVTLVL
031674  000205                                  RTS     R5

031676                                  NO.ATRAP:
031676  104011                                  HLT     11
031700  000002                                  RTI

031702                                  NO.BTRAP:
031702  104012                                  HLT     12
031704  000002                                  RTI
```

```
                                        .NLIST  BEX
031706  050377  044522  040515  EM1:    .ASCIZ  <377>/PRIMARY REGISTER ADDRESSING TIME-OUT/
031754  051777  041505  047117  EM2:    .ASCIZ  <377>"SECONDARY REGISTER READ/WRITE TEST"
032020  050377  044522  040515  EM3:    .ASCIZ  <377>"PRIMARY REGISTER READ/WRITE TEST"
032062  046777  046505  051117  EM4:    .ASCIZ  <377>"MEMORY EXTENSION READ/WRITE TEST"
032124  051777  042520  044503  EM5:    .ASCIZ  <377>/SPECIAL FUNCTION REG TEST/
032157     377  042526  052103  EM6:    .ASCIZ  <377>/VECTOR "A" FAILED TO INTERUPT/
032216  053377  041505  047524  EM7:    .ASCIZ  <377>/VECTOR "B" FAILED TO INTERUPT/
032255     377  047125  054105  EM8:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "A"/
032320  052777  042516  050130  EM9:    .ASCIZ  <377>/UNEXPECTED INTERUPT ON VECTOR "B"/
032363     377  051120  046511  EM10:   .ASCIZ  <377>/PRIMARY REGISTER ERROR/
032413     377  044514  042516  EM11:   .ASCIZ  <377>/LINE CARD STATIC TEST/
032442  051377  043505  051511  DH1:    .ASCIZ  <377>/REGISTER REFERENCED  TRAPPED FROM/
032505     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND   LINE  SEC REG  PRI REG/
032557     377  054105  042520  DH3:    .ASCIZ  <377>/EXPECTED  FOUND   PRI REG/
032612  045377  051123  050040  DH4:    .ASCIZ  <377>/JSR PC  DVSFR   EXPECTED  FOUND/
032653     377  053104  043123  DH5:    .ASCIZ  <377>/DVSFR   EXPECTED  FOUND/
032704  046777  052123  041523  DH6:    .ASCIZ  <377>/MSTSCAN  DVSFR  EXPECTED  FOUND  LINE/
        032754                          .EVEN
        000000                          SKIP=000000
032754  000000                          DATA:   0
032756  000000                          NPRLOC: 0
032760  000002                          DT1:    2
032762     006     017                          .BYTE   6,15,
032764  001262                                  SAVR1
032766     006     002                          .BYTE   6,2
032770  001264                                  SAVR2
032772  000005                          DT2:    5
032774     006     004                          .BYTE   6,4
032776  001270                                  SAVR4
033000     006     002                          .BYTE   6,2
033002  001266                                  SAVR3
033004     002     004                          .BYTE   2,4
033006  001260                                  SAVR0
033010     002     007                          .BYTE   2,7
033012  001262                                  SAVR1
033014     006     002                          .BYTE   6,2
033016  001264                                  SAVR2
033020  000003                          DT3:    3
033022     006     004                          .BYTE   6,4
033024  001272                                  SAVR5
033026     006     002                          .BYTE   6,2
033030  001270                                  SAVR4
033032     006     002                          .BYTE   6,2
033034  001266                                  SAVR3
033036  000004                          DT4:    4
033040     006     002                          .BYTE   6,2
033042  001262                                  SAVR1
033044     006     002                          .BYTE   6,2
033046  001266                                  SAVR3
033050     006     004                          .BYTE   6,4
033052  001272                                  SAVR5
033054     006     001                          .BYTE   6,1
033056  001270                                  SAVR4
033060                                  DT5:
033060  000003                                  3
```

```
033062     006      002                .BYTE   6,2
033064  001264                         SAVR2
033066     006      004                .BYTE   6,4
033070  001272                         SAVR5
033072     006      001                .BYTE   6,1
033074  001270                         SAVR4
033076  000005                 DT6:    5
033100     006      003                .BYTE   6,3
033102  001260                         SAVR0
033104     006      001                .BYTE   6,1
033106  001264                         SAVR2
033110     006      004                .BYTE   6,4
033112  001272                         SAVR5
033114     006      001                .BYTE   6,1
033116  001270                         SAVR4
033120     002      001                .BYTE   2,1
033122  001262                         SAVR1

033124                          .ERRTAB:
033124  000000                         0
033126  000000                         0
033130  000000                         0
033132  031706                         EM1
033134  032442                         DH1     ;HALT 1
033136  032760                         DT1
033140  031754                         EM2
033142  032505                         DH2     ;HALT 2
033144  032772                         DT2
033146  032020                         EM3
033150  032557                         DH3     ;HALT 3
033152  033020                         DT3
033154  032062                         EM4
033156  032505                         DH2     ;HALT 4
033160  032772                         DT2
033162  032124                         EM5
033164  032612                         DH4     ;HALT 5
033166  033036                         DT4
033170  032124                         EM5
033172  032653                         DH5     ;HALT 6
033174  033060                         DT5
033176  032157                         EM6
033200  000000                         0       ;HALT   7
033202  000000                         0
033204  032216                         EM7
033206  000000                         0       ;HALT   10
033210  000000                         0
033212  032255                         EM8
033214  000000                         0       ;HALT   11
033216  000000                         0
033220  032320                         EM9
033222  000000                         0       ;HALT 12
033224  000000                         0
033226  032363                         EM10
033230  032557                         DH3     ;HALT 13
033232  033020                         DT3
033234  032413                         EM11
```

```
033236  032704                         DH6     ;HALT   14
033240  033076                         DT6
033242                          CORMAX:
        000001                  .END
```

```
ADRCNT= 003443              617*    653*    662#
ALU    = 010000              73#    3567    3574    3609    3632    3634    3648    3658    3665    3685    3691    3713    3717
                           3719    3736    3755    3900    3928    3956    3984    4012    4040    4068    4096    4183    4307
                           4322    4337    4390
ASYNC = 004000               81#
AUTO.S  006622             1127#
BCC    = 060000              78#    4299    4311    4326    4343    4377    4428    4450    4472    4494    4516    4538    4560
                           4582    4604    4626    4648    4670    4714    4759    4804
BCCFBK  031650             5435*   5441*   5444*    5447    5451   5463#
BINWRD  003746              703*     704    741#
BIT0   = 000001              71#     884    1613    1718    1724    1746    1752    1774    1780    1802    1808    1830    1836
                           1858    1864    1883    2192    2412    2427    2600    2605    2624    2740    2805    2908    2917
                           2924    2926    2933    2935    2942    2944    2951    2953    2975    3002    3004    3011    3013
                           3020    3022    3029    3031    3069    3124    3148    3182    3184    3191    3193    3195    3197
                           3204    3206    3208    3210    3219    3223    3232    3236    3243    3245    3247    3249    3256
                           3258    3260    3262    3271    3275    3284    3288    3320    3321    3332    3340    3350    3356
                           3377    3384    3393    3406    3412    3481    3482    3503    3504    3567    3648    3656    3665
                           3685    3687    3719    3757    3898    3900    3904    3913    3914    3926    3928    3932    3941
                           3942    3954    3956    3960    3969    3970    3979    3982    3984    3988    3997    3998    4010
                           4012    4016    4025    4026    4038    4040    4044    4053    4054    4066    4068    4072    4081
                           4082    4094    4096    4100    4109    4110    4137    4138    4142    4148    4158    4161    4183
                           4187    4202    4204    4228    4230    4324    4339    4369    5097    5132    5185    5228    5271
                           5279    5317    5325    5364    5371    5376    5382    5416    5439
BIT1   = 000002              70#     884     895    1718    1724    1746    1752    1774    1780    1802    1808    1830    1836
                           1858    1864    1909    2218    2600    2624    2740    2805    2874    2906    2908    2915    2917
                           2924    2926    2933    2935    2942    2944    2951    2953    2975    3002    3004    3011    3013
                           3020    3022    3029    3031    3058    3069    3079    3180    3184    3191    3195    3197    3210
                           3217    3221    3223    3236    3243    3247    3249    3262    3269    3273    3275    3288    3320
                           3377    3392    3406    3459    3479    3481    3482    3490    3501    3503    3504    3512    3560
                           3567    3574    3608    3648    3649    3651    3657    3659    3661    3665    3666    3668    3685
                           3689    3690    3691    3692    3694    3715    3738    3757    3785    3794    3796    3844    3900
                           3904    3905    3913    3914    3928    3932    3933    3941    3942    3956    3960    3961    3969
                           3970    3984    3988    3989    3997    3998    4007    4012    4016    4017    4025    4026    4040
                           4044    4045    4053    4054    4068    4072    4073    4081    4082    4096    4100    4101    4109
                           4110    4138    4142    4148    4151    4161    4183    4204    4228    4230    4298    4309    4341
                           4371    4379    4390    4392    4414    4696    4741    4786    5097    5132    5185    5192    5228
                           5235    5271    5317    5364    5372    5376    5382    5394    5397    5421
BIT10 = 002000               61#     884    1535    1743    2039    2600    2624    2913    2931    2940    2949    2972    3027
                           3078    3179    3183    3244    3248    3257    3261    3270    3274    3283    3287    3323    3336
                           3347    4042    4051    4070    4079    4098    4107    4133    4198    4223    4876    4883    5368
                           5379
BIT11 = 004000               60#     884    1771    2065    2600    2624    2720    2786    2940    2972    3000    3009    3018
                           3027    3179    3183    3323    3336    3347    3930    3939    3958    3967    3986    3995    4014
                           4023    4042    4051    4070    4079    4098    4107    4133    4159    4198    4223
BIT12 = 010000               59#      73      75      77      79    1561    1799    3147    3951    4874    4883    5394    5397
BIT13 = 020000               58#      74      75      78      79    1587    1827    3135    3923    4916    4924    4959
BIT14 = 040000               57#      76      77      78      79     524    1855    3123    4395    4445    4448    5394    5397
BIT15 = 100000               56#    1642    1646    1652    1653    3111    3895    4914    4924    4959    5131    5147
BIT2   = 000004              69#     451     884    1320    1935    2244    2600    2624    3058    3136    3148    3178    3182
                           3191    3204    3217    3230    3243    3256    3269    3282    3459    3479    3481    3490    3492
                           3501    3503    3512    3514    3560    3567    3574    3608    3609    3632    3634    3648    3658
                           3665    3685    3691    3713    3717    3719    3736    3738    3755    3757    3785    3844    3864
                           3900    3928    3956    3984    4012    4035    4040    4068    4096    4183    4246    4298    4307
                           4322    4337    4379    4390    4414    4696    4741    4786    5097    5132    5192    5235    5279
                           5325    5376
BIT3   = 000010              68#     884    1346    1613    1961    2128    2270    2405    2411    2420    2426    2600    2624
```

```
                           2865    2871    2903    2971    2999    3056    3058    3059    3069    3079    3110    3112    3122
                           3124    3134    3136    3146    3148    3172    3319    3376    3433    3488    3558    3567    3606
                           3609    3631    3632    3634    3648    3658    3665    3681    3685    3712    3713    3717    3735
                           3736    3754    3755    3783    3841    3894    3900    3910    3922    3928    3938    3950    3956
                           3966    3978    3984    3994    4006    4012    4022    4034    4040    4050    4062    4063    4068
                           4078    4090    4096    4106    4134    4181    4183    4296    4307    4321    4322    4336    4337
                           4364    4373    4418    4440    4462    4484    4506    4528    4550    4572    4594    4616    4638
                           4660    4701    4746    4791    5096    5126    5179    5222    5265    5311    5357    5376    5391
BIT4   = 000020              67#    1372    1718    1724    1746    1752    1774    1780    1802    1808    1830    1836    1858
                           1864    2296    2667    2678    2689    2740    2805    3178    3182    3321    3332    3344    3384
                           3461    3468    3471    3490    3499    3501    3512    3560    3567    3574    3608    3609    3632
                           3648    3656    3665    3685    3687    3689    3691    3713    3715    3736    3738    3755    3757
                           3785    3844    3864    3898    3900    3926    3928    3954    3956    3982    3984    4010    4012
                           4038    4040    4066    4068    4091    4094    4096    4183    4185    4187    4246    4307    4309
                           4322    4324    4337    4339    4341    4369    4371    4390    4392    5097    5132    5145    5155
                           5185    5192    5228    5235    5271    5279    5317    5325    5360    5364    5416    5421
BIT5   = 000040              66#    1398    1718    1724    1746    1752    1774    1780    1802    1808    1830    1836    1858
                           1864    2322    2667    2678    2740    2805    3191    3195    3204    3208    3217    3221    3230
                           3234    3461    3471    3479    3481    3482    3490    3492    3493    3501    3512    3560    3608
                           3648    3656    3685    3687    3689    3715    3844    3898    3926    3954    3982    4010    4038
                           4066    4094    4183    4185    4187    4298    4309    4324    4339    4341    4369    4371    4379
                           4392    4414    4696    4741    4786    5097    5132    5185    5228    5271    5279    5317    5325
                           5360    5364    5416    5421
BIT6   = 000100              65#    1195    1424    2348    3461    3471    3560    3608    3656    3687    3689    3715    3785
                           3794    3796    3864    4185    4187    4246    4298    4309    4324    4339    4341    4379    4392
                           4414    4696    4741    4786    4836    4843    4960    4972    4996    5021    5046    5071    5097
                           5394    5397
BIT7   = 000200              64#     518     765     922     943    1195    1453    1457    1463    1464    1671    2374    3068
                           3243    3247    3256    3260    3269    3273    3282    3286    3407    3413    3461    3471    3479
                           3490    3501    3503    3504    3510    3512    3514    3515    3560    3608    3656    3687    3689
                           3715    3844    3898    3926    3954    3982    4010    4038    4066    4094    4185    4187    4298
                           4309    4324    4339    4341    4369    4371    4379    4392    4414    4696    4741    4786    4834
                           4843    4960    4972    4996    5021    5046    5071    5097    5132    5185    5192    5228    5235
                           5271    5279    5317    5325    5364    5416    5421
BIT8   = 000400              63#     884     901    1483    1670    1672    1987    2411    2426    2600    2624    2903    2904
                           2913    2922    2931    2940    2972    3009    3055    3205    3209    3231    3235    3257    3261
                           3283    3287    3323    3336    3347    3378    3389    3477    4014    4023    4070    4079    4159
                           4185    5271    5290    5317    5337    5364
BIT9   = 001000              62#     884    1195    1200    1457    1509    1646    1715    2013    2600    2624    2931    2940
                           2949    2972    3018    3027    3218    3222    3231    3235    3270    3274    3283    3287    3323
                           3336    3347    3378    3389    3407    3986    3995    4014    4023    4098    4107    4159    4198
                           4223    4366    4376    4394    4834    4836    4843    4914    4916    4919    4924    4958    4960
                           4996    5021    5046    5071    5271    5317    5364    5368    5379
BRB    = 070000              79#    2972    3000    3009    3018    3027    3192    3196    3205    3209    3218    3222    3231
                           3235    3244    3248    3257    3261    3270    3274    3283    3287    4133    4159    4198    4223
BRW     003014              457     546#
BRX     003016              458     547#
CALBCC  031652             4419*    4432   4441*    4454   4463*    4476   4485*    4498   4507*    4520   4529*    4542   4551*
                           4564    4573*    4586   4595*    4608   4617*    4630   4639*    4652   4661*    4674   4699*    4704
                           4705     4718   4744*    4749    4750    4763   4789*    4794    4795    4808   5457*   5464#
CHRCNT  003744              701*     705    721*    739#     740
CLKX    001242              150#
CLK.A   001412              250#    1045
CLK.B   001413              251#    1050
CLK.C   001414              252#    1055
CLK.D   001415              253#    1060
```

```
CNVRT = 104411          209#    476     478     48Ø     482     8Ø2     8Ø4     86Ø     917
CONVPT= 10441Ø          207#    416     818
CORMAX  Ø33242          5581#   5582
CRC.CC= 1Ø2Ø1Ø          4785    5467#
CRC16 = 12ØØØ1          4412    474Ø    5466#
CREAM   ØØ13Ø6          171#    385*    992*    993     995*    1ØØØ    1ØØ1*   1ØØ2    1ØØ5*
CSRMAP  ØØ6624          41Ø     1129#
CYCLE   ØØ5666          46Ø     496     497     982#
DATA    Ø32754          5487#
DATABP  ØØ4276          791*    794     816     819#
DATACL= 1Ø4416          219#
DATAHD  ØØ4264          79Ø*    812     815#
DELAY = 1Ø4414          215#    51Ø2
DEVADR  ØØ344Ø          615*    65Ø     66Ø#
DH1     Ø32442          5486#   5546
DH2     Ø325Ø5          5486#   5549    5555
DH3     Ø32557          5486#   5552    5576
DH4     Ø32612          5486#   5558
DH5     Ø32653          5486#   5561
DH6     Ø327Ø4          5486#   5579
DT1     Ø3276Ø          5489#   5547
DT2     Ø32772          5494#   555Ø    5556
DT3     Ø33Ø2Ø          55Ø5#   5553    5577
DT4     Ø33Ø36          5512#   5559
DT5     Ø33Ø6Ø          5521#   5562
DT6     Ø33Ø76          5529#   558Ø
DVACTV  ØØ13ØØ          165#    432*    433     982     987     1161*   1167*   1168*   1172    1191
DVCRØØ  ØØ15ØØ          281#
DVCRØ1  ØØ1524          292#
DVCRØ2  ØØ155Ø          3Ø3#
DVCRØ3  ØØ1574          314#
DVCRØ4  ØØ162Ø          325#
DVCRØ5  ØØ1644          336#
DVCRØ6  ØØ167Ø          347#
DVCRØ7  ØØ1714          358#
DVLCR   ØØ137Ø          233#    9Ø1*    9Ø2     1Ø21*   1Ø22*   1Ø23    127Ø    1292    1714    1742    177Ø    1798    1826
                        1854    2666    2677    29Ø2    2976    2998    3298    3326    3339    335Ø    3355    338Ø    3393
                        3397    34Ø8    3414    39Ø3    3912    3931    394Ø    3959    3968    3987    3996    4Ø15    4Ø24
                        4Ø43    4Ø52    4Ø71    4Ø8Ø    4Ø99    41Ø8    4136    4147    416Ø    42Ø3    4229    5369    538Ø
DVNSR   ØØ14Ø2          238#    1Ø31*   1Ø32*   1Ø33    1266    1288    2168    51Ø6    5129
DVNUM   ØØ13Ø1          166#    381     486     1133*   1154*   1155    1162    1164
DVRIC   ØØ1366          232#    1Ø19*   1Ø2Ø*   1Ø21    1673    1693    3116    3128    314Ø    3152    3465    3523    3533
                        3564    3571    3578    3613    3625    3639    3652    3662    3669    3683    37Ø3    3723    3789
                        38ØØ    3849    43Ø4    4316    433Ø    4348    4381    4433    4455    4477    4499    4521    4543
                        4565    4587    46Ø9    4631    4653    4675    4719    4764    48Ø9    518Ø    5223    5266    5312
                        5358
DVRLVL  ØØ1354          227#    1Ø36*   1Ø37*   1Ø38    5473*
DVRVEC  ØØ1352          226#    5Ø3     1ØØ7*   1Ø36    5471*
DVSCR   ØØ1362          23Ø#    5ØØ     878*    891*    895*    1ØØ6*   1Ø17    1222    1253    1278    1319    1345    1371
                        1397    1423    1451    1482    15Ø8    1534    156Ø    1586    1612    164Ø    1669    2128*   24Ø3
                        2672*   2717    2727    2783    2791*   2792    2865*   2871*   2874    29Ø3*   2922*   2971*   2999*
                        3Ø55*   3Ø56*   3Ø62    3Ø72    3Ø83    311Ø*   3122*   3134*   3146*   3172*   3319*   3376*   34Ø7*
                        3413*   3433*   3558*   36Ø6*   3631*   3649*   3651*   3657*   3659*   3661*   3666*   3668*   3681*
                        3684    3712*   3735*   3754*   3783*   3841*   3894*   391Ø*   3922*   3938*   395Ø*   3966*   3978*
                        3994*   4ØØ6*   4Ø22*   4Ø34*   4Ø5Ø*   4Ø62*   4Ø78*   4Ø9Ø*   41Ø6*   4134*   4181*   4296*   4321*
                        4336*   4364*   4418*   444Ø*   4462*   4484*   45Ø6*   4528*   455Ø*   4572*   4594*   4616*   4638*
```

```
                        466Ø*   47Ø1*   4746*   4791*   4834*   4836*   4838*   4843*   485Ø*   4853*   4874*   4876*   4878*
                        4883*   489Ø*   4893*   4914*   4916*   4918*   4919*   4924*   4931*   4934*   4958*   4959*   496Ø*
                        4972*   4996*   4999*   5Ø21*   5Ø24*   5Ø46*   5Ø49*   5Ø71*   5Ø75*   5Ø93    5Ø96*   51ØØ    51Ø7
                        5126*   5134    5179*   5222*   5265*   5311*   5357*   536Ø*   5391*
DVSCRH  ØØ1364          231#    1Ø17*   1Ø18*   1Ø19
DVSFR   ØØ14ØØ          237#    1Ø29*   1Ø3Ø*   1Ø31    2129    2867*   2868    2877    29Ø1    2972*   2974    2997    3Ø57
                        31Ø9    3173    3321*   3323*   3325    3332*   3336*   3338    3344*   3347*   3354    3379*   3386*
                        3391*   3434    3559    36Ø7    3784    3843    3896*   3898*   39ØØ*   39Ø2*   3911*   3924*   3926*
                        3928*   393Ø*   3939*   3952*   3954*   3956*   3958*   3967*   398Ø*   3982*   3984*   3986*   3995*
                        4ØØ8*   4Ø1Ø*   4Ø12*   4Ø14*   4Ø23*   4Ø36*   4Ø38*   4Ø4Ø*   4Ø42*   4Ø51*   4Ø64*   4Ø66*   4Ø68*
                        4Ø7Ø*   4Ø79*   4Ø92*   4Ø94*   4Ø96*   4Ø98*   41Ø7*   4135*   4144*   4146*   4159*   4164    4182
                        4198*   4223*   4227    4297    4365    4413    4697    4742    4787    5Ø97*   513Ø    5181    5224
                        5267    5313    5359    5394*   5397*
DVSRA   ØØ1376          236#    882     1Ø27*   1Ø28*   1Ø29    1256*   2Ø91    2448    25Ø8    2552    2596    2676*   2719*
                        2785*   3827    3895*   3923*   3951*   3979*   4ØØ7*   4Ø35*   4Ø63*   4Ø91*   4143*   4193*   42Ø8
                        4255    4366*   4373*   5182*   5189*   5225*   5232*   5268*   5275*   5283    5289*   529Ø    5314*
                        5321*   5329    5336*   5337    5361*   5413*   5418*   5423*
DVSRS   ØØ1372          234#    881     1Ø23*   1Ø24*   1Ø25    1255*   1882    19Ø8    1934    196Ø    1986    2Ø12    2Ø38
                        2Ø64    2418    2449*   25Ø9*   2553*   2569*   2597*   26ØØ*   26Ø6*   26Ø9*   2615    2624*   2633*
                        2673*   2718*   2784*   3825    3839*   389Ø*   4191*   4253
DVSRSH  ØØ1374          235#    1Ø25*   1Ø26*   1Ø27    245Ø*   251Ø*   252Ø*   2554*   257Ø*   26Ø7*   2618    2634*   2675*
                        3826    4192*   4254
DVTLVL  ØØ136Ø          229#    1Ø4Ø*   1Ø41*   5474*
DVTRØØ  ØØ15Ø2          282#
DVTRØ1  ØØ1526          293#
DVTRØ2  ØØ1552          3Ø4#
DVTRØ3  ØØ1576          315#
DVTRØ4  ØØ1622          326#
DVTRØ5  ØØ1646          337#
DVTRØ6  ØØ1672          348#
DVTRØ7  ØØ1716          359#
DVTVEC  ØØ1356          228#    1Ø38*   1Ø39*   1Ø4Ø    5472*
DV.END  ØØ174Ø          369#    993     1ØØ2    1131
DV.MAP  ØØ15ØØ          171     28Ø#    385     412     995     1ØØ5    1129    1134    1183
DVØØ.A  ØØ15Ø4          283#
DVØØ.B  ØØ151Ø          285#
DVØØ.C  ØØ1514          287#
DVØØ.D  ØØ152Ø          289#
DVØ1.A  ØØ153Ø          294#
DVØ1.B  ØØ1534          296#
DVØ1.C  ØØ154Ø          298#
DVØ1.D  ØØ1544          3ØØ#
DVØ2.A  ØØ1554          3Ø5#
DVØ2.B  ØØ156Ø          3Ø7#
DVØ2.C  ØØ1564          3Ø9#
DVØ2.D  ØØ157Ø          311#
DVØ3.A  ØØ16ØØ          316#
DVØ3.B  ØØ16Ø4          318#
DVØ3.C  ØØ161Ø          32Ø#
DVØ3.D  ØØ1614          322#
DVØ4.A  ØØ1624          327#
DVØ4.B  ØØ163Ø          329#
DVØ4.C  ØØ1634          331#
DVØ4.D  ØØ164Ø          333#
DVØ5.A  ØØ165Ø          338#
DVØ5.B  ØØ1654          34Ø#
```

```
DV05.C  001660      342#
DV05.D  001664      344#
DV06.A  001674      349#
DV06.B  001700      351#
DV06.C  001704      353#
DV06.D  001710      355#
DV07.A  001720      360#
DV07.B  001724      362#
DV07.C  001730      364#
DV07.D  001734      366#
EM1     031706     5486#   5545
EM10    032363     5486#   5575
EM11    032413     5486#   5578
EM2     031754     5486#   5548
EM3     032020     5486#   5551
EM4     032062     5486#   5554
EM5     032124     5486#   5557    5560
EM6     032157     5486#   5563
EM7     032216     5486#   5566
EM8     032255     5486#   5569
EM9     032320     5486#   5572
ERRCNT  001232      142#    387*    509     827*
ERRFLG  001311      177#    383*    471*    538*    779*    792     806*    861*
ERRMSG  004252      789*    807     810#
ERTAB0  004366      804     836#
EXIT  = 000205       81#   5406
EXITER  004322      822     827#
FIX.00  006516     1046    1051    1056    1061    1095#
HALTS   004302      775     821#
HILIM   003436      614*    641     659#
ICOUNT  001222      138#    536     541*   2546*
INBUF   005520      584     620     967#
INIFLG  001310      176#    392     407*
INSTER= 104404      199#    635
INSTR = 104403      197#   1068
INSTR2  003236      591     603#
LIGHT   000174      110#    402
LIGHTS  001200      121#    402*    473*
LIMITS  003364      630     641#
LOBITS  003442      616*    645     661#    662
LOCK    001220      137#    540*    554     556     798    1220*   1251*   1279*   2445*   2461*   2472*   2504*   2519*
                   2548*   2595*   2664*   3053*   3067*   3077*   3170*   3189*   3202*   3215*   3228*   3241*   3254*
                   3267*   3280*   3431*   3487*   3498*   3509*   3604*   3622*   3646*   3679*   3710*   3733*   3752*
                   4416*   4438*   4460*   4482*   4504*   4526*   4548*   4570*   4592*   4614*   4636*   4658*   5124*
                   5177*   5220*   5263*   5309*
LOGICA  002560      107     490#
LOKFLG  001312      178#
LOLIM   003434      613*    643     658#
LPCNT   001224      139#    535*    536     539*
LRC8  = 000200     4695    5465#
LSTERR  001234      143#    388*    470*    522*    776     778*    862*
L.DATA  031420     4424    4446    4468    4490    4512    4534    4556    4578    4600    4622    4644    4666    4710
                   4755    4800    5413#
L00.03  001416      255#   1008*   1043
L04.07  001420      256#   1010*   1048
L08.11  001422      257#   1012*   1053
```

```
L12.15  001424      258#   1014*   1058
MAR17   014224     2596#
MASKX   001244      151#
MASK.A  001406      245#   1044
MASK.B  001407      246#   1049
MASK.C  001410      247#   1054
MASK.D  001411      248#   1059
MASTEK  005400      800     958#
MCRLF   005104      569     692     796     797     805     946     958#   1067    1085
MCSRX   005330      475     958#
MDATA   005624      719     729     971#
MEMEXT  014620     2671    2687    2697#
MEPASS  005145      474     958#
MERRPC  005454      803     958#
MERRX   005355      481     958#
MERR2   005174      958#    984    1174
MERR3   005243      429     958#
MLOCK   005301      453     958#
MNEW    005402      424     958#
MPASSX  005344      479     958#
MPFAIL  005107      859     958#
MQM     005100      599     958#   1090
MR      005171      461     958#
MRESET= 004000       81#    878     891    2791
MSTCLR= 104412      211#    863    2864    2900    2970    2996    3054    3108    3121    3133    3145    3171    3317
                   3375    3432    3557    3605    3623    3680    3702    3711    3734    3753    3782    3822    3893
                   3909    3921    3937    3949    3965    3977    3993    4005    4021    4033    4049    4061    4077
                   4089    4105    4131    4180    4295    4320    4335    4363    4417    4439    4461    4483    4505
                   4527    4549    4571    4593    4615    4637    4659    4700    4745    4790    4828    4868    4908
                   4951    4991    5016    5041    5066    5092    5125    5178    5221    5264    5310    5356
MTITLE  001000      119#    406
MTSTN   005366      801     958#   1069
MTSTPC  005267      958#
MVECX   005336      477     958#
NEXT    001216      136#    542     832    1219*   1250*   1318*   1344*   1370*   1396*   1422*   1450*   1481*   1507*
                   1533*   1559*   1585*   1611*   1639*   1668*   1692*   1713*   1741*   1769*   1797*   1825*   1853*
                   1881*   1907*   1933*   1959*   1985*   2011*   2037*   2063*   2090*   2127*   2167*   2190*   2216*
                   2242*   2268*   2294*   2320*   2346*   2372*   2402*   2444*   2503*   2547*   2663*   2715*   2781*
                   2863*   2899*   2969*   2995*   3052*   3107*   3169*   3316*   3374*   3430*   3556*   3600*   3781*
                   3821*   3889*   4130*   4179*   4294*   4362*   4411*   4694*   4739*   4784*   4826*   4866*   4906*
                   4949*   4989*   5014*   5039*   5064*   5091*   5123*   5176*   5219*   5262*   5308*   5355*
NOLIST= ****** U      1
NO.ATR  031676     4830    4847    4870    4887    4910    4928    4993    5018    5043    5477#
NO.BTR  031702     4831    4848    4871    4888    4911    4929    4994    5019    5044    5069    5481#
NPR   = 040000       76#   5190    5233    5281    5327    5366
NPRLOC  032756     5182    5187*   5195    5200*   5225    5230*   5238    5244*   5268    5273*   5276*   5284    5314
                   5319*   5322*   5330    5488#
PARAM = 104405      201#   1070
PARAM1  003304      619#    636
PARBIT= 040000       81#   1114
PARERR  003360      622     624     626     635#    642     644     646
PASCNT  001230      141#    382*    472*    473     506
PC    =%000007       48#    410*    491*    521*    768*    947*   1046*   1051*   1056*   1061*   1077    1078*   1117*
                   1209*
PERFOR= 004537       81#   3860    4243    5142
PFTAB   004470      860     866#
```

```
POPR0 = 012600             55#    826
POP1SP= 005726             55#
POP2SP= 022626             55#    544
PS    = 177776             53#    378*    447*   1194*   2716*   2782*   4827*   4833*   4867*   4873*   4907*   4913*   4950*
                         4961*   4990*   4997*   5015*   5022*   5040*   5047*   5065*   5072*
PUSHR0= 010046             55#    823
PUSH1S= 005746             55#
PUSH2S= 024646             55#
QV.FLG  001313            179#    384*    485*    533
RAM   = 020000             74#   3842    3858    3896    3924    3952    3980    4008    4036    4064    4092    4144    4185
                         4214    4367    4374    5153    5183    5226    5269    5277    5315    5323    5362    5414    5419
                         5424
RAMCLR= 104413            213#    864
RESREG  004300            817     820#
RESTAR  004414            847     853#
RESTRT  002572            484     488     496#
RESV16  001404            239#   1033*   1034*   2191    2217    2243    2269    2295    2321    2347    2373    2594*   2613*
                         2626    2629*
RES05 = 104407            205#    820
RETURN  001214            135#    390*    460*    462     496*    542*    545     832*    834     865    1084*   1092*   1093
ROMCLK= 104415            217#   2872    3061    3071    3082    3115    3127    3139    3151    3294    3322    3334    3345
                         3387    3464    3520    3522    3529    3531    3562    3568    3570    3575    3577    3610    3612
                         3633    3635    3637    3686    3688    3714    3716    3718    3720    3722    3737    3739    3741
                         3756    3758    3760    3787    3795    3797    3799    3846    3848    3865    3897    3899    3901
                         3925    3927    3929    3953    3955    3957    3981    3983    3985    4009    4011    4013    4037
                         4039    4041    4065    4067    4069    4093    4095    4097    4145    4197    4216    4218    4222
                         4238    4247    4300    4302    4308    4310    4312    4314    4323    4325    4327    4329    4338
                         4340    4342    4344    4346    4368    4370    4372    4375    4378    4380    4391    4393    4429
                         4431    4451    4453    4473    4475    4495    4497    4517    4519    4539    4541    4561    4563
                         4583    4585    4605    4607    4627    4629    4649    4651    4671    4673    4715    4717    4760
                         4762    4805    4807    5099    5133    5154    5184    5186    5191    5193    5227    5229    5234
                         5236    5270    5272    5278    5280    5282    5316    5318    5324    5326    5328    5363    5365
                         5367    5378    5395    5398    5415    5417    5420    5422    5425
RUN     001304            169#    386*    987     990*    991*    998*    999*
R0     =%000000            41#    425*    433*    434*    436*    438*    440     441     523     529*    543*    676     681*
                          693     706*    710*    720     736*    824*    869     870*    871*    873*    899     900*    905*
                          908*   1000*   1006    1007    1008    1009    1010    1011    1012    1013    1014    1015    1016*
                         1022    1024    1030    1032    1034    1037    1039    1041    1043*   1048*   1053*   1058*   1076*
                         1077    1080    1082    1084    1087    1088    1095    1114    1175*   1184*   1186*   1188    1196*
                         1197*   1221*   1230*   1252*   1258    1265    1268    1272    1277*   1281    1287    1290    1294
                         2446*   2449    2487*   2488    2505*   2509    2530*   2531    2549*   2553    2562*   2563    2566*
                         2569    2582*   2583    2603*   2606    2615    2633    2640*   2641    2665*   2673    2692*   2693
                         2726*   2901*   2904*   2905    2913*   2914    2923    2931*   2932    2940*   2941    2949*   2950
                         2997*   3000*   3001    3009*   3010    3018*   3019    3027*   3028    3057*   3058*   3060    3069*
                         3070    3079*   3081    3109*   3112*   3114    3124*   3126    3136*   3138    3148*   3150    3173*
                         3291*   3293*   3295*   3296    3318*   3330*   3434*   3463*   3517*   3519*   3521*   3527*   3528*
                         3530*   3559*   3561*   3567*   3569*   3574*   3576*   3607*   3609*   3611*   3632*   3634*   3636*
                         3648*   3650*   3656*   3658*   3660*   3665*   3667*   3685*   3687*   3689*   3691*   3693*   3713*
                         3715*   3717*   3719*   3721*   3736*   3738*   3740*   3755*   3757*   3759*   3784*   3786*   3794*
                         3796*   3798*   3823*   3829    3832    3833    3835*   3843*   3845*   3847*   3859    3862*   3864*
                         4182*   4196*   4214*   4215*   4217*   4221*   4237*   4242    4245*   4246*   4251*   4257    4263
                         4264    4266*   4297*   4299*   4301*   4307*   4309*   4311*   4313*   4322*   4324*   4326*   4328*
                         4337*   4339*   4341*   4343*   4345*   4365*   4367*   4369*   4371*   4374*   4377*   4379*   4390*
                         4392*   4413*   4428*   4430*   4450*   4452*   4472*   4474*   4494*   4496*   4516*   4518*   4538*
                         4540*   4560*   4562*   4582*   4584*   4604*   4606*   4626*   4628*   4648*   4650*   4670*   4672*
                         4697*   4714*   4716*   4742*   4759*   4761*   4787*   4804*   4806*   4953*   4964    4968*   4969
```

```
                         4974*   5098*   5103*   5127*   5145    5148*   5403*   5405*   5414*   5416*   5419*   5421*   5424*
                         5429    5436*   5438*   5439    5442*   5443*   5444    5447*   5450*   5452*   5454    5460*
R1     =%000001            42#    437*    438     439*    440     487*    491     675     682*    694     698*    700     701
                          702     703     735*    879     881*    884*    887*   1044*   1049*   1054*   1059*   1099*   1104*
                         1109*   1112*   1135*   1137    1139    1141    1144    1157*   1158    1164*   1165    1169*   1185*
                         1186    1187*   1188    1189    1222*   1225    1229*   2447*   2450    2483*   2484    2486*   2506*
                         2510    2514*   2515    2517*   2520    2527*   2528    2550*   2554    2556*   2557    2559*   2567*
                         2570    2576*   2577    2581*   2604*   2607    2618    2634    2636*   2637    2639*   2674*   2675
                         3292*   3518*   3683*   3695    3742    3761    3824*   3830    3836*   3837    4190*   4192    4195
                         4215    4220    4239*   4240    4255*   4259    4698*   4702    4703    4723*   4743*   4747    4748
                         4768*   4788*   4792    4793    4813*   5149*   5396*   5399*   5403    5430    5448*   5449    5450
                         5459*
R2     =%000002            43#    674     683*   1045*   1050*   1055*   1060*   1100*   1105*   1110*   1113*   1116*   1129*
                         1130*   1131    1134*   1144*   1145    1146*   1147*   1148*   1149*   1150*   1151*   1152*   1153*
                         1160*   1183*   1195*   1199*   1200*   1201*   1202*   1204*   1205*   1235*   1257*   1269*   1273*
                         1274*   1280*   1291*   1295*   1296*   2448*   2452*   2453    2462*   2463    2473*   2474    2482*
                         2508*   2511*   2521    2552*   2555*   2571    2596*   2598*   2608*   2610    2614    2635*   2666*
                         2721*   2723*   2787*   2789*   2905*   2914*   2923*   2932*   2941*   2950*   2974*   3001*   3010*
                         3019*   3028*   3060*   3070*   3081*   3114*   3126*   3138*   3150*   3177*   3181*   3190*   3194*
                         3203*   3207*   3216*   3220*   3229*   3233*   3242*   3246*   3255*   3259*   3268*   3272*   3281*
                         3285*   3292    3293    3295    3297    3302*   3325*   3338*   3349*   3352*   3354*   3378*   3379
                         3384*   3386    3389*   3391    3458*   3459*   3461*   3462*   3463    3480*   3491*   3502*   3513*
                         3518    3519    3528    3536*   3560*   3561    3569    3576    3608*   3611    3636    3650    3660
                         3667    3693    3721    3740    3759    3785*   3786    3798    3825*   3829*   3844*   3847    4133*
                         4135    4146    4164*   4183*   4201    4211*   4217    4226    4227*   4234*   4253*   4258*   4298*
                         4301    4313    4328    4345    4414*   4430    4452    4474    4496    4518    4540    4562    4584
                         4606    4628    4650    4672    4696*   4716    4741*   4761    4786*   4806    5130*   5132*   5152*
                         5153*   5181*   5183*   5185*   5190*   5192*   5224*   5226*   5228*   5233*   5235*   5267*   5269*
                         5271*   5277*   5279*   5281*   5313*   5315*   5317*   5323*   5325*   5327*   5359*   5362*   5364*
                         5366*   5368*   5376*   5379*   5431    5449*   5451*   5452    5458*
R3     =%000003            44#    578     585*    595*    598*    600     604*    673     684*    695     707*    708*    709*
                          710     719*    720*    725*    728*    734*   1095*   1096*   1097    1102    1107    1253*   1254*
                         1259*   1260    1265    1266    1268    1270    1272    1278*   1282    1287    1288    1290    1292
                         1294    1319*   1321*   1322    1326*   1327    1345*   1347*   1348    1352*   1353    1371*   1373*
                         1374    1378*   1379    1397*   1399*   1400    1404*   1405    1423*   1425*   1426    1430*   1431
                         1451*   1453*   1454    1458*   1459    1464*   1465    1482*   1484*   1485    1489*   1490    1508*
                         1510*   1511    1515*   1516    1534*   1536*   1537    1541*   1542    1560*   1562*   1563    1567*
                         1568    1586*   1588*   1589    1593*   1594    1612*   1614*   1615    1619*   1620    1640*   1642*
                         1643    1647*   1648    1653*   1654    1669*   1671*   1672*   1674    1678*   1693*   1696*   1697
                         1714*   1716*   1717    1722*   1723    1742*   1744*   1745    1750*   1751    1770*   1772*   1773
                         1778*   1779    1798*   1800*   1801    1806*   1807    1826*   1828*   1829    1834*   1835    1854*
                         1856*   1857    1862*   1863    1882*   1884*   1885    1889*   1890    1908*   1910*   1911    1915*
                         1916    1934*   1936*   1937    1941*   1942    1960*   1962*   1963    1967*   1968    1986*   1988*
                         1989    1993*   1994    2012*   2014*   2015    2019*   2020    2038*   2040*   2041    2045*   2046
                         2064*   2066*   2067    2071*   2072    2091*   2093*   2094    2099*   2100    2105*   2106    2113*
                         2129*   2131*   2132    2137*   2138    2143*   2144    2151*   2168*   2170    2173*   2174    2191*
                         2193*   2194    2198*   2199    2217*   2219*   2220    2224*   2225    2243*   2245*   2246    2250*
                         2251    2269*   2271*   2272    2276*   2277    2295*   2297*   2298    2302*   2303    2321*   2323*
                         2324    2328*   2329    2347*   2349*   2350    2354*   2355    2373*   2375*   2376    2380*   2381
                         2403*   2404*   2406*   2407    2412*   2413    2417*   2418*   2419*   2421*   2422    2427*   2428
                         2453*   2454    2463*   2464    2474*   2475    2521*   2522    2571*   2572    2614*   2621    2677*
                         2678*   2680    2717*   2720*   2722*   2727*   2729    2733    2735    2738    2739    2744    2745
                         2748    2749    2752    2753    2758    2759    2762    2763    2783*   2786*   2788*   2792*   2794
                         2798    2800    2803    2804    2809    2810    2813    2814    2817    2818    2823    2824    2827
                         2828    2902*   2907    2916    2925    2934    2943    2952    2998*   3003    3012    3021    3030
                         3296*   3460*   3462    3468*   3469    3471    3478*   3479*   3489*   3490*   3500*   3501*   3511*
```

```
                          3512*   3521    3530    3684*   3690*   3692*   3694*   3793*   3806*   3826*   3830*   3842*   3845
                          3854*   3855    3858*   4132*   4143    4149    4156*   4185*   4194*   4195*   4196    4219*   4220*
                          4221    4254*   4257*   4382*   4385*   4424*   4446*   4468*   4490*   4512*   4534*   4556*   4578*
                          4600*   4622*   4644*   4666*   4710*   4755*   4800*   4952*   4962*   5129*   5137    5180*   5194
                          5223*   5237    5266*   5312*   5358*   5390    5392*   5400*   5402*   5413    5418    5423    5426*
R4      =%000004            45#    579     584*    588*    589*    590     597*    601     603*    611     620*    621     623
                           625     627*    628     629     650*    651*    655*    672     685*    696     704*    707     712*
                           714*    716*    733*    783*    784*    785*    786*    787*    788*    789     790     791     880
                           882*    883*    886*   1260*   1261    1282*   1283    1322*   1323    1327*   1329    1348*   1349
                          1353*   1355    1374*   1375    1379*   1381    1400*   1401    1405*   1407    1426*   1427    1431*
                          1433    1454*   1459*   1460    1465*   1466    1485*   1486    1490*   1492    1511*   1512    1516*
                          1518    1537*   1538    1542*   1544    1563*   1564    1568*   1570    1589*   1590    1594*   1596
                          1615*   1616    1620*   1622    1643*   1648*   1649    1654*   1655    1674*   1675    1697*   1698
                          1717*   1718*   1719    1723*   1724*   1726    1745*   1746*   1747    1751*   1752*   1754    1773*
                          1774*   1775    1779*   1780*   1782    1801*   1802*   1803    1807*   1808*   1810    1829*   1830*
                          1831    1835*   1836*   1838    1857*   1858*   1859    1863*   1864*   1866    1885*   1886    1890*
                          1892    1911*   1912    1916*   1918    1937*   1938    1942*   1944    1963*   1964    1968*   1970
                          1989*   1990    1994*   1996    2015*   2016    2020*   2022    2041*   2042    2046*   2048    2067*
                          2068    2072*   2074    2094*   2095    2100*   2101    2106*   2107    2132*   2133    2138*   2139
                          2144*   2145    2171*   2172*   2173    2174*   2175    2194*   2195    2199*   2201    2220*   2221
                          2225*   2227    2246*   2247    2251*   2253    2272*   2273    2277*   2279    2298*   2299    2303*
                          2305    2324*   2325    2329*   2331    2350*   2351    2355*   2357    2376*   2377    2381*   2383
                          2407*   2408    2413*   2414    2422*   2423    2428*   2429    2451*   2452    2454    2458*   2460*
                          2462    2464    2469*   2471*   2473    2475    2480*   2507*   2511    2512*   2518*   2522    2526*
                          2551*   2555    2560*   2568*   2572    2579*   2605*   2608    2621    2631*   2635    2667*   2672
                          2680    2689*   2729*   2730    2735*   2739*   2740*   2741    2745*   2749*   2753*   2755    2763*
                          2765    2794*   2795    2800*   2804*   2805*   2806    2810*   2814*   2818*   2820    2828*   2830
                          2907*   2908*   2910    2916*   2917*   2919    2925*   2926*   2928    2934*   2935*   2937    2943*
                          2944*   2946    2952*   2953*   2955    2976*   2977    3003*   3004*   3006    3012*   3013*   3015
                          3021*   3022*   3024    3030*   3031*   3033    3062*   3063    3072*   3073    3083*   3084    3116*
                          3117    3128*   3129    3140*   3141    3152*   3153    3298*   3299    3326*   3327    3339*   3341
                          3355*   3357    3380*   3381    3388*   3395*   3397*   3398    3408*   3409    3414*   3415    3465*
                          3523*   3524    3533*   3564*   3571*   3578*   3613*   3615    3625*   3639*   3652*   3653    3662*
                          3669*   3670    3695*   3696    3703*   3723*   3725    3742*   3744    3761*   3763    3789*   3800*
                          3801    3828*   3831    3832    3849*   3850    3903*   3904*   3906    3912*   3913*   3915    3931*
                          3932*   3934    3940*   3941*   3943    3959*   3960*   3962    3968*   3969*   3971    3987*   3988*
                          3990    3996*   3997*   3999    4015*   4016*   4018    4024*   4025*   4027    4043*   4044*   4046
                          4052*   4053*   4055    4071*   4072*   4074    4080*   4081*   4083    4099*   4100*   4102    4108*
                          4109*   4111    4136*   4138*   4139    4147*   4148*   4152    4160*   4161*   4162    4187*   4200
                          4203*   4204*   4205    4212*   4225    4229*   4230*   4231    4235*   4237    4259*   4260    4304*
                          4316*   4317    4330*   4331    4348*   4381*   4384*   4387    4433*   4434    4455*   4456    4477*
                          4478    4499*   4500    4521*   4522    4543*   4544    4565*   4566    4587*   4588    4609*   4610
                          4631*   4632    4653*   4654    4675*   4676    4719*   4720    4764*   4765    4809*   4810    5137*
                          5138    5194*   5196    5237*   5240    5284*   5285    5330*   5331*   5332    5369*   5370    5373
                          5380*   5381    5383
R5      =%000005            46#    562     563*    567     572     574*    610     612*    613     614     615     616     617
                           618     619*    628*    631*    632*    633*    641     643     645     651     652*    656*    671
                           686*    697     705*    717*    732*    781*    782*    783     785    1258*   1259    1261    1281*
                          1283    1320*   1321    1323    1326    1328*   1329    1346*   1347    1349    1352    1354*   1355
                          1372*   1373    1375    1378    1380*   1381    1398*   1399    1401    1404    1406*   1407    1424*
                          1425    1427    1430    1432*   1433    1452*   1457*   1458    1460    1463*   1466    1483*   1484
                          1486    1489    1491*   1492    1509*   1510    1512    1515    1517*   1518    1535*   1536    1538
                          1541    1543*   1544    1561*   1562    1564    1567    1569*   1570    1587*   1588    1590    1593
                          1595*   1596    1613*   1614    1616    1619    1621*   1622    1641*   1646*   1647    1649    1652*
                          1655    1670*   1675    1695*   1698    1715*   1716    1719    1722    1725*   1726    1743*   1744
                          1747    1750    1753*   1754    1771*   1772    1775    1778    1781*   1782    1799*   1800    1803
```

```
                          1806    1809*   1810    1827*   1828    1831    1834    1837*   1838    1855*   1856    1859    1862
                          1865*   1866    1883*   1884    1886    1889    1891*   1892    1909*   1910    1912    1915    1917*
                          1918    1935*   1936    1938    1941    1943*   1944    1961*   1962    1964    1967    1969*   1970
                          1987*   1988    1990    1993    1995*   1996    2013*   2014    2016    2019    2021*   2022    2039*
                          2040    2042    2045    2047*   2048    2065*   2066    2068    2071    2073*   2074    2092*   2093
                          2095    2098*   2099    2101    2104*   2105    2107    2111*   2130*   2131    2133    2136*   2137
                          2139    2142*   2143    2145    2149*   2170*   2171    2175    2192*   2193    2195    2198    2200*
                          2201    2218*   2219    2221    2224    2226*   2227    2244*   2245    2247    2250    2252*   2253
                          2270*   2271    2273    2276    2278*   2279    2296*   2297    2299    2302    2304*   2305    2322*
                          2323    2325    2328    2330*   2331    2348*   2349    2351    2354    2356*   2357    2374*   2375
                          2377    2380    2382*   2383    2405*   2406    2408    2411*   2414    2420*   2421    2423    2426*
                          2429    2668*   2685*   2688*   2728*   2730    2734*   2741    2754*   2755    2764*   2765    2793*
                          2795    2799*   2806    2819*   2820    2829*   2830    2866*   2867    2868    2877    2906*   2910
                          2915*   2919    2924*   2928    2933*   2937    2942*   2946    2951*   2955    2975*   2977    3002*
                          3006    3011*   3015    3020*   3024    3029*   3033    3059*   3063    3068*   3073    3078*   3084
                          3111*   3117    3123*   3129    3135*   3141    3147*   3153    3297*   3299    3320*   3327    3340*
                          3341    3356*   3357    3377*   3381    3392*   3398    3406*   3409    3412*   3415    3457*   3477*
                          3488*   3499*   3510*   3524    3532*   3563*   3614*   3615    3624*   3638*   3647*   3653    3670
                          3673*   3682*   3696    3699*   3700    3724*   3725    3743*   3744    3762*   3763    3788*   3792*
                          3801    3804*   3805*   3827*   3831*   3840*   3850    3853*   3857*   3863*   3866    3905*   3906
                          3914*   3915    3933*   3934    3942*   3943    3961*   3962    3970*   3971    3989*   3990    3998*
                          3999    4017*   4018    4026*   4027    4045*   4046    4054*   4055    4073*   4074    4082*   4083
                          4101*   4102    4110*   4111    4137*   4139    4142*   4151*   4152    4158*   4162    4189*   4191
                          4199    4202*   4205    4213*   4224    4228*   4231    4236*   4248*   4249    4256*   4260    4263
                          4303*   4315*   4317    4331    4347*   4376*   4387    4394*   4395    4420*   4432*   4434    4442*
                          4454*   4456    4464*   4476*   4478    4486*   4498*   4500    4508*   4520*   4522    4530*   4542*
                          4544    4552*   4564*   4566    4574*   4586*   4588    4596*   4608*   4610    4618*   4630*   4632
                          4640*   4652*   4654    4662*   4674*   4676    4706*   4718*   4720    4751*   4763*   4765    4796*
                          4808*   4810    4829*   4839*   4846*   4869*   4879*   4886*   4909*   4920*   4927*   4954*   4992*
                          5017*   5042*   5067*   5136*   5138    5188*   5195*   5196    5231*   5238*   5239*   5240    5274*
                          5283*   5285    5320*   5329*   5332    5370*   5371*   5372*   5373    5381*   5382*   5383    5392
                          5432    5433    5434    5461*   5471    5472    5473    5474    5475*
SAVACT  001302             167#    427    1172*
SAVNUM  001303             168#    381*    483*    486*   1165*
SAVPC   001276             164#    667*    838
SAVR0   001260             157#    676*    681    5500    5531
SAVR1   001262             158#    675*    682    5491    5502    5514    5539
SAVR2   001264             159#    674*    683    5493    5504    5524    5533
SAVR3   001266             160#    673*    684    5490    5511    5516
SAVR4   001270             161#    672*    685    5496    5509    5520    5528    5537
SAVR5   001272             162#    671*    686    5507    5518    5526    5535
SAVSP   001274             163#
SAV05 = 104406             203#    780
SCOPE = 104400             191#   1234    1298    1332    1358    1384    1410    1436    1469    1495    1521    1547    1573
                          1599    1625    1658    1679    1701    1729    1757    1785    1813    1841    1869    1895    1921
                          1947    1973    1999    2025    2051    2077    2114    2152    2178    2204    2230    2256    2282
                          2308    2334    2360    2386    2432    2490    2533    2643    2695    2768    2833    2880    2958
                          2980    3036    3088    3157    3290    3360    3418    3516    3581    3767    3808    3868    4115
                          4166    4270    4351    4397    4680    4725    4770    4815    4851    4891    4932    4967    5000
                          5025    5050    5076    5110    5157    5202    5246    5292    5339    5386
SCOP1 = 104401             193#   1228    1264    1286    2457    2467    2478    2525    2575    2683    3066    3076    3087
                          3120    3132    3144    3156    3185    3198    3211    3224    3237    3250    3263    3276    3289
                          3621    3645    3675    3706    3728    3747    3766    4437    4459    4481    4503    4525    4547
                          4569    4591    4613    4635    4657    4679    5141    5199    5243    5288    5335
SERV.G  004640             521     768     912#    913
SETSCA  031342            3860    4243    5142    5389#
```

```
SETVEC  031654         4829    4839    4846    4869    4879    4886    4909    4920    4927    4954    4992    5017    5042
                       5067    5470#
SIMBCC  031474         4420    4442    4464    4486    4508    4530    4552    4574    4596    4618    4640    4662    4706
                       4751    4796    5429#
SKIP  = 000000         1301    1305    5486#
SP    =%000006           47#    379*    394*    395*    401     404     405     448*    517*    518*    519     523*    543
                        556*    562*    563     564*    574     578*    579*    580     581*    597     598     600*    601*
                        603     604     610*    611*    612     618*    655     656     667     693*    694*    695*    696*
                        697*    698     699*    732     733     734     735     736     749*    750*    751*    752*    753*
                        754*    755*    756     764*    765*    766     776     778     781     824     833*    855*    869*
                        873     879*    880*    886     887     899*    908     921*    922*    923     928     934     936
                        938*    939     942*    943*    944    1178*   1204    1206    1207*   1235    3859*   3862    4199*
                       4200*   4201*   4211    4212    4213    4224*   4225*   4226*   4234    4235    4236    4242*   4245
                       4852*   4892*   4933*   5001*   5026*   5051*   5077*   5390*   5402    5429*   5430*   5431*   5458
                       5459    5460
SPACNT= 003745          702*    723     726*    740#
SSWR    000176          112#    403
STACK = 001200           54#    379     448     833     855    4852    4892    4933
STAT    001236          148#
SV05    003452          671#
SWR     001202          123#    398     403*    408     422     427     432     451     515     524     531     552     565
                        762     769     774     821     828     830     914     915*    925*    930*    931*    932*    939*
                       1062
SW00  = 000001           34#    422
SW01  = 000002           33#   1062
SW02  = 000004           32#
SW03  = 000010           31#
SW04  = 000020           30#
SW05  = 000040           29#
SW06  = 000100           28#
SW07  = 000200           27#
SW08  = 000400           26#    828
SW09  = 001000           25#    552
SW10  = 002000           24#    830
SW11  = 004000           23#    531
SW12  = 010000           22#    565     769
SW13  = 020000           21#    774
SW14  = 040000           20#
SW15  = 100000           19#
SYNA00  001506          284#
SYNA01  001532          295#
SYNA02  001556          306#
SYNA03  001602          317#
SYNA04  001626          328#
SYNA05  001652          339#
SYNA06  001676          350#
SYNA07  001722          361#
SYNB00  001512          286#
SYNB01  001536          297#
SYNB02  001562          308#
SYNB03  001606          319#
SYNB04  001632          330#
SYNB05  001656          341#
SYNB06  001702          352#
SYNB07  001726          363#
SYNCX   001240          149#
```

```
SYNC00  001516          288#
SYNC01  001542          299#
SYNC02  001566          310#
SYNC03  001612          321#
SYNC04  001636          332#
SYNC05  001662          343#
SYNC06  001706          354#
SYNC07  001732          365#
SYNC2A  001426          260#   1009*
SYNC2B  001430          261#   1011*
SYNC2C  001432          262#   1013*
SYNC2D  001434          263#   1015*
SYND00  001522          290#
SYND01  001546          301#
SYND02  001572          312#
SYND03  001616          323#
SYND04  001642          334#
SYND05  001666          345#
SYND06  001712          356#
SYND07  001736          367#
S.C   = 050000           77#   2866    3058    3069    3079    3112    3124    3136    3148    3291    3321    3332    3344
                       3384    3517    3527    3738    3757    3794    3796    5376
TEMP    005562          706     856*    857*    969#
TEMP1   001246          152#    412*    413     418*    960    1191*   1192*   2669*   2690*   4252*   4258    4267*   4268
                       5131*   5136    5144*   5147*   5151*   5432*   5455*
TEMP2   001250          153#    413*    414     962    5128*   5150*   5151    5152    5155    5433*   5437*
TEMP3   001252          154#   5434*   5436    5446*   5448    5453*   5454*   5457
TEMP4   001254          155#
TEMP5   001256          156#   2671*   2674    2684*   2687*
TKCSR   001204          127#    527     586     912     919     940
TKDBR   001206          128#    517     529     588     594     764     921     942
TLAST = 031164         1088    5486#
TPCSR   001210          129#    570     592     771     926
TPDBR   001212          130#    572*    594*    773*    928*
TRPOK   003762          752#
TSTNO   001226          140#    389*    841     868    1073    1080    1082    1218*   1249*   1317*   1343*   1369*   1395*
                       1421*   1449*   1480*   1506*   1532*   1558*   1584*   1610*   1638*   1667*   1691*   1712*   1740*
                       1768*   1796*   1824*   1852*   1880*   1906*   1932*   1958*   1984*   2010*   2036*   2062*   2089*
                       2126*   2166*   2189*   2215*   2241*   2267*   2293*   2319*   2345*   2371*   2401*   2443*   2502*
                       2545*   2662*   2714*   2780*   2862*   2898*   2968*   2994*   3051*   3106*   3168*   3315*   3373*
                       3429*   3555*   3599*   3780*   3820*   3888*   4129*   4178*   4293*   4361*   4410*   4693*   4738*
                       4783*   4825*   4865*   4905*   4948*   4988*   5013*   5038*   5063*   5090*   5122*   5175*   5218*
                       5261*   5307*   5354*
TST1    007256         1076    1092    1218#
TST10   010154         1422    1449#
TST100  021560         3821    3888#
TST101  023162         3889    4129#
TST102  023362         4130    4178#
TST103  023776         4179    4293#
TST104  024252         4294    4361#
TST105  024444         4362    4410#
TST106  026100         4411    4693#
TST107  026256         4694    4738#
TST11   010250         1450    1480#
TST110  026434         4739    4783#
TST111  026612         4784    4825#
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 121                                                      PAGE: 0145
DZDVAB.P11       CROSS REFERENCE TABLE -- USER SYMBOLS

TST112  026762          4826    4865#
TST113  027132          4866    4905#
TST114  027310          4906    4948#
TST115  027450          4949    4988#
TST116  027540          4989    5013#
TST117  027630          5014    5038#
TST12   010324          1481    1506#
TST120  027720          5039    5063#
TST121  030012          5064    5090#
TST122  030114          5091    5122#
TST123  030314          5123    5175#
TST124  030454          5176    5218#
TST125  030616          5219    5261#
TST126  031000          5262    5307#
TST127  031164          5308    5354#   5486
TST13   010400          1507    1532#
TST130= ****** U        5355
TST14   010454          1533    1558#
TST15   010530          1559    1584#
TST16   010604          1585    1610#
TST17   010660          1611    1638#
TST2    007366          1219    1249#
TST20   010754          1639    1667#
TST21   011032          1668    1691#
TST22   011072          1692    1712#
TST23   011156          1713    1740#
TST24   011242          1741    1768#
TST25   011326          1769    1796#
TST26   011412          1797    1824#
TST27   011476          1825    1852#
TST3    007620          1250    1317#
TST30   011562          1853    1880#
TST31   011636          1881    1906#
TST32   011712          1907    1932#
TST33   011766          1933    1958#
TST34   012042          1959    1984#
TST35   012116          1985    2010#
TST36   012172          2011    2036#
TST37   012246          2037    2062#
TST4    007674          1318    1343#
TST40   012322          2063    2089#
TST41   012424          2090    2126#
TST42   012534          2127    2166#
TST43   012576          2167    2189#
TST44   012652          2190    2215#
TST45   012726          2216    2241#
TST46   013002          2242    2267#
TST47   013056          2268    2293#
TST5    007750          1344    1369#
TST50   013132          2294    2319#
TST51   013206          2320    2345#
TST52   013262          2346    2371#
TST53   013336          2372    2401#
TST54   013472          2402    2443#
TST55   013674          2444    2502#
TST56   014034          2503    2545#


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 122                                                      PAGE: 0146
DZDVAB.P11       CROSS REFERENCE TABLE -- USER SYMBOLS

TST57   014432          2547    2662#
TST6    010024          1370    1395#
TST60   014626          2663    2714#
TST61   015042          2715    2780#
TST62   015260          2781    2862#
TST63   015362          2863    2898#
TST64   015626          2899    2968#
TST65   015704          2969    2994#
TST66   016072          2995    3051#
TST67   016264          3052    3106#
TST7    010100          1396    1421#
TST70   016506          3107    3168#
TST71   017170          3169    3315#
TST72   017412          3316    3373#
TST73   017616          3374    3429#
TST74   020156          3430    3555#
TST75   020276          3556    3599#
TST76   021206          3600    3780#
TST77   021334          3781    3820#
TTST    002702           454*    455*    457*    458*    525#
TWOSYN= 010000            81#
TYPDAT  004266           795     813     816#
TYPE  = 104402           195#    406     411     424     429     453     461     474     475     477     479     481     569
                         582     599     692     729     796     797     800     801     803     805     809     814     859
                         916     918     946     984    1067    1085    1090    1174
TYPMSG  004166           793     796#
VECMAP  007102          1173    1181#
WRDCNT  003742           700*    730*    738#
WRKO.F  004254           808     811#
XBX     004060           770     772     774#
XCSR    002604           476     498#
XERR    002626           482     507#
XFR   = 030000            75#   3458    3478    3489    3500    3511    3560    3608    3656    3687    3689    3715    3785
                        3844    3864    3898    3926    3954    3982    4010    4038    4066    4094    4187    4246    4298
                        4309    4324    4339    4341    4369    4371    4379    4392    4414    4696    4741    4786    5097
                        5132    5185    5192    5228    5235    5271    5279    5317    5325    5364    5416    5421
XHEAD   005461           411     958#
XPASS   002620           480     504#
XPOLY   031646          4412*   4695*   4740*   4785*   5442    5453    5462#
XSTATQ  005506           417     958#
XTSTN   004374           802     839#
XVEC    002612           478     501#
$CRAP = 177777             1#   1211#   1214#   1240#   1245#   1309#   1313#   1335#   1339#   1361#   1365#   1387#   1391#
                        1413#   1417#   1439#   1445#   1472#   1476#   1498#   1502#   1524#   1528#   1550#   1554#   1576#
                        1580#   1602#   1606#   1628#   1634#   1660#   1663#   1682#   1687#   1704#   1708#   1732#   1736#
                        1760#   1764#   1788#   1792#   1816#   1820#   1844#   1848#   1872#   1876#   1898#   1902#   1924#
                        1928#   1950#   1954#   1976#   1980#   2002#   2006#   2028#   2032#   2054#   2058#   2080#   2084#
                        2117#   2121#   2155#   2162#   2181#   2185#   2207#   2211#   2233#   2237#   2259#   2263#   2285#
                        2289#   2311#   2315#   2337#   2341#   2363#   2367#   2389#   2397#   2434#   2439#   2493#   2498#
                        2536#   2541#   2647#   2657#   2705#   2710#   2771#   2776#   2836#   2851#   2853#   2858#   2882#
                        2894#   2960#   2964#   2982#   2990#   3040#   3047#   3091#   3102#   3160#   3164#   3304#   3311#
                        3362#   3369#   3421#   3425#   3539#   3551#   3583#   3595#   3770#   3776#   3810#   3816#   3870#
                        3884#   4118#   4125#   4169#   4174#   4272#   4279#   4281#   4289#   4353#   4357#   4401#   4406#
                        4683#   4689#   4728#   4734#   4773#   4779#   4818#   4821#   4858#   4861#   4898#   4901#   4939#
                        4944#   4980#   4984#   5005#   5009#   5030#   5034#   5055#   5059#   5081#   5086#   5113#   5118#
                        5163#   5171#   5206#   5214#   5250#   5257#   5296#   5303#   5343#   5350#
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 123                                                        PAGE: 0147
DZDVAB.P11      CROSS REFERENCE TABLE -- USER SYMBOLS

$E     = 000131              1#   1219   1221#  1250   1252#  1318   1319#  1344   1345#  1370   1371#  1396   1397#
                          1422   1423#  1450   1451#  1481   1482#  1507   1508#  1533   1534#  1559   1560#  1585
                          1586#  1611   1612#  1639   1640#  1668   1669#  1692   1693#  1713   1714#  1741   1742#
                          1769   1770#  1797   1798#  1825   1826#  1853   1854#  1881   1882#  1907   1908#  1933
                          1934#  1959   1960#  1985   1986#  2011   2012#  2037   2038#  2063   2064#  2090   2091#
                          2127   2128#  2167   2168#  2190   2191#  2216   2217#  2242   2243#  2268   2269#  2294
                          2295#  2320   2321#  2346   2347#  2372   2373#  2402   2403#  2444   2446#  2503   2505#
                          2547   2549#  2663   2665#  2715   2716#  2781   2782#  2863   2864#  2899   2900#  2969
                          2970#  2995   2996#  3052   3054#  3107   3108#  3169   3171#  3316   3317#  3374   3375#
                          3430   3432#  3556   3557#  3600   3601#  3781   3782#  3821   3822#  3889   3890#  4130
                          4131#  4179   4180#  4294   4295#  4362   4363#  4411   4412#  4694   4695#  4739   4740#
                          4784   4785#  4826   4827#  4866   4867#  4906   4907#  4949   4950#  4989   4990#  5014
                          5015#  5039   5040#  5064   5065#  5091   5092#  5123   5125#  5176   5178#  5219   5221#
                          5262   5264#  5308   5310#  5355   5356#
$N     = 000127              1#   1211   1216   1221#  1240   1247   1252#  1309   1315   1319#  1335   1341   1345#
                          1361   1367   1371#  1387   1393   1397#  1413   1419   1423#  1439   1447   1451#  1472
                          1478   1482#  1498   1504   1508#  1524   1530   1534#  1550   1556   1560#  1576   1582
                          1586#  1602   1608   1612#  1628   1636   1640#  1660   1665   1669#  1682   1689   1693#
                          1704   1710   1714#  1732   1738   1742#  1760   1766   1770#  1788   1794   1798#  1816
                          1822   1826#  1844   1850   1854#  1872   1878   1882#  1898   1904   1908#  1924   1930
                          1934#  1950   1956   1960#  1976   1982   1986#  2002   2008   2012#  2028   2034   2038#
                          2054   2060   2064#  2080   2087   2091#  2117   2124   2128#  2155   2164   2168#  2181
                          2187   2191#  2207   2213   2217#  2233   2239   2243#  2259   2265   2269#  2285   2291
                          2295#  2311   2317   2321#  2337   2343   2347#  2363   2369   2373#  2389   2399   2403#
                          2434   2441   2446#  2493   2500   2505#  2536   2543   2549#  2647   2660   2665#  2705
                          2712   2716#  2771   2778   2782#  2836   2853   2860   2864#  2882   2896   2900#  2960
                          2966   2970#  2982   2992   2996#  3040   3049   3054#  3091   3104   3108#  3160   3166
                          3171#  3304   3313   3317#  3362   3371   3375#  3421   3427   3432#  3539   3553   3557#
                          3583   3597   3601#  3770   3778   3782#  3810   3818   3822#  3870   3886   3890#  4118
                          4127   4131#  4169   4176   4180#  4272   4281   4291   4295#  4353   4359   4363#  4401
                          4408   4412#  4683   4691   4695#  4728   4736   4740#  4773   4781   4785#  4818   4823
                          4827#  4858   4863   4867#  4898   4903   4907#  4939   4946   4950#  4980   4986   4990#
                          5005   5011   5015#  5030   5036   5040#  5055   5061   5065#  5081   5088   5092#  5113
                          5120   5125#  5163   5173   5178#  5206   5216   5221#  5250   5259   5264#  5296   5305
                          5310#  5343   5352   5356#  5486#
$Y     = 000017              1#    182#   191    193#   195#   197#   199#   201#   203#   205#   207#   209#   211#
                           213#   215#   217#   219#   221#
.      = 033242             92#     93     96#   103#   104#   105#   106#   109#   111#   114#   118#   120#   165#
                           166#   167#   168#   169#   170#   279#   281#   282#   283#   284#   285#   286#   287#
                           288#   289#   290#   292#   293#   294#   295#   296#   297#   298#   299#   300#   301#
                           303#   304#   305#   306#   307#   308#   309#   310#   311#   312#   314#   315#   316#
                           317#   318#   319#   320#   321#   322#   323#   325#   326#   327#   328#   329#   330#
                           331#   332#   333#   334#   336#   337#   338#   339#   340#   341#   342#   343#   344#
                           345#   347#   348#   349#   350#   351#   352#   353#   354#   355#   356#   358#   359#
                           360#   361#   362#   363#   364#   365#   366#   367#   431    520    767    849    858
                           872    910#   920    927    941    968#   970#   972#   986   1177   1198   2702#  4140
                          4206   4209   4232   4708#  4709#  4711#  4712#  4753#  4754#  4756#  4757#  4798#  4799#
                          4801#  4802#  4963   5135   5374   5384   5486#
.BEGIN   002332            447#
.CNVRT   003542            210    693#
.CONVR   003536            208    692#
.DATAC   004576            220    898#
.DELAY   004476            216    869#
.EOP     002436            469#  5355
.ERRTA   033124            788   5541#
.HLT     004002             99    761#
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 124                                                        PAGE: 0148
DZDVAB.P11      CROSS REFERENCE TABLE -- USER SYMBOLS

.INSTE   003224            200    599#
.INSTR   003120            198    578#
.INST1   003140            582#   602
.MSG     003142            580*   583#
.MSTCL   004556            212    890#
.PARAM   003244            202    610#
.PFAIL   004402             97    380    846#   854
.RAMCL   004516            214    877#
.RES05   003504            206    681#
.ROMCL   004566            218    894#
.SAV05   003444            204    667#
.SCOPE   002634            192    514#
.SCOP1   003020            194    552#
.START   001742            115    378#   390
.TRPSR   003750            101    749#
.TRPTA   001314            190#   754
.TYPE    003044            196    562#
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 126                                                          PAGE: 0149
DZDVAB.P11        CROSS REFERENCE TABLE -- MACRO NAMES

DVEND       1#     463
DVFRNT      1#
HLT        55#     907    1236    1263    1285    1325    1331    1351    1357    1377    1383    1403    1409    1429    1435
          1456    1462    1468    1488    1494    1514    1520    1540    1546    1566    1572    1592    1598    1618    1624
          1645    1651    1657    1677    1700    1721    1728    1749    1756    1777    1784    1805    1812    1833    1840
          1861    1868    1888    1894    1914    1920    1940    1946    1966    1972    1992    1998    2018    2024    2044
          2050    2070    2076    2097    2103    2109    2135    2141    2147    2177    2197    2203    2223    2229    2249
          2255    2275    2281    2301    2307    2327    2333    2353    2359    2379    2385    2410    2416    2425    2431
          2456    2466    2477    2524    2574    2617    2620    2623    2628    2682    2732    2737    2743    2747    2751
          2757    2761    2767    2797    2802    2808    2812    2816    2822    2826    2832    2870    2876    2879    2912
          2921    2930    2939    2948    2957    2979    3008    3017    3026    3035    3065    3075    3086    3119    3131
          3143    3155    3301    3329    3343    3359    3383    3400    3411    3417    3467    3526    3535    3566    3573
          3580    3617    3627    3641    3655    3664    3672    3698    3705    3727    3746    3765    3791    3803    3852
          3908    3917    3936    3945    3964    3973    3992    4001    4020    4029    4048    4057    4076    4085    4104
          4113    4141    4154    4165    4207    4210    4233    4262    4306    4319    4333    4350    4389    4436    4458
          4480    4502    4524    4546    4568    4590    4612    4634    4656    4678    4722    4767    4812    4845    4855
          4885    4895    4926    4936    4966    4971    4976    5074    5095    5105    5109    5140    5198    5242    5287
          5334    5375    5385    5478    5482
$AAA        1#    2905    2914    2923    2932    2941    2950    3001    3010    3019    3028
$ABREG      1#    3538
$ADDT1      1#    1211
$ADDT2      1#    1308    1334    1360    1386    1412    1471    1497    1523    1549    1575    1601    1703    1731    1759
          1787    1815    1843    1871    1897    1923    1949    1975    2001    2027    2053    2180    2206    2232    2258
          2284    2310    2336    2362
$ADDT3      1#    1681    2154
$ADDT4      1#    2079    2116
$ADDT5      1#    1239
$ADT6       1#    1627
$ADT6A      1#    1438
$ALURE      1#    3186    3199    3212    3225    3238    3251    3264    3277
$BINBC      1#    4682    4727    4772
$BUFFE      1#     964
$BYTE0      1#    2388
$CHUCK      1#    3893    3921    3949    3977    4005    4033    4061    4089
$CK15       1#
$CK150      1#
$CLR.T      1#
$CRCX       1#    4416    4438    4460    4482    4504    4526    4548    4570    4592    4614    4636    4658
$CYCLE      1#     973
$DATXF      1#    3483    3494    3505
$DOSFR      1#    2835
$EOP        1#     463
$FINI       1#    5486
$GETFL      1#
$GETPA      1#    1068
$HEADE      1#
$HOHO1      1#    3060    3070    3081    3114    3126    3138    3150
$INIT1      1#    2704    2770
$INTA       1#    4817    4857    4897
$INTB       1#    4938
$MEMEX      1#    2646
$MSCAN      1#    3769
$MSG        1#     958
$NSR1       1#    5080
$NSR2       1#    5162    5205    5249    5295
$NSR3       1#    5342


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 127                                                          PAGE: 0150
DZDVAB.P11        CROSS REFERENCE TABLE -- MACRO NAMES

$PFAIL      1#     842
$PRIO       1#    4979    5004    5029    5054
$RAMCL      1#     869
$RXSHI      1#
$SCOPE      1#     510
$SECT1      1#    2434
$SECT2      1#    2493
$SECT3      1#    2536
$SETCL      1#    3039
$SETLI      1#
$SETSC      1#    5388
$SETSY      1#
$SET.T      1#
$SILOI      1#
$SIMBC      1#    5429
$TRPDE      1#     191     193     195     197     199     201     203     205     207     209     211     213     215     217
           219
$TSTN       1#    1216    1247    1315    1341    1367    1393    1419    1447    1478    1504    1530    1556    1582    1608
          1636    1665    1689    1710    1738    1766    1794    1822    1850    1878    1904    1930    1956    1982    2008
          2034    2060    2087    2124    2164    2187    2213    2239    2265    2291    2317    2343    2369    2399    2441
          2500    2543    2660    2712    2778    2860    2896    2966    2992    3049    3104    3166    3313    3371    3427
          3553    3597    3778    3818    3886    4127    4176    4291    4359    4408    4691    4736    4781    4823    4863
          4903    4946    4986    5011    5036    5061    5088    5120    5173    5216    5259    5305    5352
$TXSHI      1#
$VARIA      1#     117
$XFER       1#    3420
$XXCRC      1#    4400
$XZ         1#    1211    1214    1240    1245    1309    1313    1335    1339    1361    1365    1387    1391    1413    1417
          1439    1445    1472    1476    1498    1502    1524    1528    1550    1554    1576    1580    1602    1606    1628
          1634    1660    1663    1682    1687    1704    1708    1732    1736    1760    1764    1788    1792    1816    1820
          1844    1848    1872    1876    1898    1902    1924    1928    1950    1954    1976    1980    2002    2006    2028
          2032    2054    2058    2080    2084    2117    2121    2155    2162    2181    2185    2207    2211    2233    2237
          2259    2263    2285    2289    2311    2315    2337    2341    2363    2367    2389    2397    2434    2439    2493
          2498    2536    2541    2647    2657    2705    2710    2771    2776    2836    2851    2853    2858    2882    2894
          2960    2964    2982    2990    3040    3047    3091    3102    3160    3164    3304    3311    3362    3369    3421
          3425    3539    3551    3583    3595    3770    3776    3810    3816    3870    3884    4118    4125    4169    4174
          4272    4279    4281    4289    4353    4357    4401    4406    4683    4689    4728    4734    4773    4779    4818
          4821    4858    4861    4898    4901    4939    4944    4980    4984    5005    5009    5030    5034    5055    5059
          5081    5086    5113    5118    5163    5171    5206    5214    5250    5257    5296    5303    5343    5350
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 129                                                  PAGE: 0151
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADC        5438
ADCB        991     999
ADD         418     564     581     652     699     709     754     785     788     884     992    1001    1022    1024    1030
           1032    1034    1037    1039    1041    1157    1202    1229    2512    2526    2560    2579    2600    2624    3352
           3395    3468    3853    4394
ASL         631     632     633     752     784     786
BCC         904    1193
BCS         933    2481    2632
BEQ         415     423     452     488     516     525     534     553     555     591     622     630     724     763     770
            777     793     799     808     813     817     831     924    1063    1115    1156    1163    1262    1284    1324
           1330    1350    1356    1376    1382    1402    1408    1428    1434    1455    1461    1467    1487    1493    1513
           1519    1539    1545    1565    1571    1591    1597    1617    1623    1644    1650    1656    1676    1699    1720
           1727    1748    1755    1776    1783    1804    1811    1832    1839    1860    1867    1887    1893    1913    1919
           1939    1945    1965    1971    1991    1997    2017    2023    2043    2049    2069    2075    2096    2102    2108
           2134    2140    2146    2176    2196    2202    2222    2228    2248    2254    2274    2280    2300    2306    2326
           2332    2352    2358    2378    2384    2409    2415    2424    2430    2455    2465    2476    2523    2573    2612
           2616    2619    2622    2681    2731    2736    2742    2746    2750    2756    2766    2796    2801    2807    2811
           2815    2821    2831    2869    2875    2911    2920    2929    2938    2947    2956    2978    3007    3016    3025
           3034    3064    3074    3085    3118    3130    3142    3154    3300    3328    3342    3358    3382    3394    3399
           3410    3416    3466    3470    3525    3534    3565    3572    3579    3616    3626    3640    3654    3663    3671
           3697    3704    3726    3745    3764    3790    3802    3851    3907    3916    3935    3944    3963    3972    3991
           4000    4019    4028    4047    4056    4075    4084    4103    4112    4140    4153    4163    4206    4209    4232
           4261    4305    4318    4332    4349    4388    4396    4435    4457    4479    4501    4523    4545    4567    4589
           4611    4633    4655    4677    4721    4766    4811    4965    4970    4975    5139    5146    5156    5197    5241
           5286    5291    5333    5338    5374    5384    5393    5440
BGT         626     937
BHI         642
BIC         434     518     708     753     765     787     922     938     943    1096    1200    1205    1326    1352    1378
           1404    1430    1463    1464    1489    1515    1541    1567    1593    1619    1652    1653    1718    1722    1724
           1746    1750    1752    1774    1778    1780    1802    1806    1808    1830    1834    1836    1858    1862    1864
           1889    1915    1941    1967    1993    2019    2045    2071    2198    2224    2250    2276    2302    2328    2354
           2380    2678    2740    2805    2871    2908    2917    2922    2926    2935    2944    2953    3004    3013    3022
           3031    3340    3412    3413    3461    3805    3857    3904    3913    3932    3941    3960    3969    3988    3997
           4016    4025    4044    4053    4072    4081    4100    4109    4138    4148    4151    4161    4194    4204    4219
           4230    4972    5147    5371    5444    5450    5451    5453
BICB        589     627
BIS         895     901     939    1146    1147    1148    1149    1150    1151    1152    1153    1199    1204    1672    2791
           3055    3056    3068    3078    3293    3356    3407    3459    3462    3479    3490    3501    3512    3519    3528
           3649    3651    3657    3659    3661    3666    3668    3690    3692    3694    4195    4215    4220    4843    4883
           4924    4959    4960    5153    5360    5372    5382    5391    5452    5454
BISB        628    5151
BIT         422     451     524     531     552     565     769     774     828     830     912    1062    1114    2874    3350
           3393    3471    4395    5145    5155    5290    5337    5439
BITB        645     987
BLO         644
BLOS        428    1190
BLT         624     935
BMI         409    5101
BNE         393     421     442     450     484     520     532     537     566     573     596     646     654     718     722
            727     731     767     775     795     829     858     872     885     906     913     945     983     988     994
           1004    1066    1079    1081    1083    1089    1098    1103    1108    1132    1138    1140    1142    1159    1170
           1198    1231    1267    1271    1275    1289    1293    1297    2112    2150    2459    2470    2485    2489    2516
           2529    2532    2558    2564    2578    2584    2602    2625    2627    2638    2642    2686    2691    2694    2724
           2790    2878    3331    3351    3353    3396    3472    3674    3701    3807    3834    3838    3856    3867    4157
           4241    4250    4265    4269    4386    4724    4769    4814    4963    5104    5201    5245    5401    5456
BPL         528     568     571     587     593     772     822     920     927     941    2760    2825    4150    5094    5108


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 130                                                  PAGE: 0152
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

           5135
BR          400     419     431     456     526     530     602     634     636     849     986     996    1086    1091    1101
           1106    1111    1177    1203    4854    4856    4894    4896    4935    4937
CLC         711     713     715     929     989     997    1166    2110    2148    2468    4155    4383    5404    5445
CLR         382     387     388     396     425     439     470     522     539     540     619     856     862     883     900
            925    1130    1175    1194    1196    1201    1233    1254    1255    1256    1328    1354    1380    1406    1432
           1452    1491    1517    1543    1569    1595    1621    1641    1678    1695    1725    1753    1781    1809    1837
           1865    1891    1917    1943    1969    1995    2021    2047    2073    2098    2113    2136    2151    2200    2226
           2252    2278    2304    2330    2356    2382    2404    2417    2419    2446    2447    2451    2482    2486    2505
           2506    2507    2517    2518    2549    2550    2551    2559    2566    2567    2568    2581    2594    2595    2597
           2603    2604    2609    2629    2639    2665    2676    2718    2719    2728    2734    2764    2784    2785    2793
           2799    2829    3349    3388    3457    3460    3532    3563    3624    3638    3788    3823    3824    3835    3839
           3890    4189    4190    4251    4252    4266    4303    4347    4419    4441    4463    4485    4507    4529    4551
           4573    4595    4617    4639    4661    4698    4699    4743    4744    4788    4789    4833    4838    4850    4853
           4873    4878    4890    4893    4913    4918    4931    4934    4952    4961    4999    5024    5049    5075    5098
           5127    5128    5148    5149    5187    5188    5189    5230    5231    5232    5273    5274    5275    5276    5319
           5320    5321    5322    5435
CLRB        383     384     471     538     728     779     861    1099    1133    1161
CMP         401     414     440     441     515     536     641     643     762     776     934     936     993    1002    1077
           1080    1082    1088    1102    1107    1131    1139    1158    1188    1189    1206    1261    1265    1266    1268
           1270    1272    1283    1287    1288    1290    1292    1294    1323    1329    1349    1355    1375    1381    1401
           1407    1427    1433    1460    1466    1486    1492    1512    1518    1538    1544    1564    1570    1590    1596
           1616    1622    1649    1655    1673    1675    1698    1719    1726    1747    1754    1775    1782    1803    1810
           1831    1838    1859    1866    1886    1892    1912    1918    1938    1944    1964    1970    1990    1996    2016
           2022    2042    2048    2068    2074    2095    2101    2107    2133    2139    2145    2175    2195    2201    2221
           2227    2247    2253    2273    2279    2299    2305    2325    2331    2351    2357    2377    2383    2408    2414
           2423    2429    2454    2464    2475    2484    2488    2515    2522    2528    2531    2557    2563    2572    2577
           2583    2610    2621    2637    2641    2680    2693    2730    2741    2755    2765    2795    2806    2820    2830
           2868    2877    2910    2919    2928    2937    2946    2955    2977    3006    3015    3024    3033    3063    3073
           3084    3117    3129    3141    3153    3299    3327    3341    3357    3381    3398    3409    3415    3469    3524
           3615    3653    3670    3696    3700    3725    3744    3763    3801    3833    3837    3850    3866    3906    3915
           3934    3943    3962    3971    3990    3999    4018    4027    4046    4055    4074    4083    4102    4111    4139
           4152    4162    4205    4208    4231    4240    4249    4260    4264    4268    4317    4331    4387    4434    4456
           4478    4500    4522    4544    4566    4588    4610    4632    4654    4676    4720    4765    4810    4969    5138
           5196    5240    5285    5332    5373    5383
CMPB        427     519     590     621     623     625     629     766     923     944    1155    2615    2618    3832    3855
           4263
COM        2172    2458    5441    5443
COMB        407
DEC         595     717     730     871    1169    1230    1269    1273    1274    1291    1295    1296    2685    2690    2723
           2789    3330    3806    4385    4968    4974    5400    5455
DECB        483     653     721     726
EMT          55
HALT         93     426     430     435     825     848     985    1176
INC         472     535     827     857     905    1018    1020    1026    1028    1197    2483    2487    2514    2527    2530
           2556    2562    2576    2582    2636    2640    2684    2692    2726    3673    3699    3792    3804    3836    3854
           3863    4239    4248    4267    5103    5144    5150    5289    5336    5396    5399
INCB       1116    1154    1168    4723    4768    4813    4962    5200    5244
JMP         115     462     497     545     756     834     865    1093    1173
JSR         410     491     521     768    1046    1051    1056    1061    3177    3181    3190    3194    3203    3207    3216
           3220    3229    3233    3242    3246    3255    3259    3268    3272    3281    3285    3480    3491    3502    3513
           4420    4424    4442    4446    4464    4468    4486    4490    4508    4512    4530    4534    4552    4556    4574
           4578    4596    4600    4618    4622    4640    4644    4662    4666    4706    4710    4751    4755    4796    4800
           4829    4839    4846    4869    4879    4886    4909    4920    4927    4954    4992    5017    5042    5067
MOV         378     379     380     385     389     390     394     395     397     402     403     404     405     412     413
            436     437     438     447     448     454     455     457     458     460     473     487     496     517     523
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 131                                             PAGE: 0153
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

           529     541     542     543     556     562     563     574     578     579     580     584     585     594     597
           598     600     601     603     604     610     611     612     613     614     615     618     620     650     651
           655     656     667     671     672     673     674     675     676     681     682     683     684     685     686
           693     694     695     696     697     698     700     703     704     706     707     719     732     733     734
           735     736     749     751     755     764     778     781     783     789     790     791     824     832     833
           847     854     855     869     870     873     878     879     880     881     882     886     887     891     899
           902     908     914     915     921     928     942     995    1000    1005    1006    1007    1008    1009    1010
          1011    1012    1013    1014    1015    1016    1017    1019    1021    1023    1025    1027    1029    1031    1033
          1036    1038    1040    1043    1044    1045    1048    1049    1050    1053    1054    1055    1058    1059    1060
          1076    1078    1084    1092    1095    1129    1134    1135    1136    1144    1160    1171    1178    1181    1182
          1183    1184    1185    1186    1187    1195    1207    1218    1219    1220    1221    1222    1223    1224    1232
          1235    1249    1250    1251    1252    1253    1257    1258    1259    1260    1277    1278    1279    1280    1281
          1282    1317    1318    1319    1320    1321    1322    1327    1343    1344    1345    1346    1347    1348    1353
          1369    1370    1371    1372    1373    1374    1379    1395    1396    1397    1398    1399    1400    1405    1421
          1422    1423    1424    1425    1426    1431    1449    1450    1451    1453    1454    1457    1458    1459    1465
          1480    1481    1482    1483    1484    1485    1490    1506    1507    1508    1509    1510    1511    1516    1532
          1533    1534    1535    1536    1537    1542    1558    1559    1560    1561    1562    1563    1568    1584    1585
          1586    1587    1588    1589    1594    1610    1611    1612    1613    1614    1615    1620    1638    1639    1640
          1642    1643    1646    1647    1648    1654    1667    1668    1669    1670    1671    1674    1691    1692    1693
          1696    1697    1712    1713    1714    1715    1716    1717    1723    1740    1741    1742    1743    1744    1745
          1751    1768    1769    1770    1771    1772    1773    1779    1796    1797    1798    1799    1800    1801    1807
          1824    1825    1826    1827    1828    1829    1835    1852    1853    1854    1855    1856    1857    1863    1880
          1881    1882    1883    1884    1885    1890    1906    1907    1908    1909    1910    1911    1916    1932    1933
          1934    1935    1936    1937    1942    1958    1959    1960    1961    1962    1963    1968    1984    1985    1986
          1987    1988    1989    1994    2010    2011    2012    2013    2014    2015    2020    2036    2037    2038    2039
          2040    2041    2046    2062    2063    2064    2065    2066    2067    2072    2089    2090    2091    2092    2093
          2094    2099    2100    2104    2105    2106    2126    2127    2128    2129    2130    2131    2132    2137    2138
          2142    2143    2144    2166    2167    2168    2170    2171    2173    2174    2189    2190    2191    2192    2193
          2194    2199    2215    2216    2217    2218    2219    2220    2225    2241    2242    2243    2244    2245    2246
          2251    2267    2268    2269    2270    2271    2272    2277    2293    2294    2295    2296    2297    2298    2303
          2319    2320    2321    2322    2323    2324    2329    2345    2346    2347    2348    2349    2350    2355    2371
          2372    2373    2374    2375    2376    2381    2401    2402    2403    2405    2407    2411    2413    2418    2420
          2422    2426    2428    2443    2444    2445    2448    2452    2453    2460    2461    2462    2463    2471    2472
          2473    2474    2502    2503    2504    2508    2511    2519    2521    2545    2546    2547    2548    2552    2555
          2571    2596    2598    2605    2608    2613    2614    2635    2662    2663    2664    2666    2667    2668    2669
          2671    2672    2677    2687    2688    2714    2715    2716    2717    2720    2721    2722    2727    2729    2735
          2739    2745    2749    2753    2754    2763    2780    2781    2782    2783    2786    2787    2788    2792    2794
          2800    2804    2810    2814    2818    2819    2828    2862    2863    2865    2866    2867    2898    2899    2901
          2902    2903    2904    2905    2906    2907    2913    2914    2915    2916    2923    2924    2925    2931    2932
          2933    2934    2940    2941    2942    2943    2949    2950    2951    2952    2968    2969    2971    2972    2974
          2975    2976    2994    2995    2997    2998    2999    3000    3001    3002    3003    3009    3010    3011    3012
          3018    3019    3020    3021    3027    3028    3029    3030    3051    3052    3053    3057    3058    3059    3060
          3062    3067    3069    3070    3072    3077    3079    3081    3083    3106    3107    3109    3110    3111    3112
          3114    3116    3122    3123    3124    3126    3128    3134    3135    3136    3138    3140    3146    3147    3148
          3150    3152    3168    3169    3170    3172    3173    3189    3202    3215    3228    3241    3254    3267    3280
          3291    3292    3295    3296    3297    3298    3315    3316    3318    3319    3320    3321    3323    3325    3326
          3332    3336    3338    3339    3344    3347    3354    3355    3373    3374    3376    3377    3378    3379    3380
          3384    3386    3389    3391    3392    3397    3406    3408    3414    3429    3430    3431    3433    3434    3458
          3463    3465    3477    3478    3487    3488    3489    3498    3499    3500    3509    3510    3511    3517    3518
          3521    3523    3527    3530    3533    3555    3556    3558    3559    3560    3561    3564    3567    3569    3571
          3574    3576    3578    3599    3600    3604    3606    3607    3608    3609    3611    3613    3614    3622    3625
          3631    3632    3634    3636    3639    3646    3647    3648    3650    3652    3656    3658    3660    3662    3665
          3667    3669    3679    3681    3682    3683    3684    3685    3687    3689    3691    3693    3695    3703    3710
          3712    3713    3715    3717    3719    3721    3723    3724    3733    3735    3736    3738    3740    3742    3743
          3752    3754    3755    3757    3759    3761    3762    3780    3781    3783    3784    3785    3786    3789    3793
          3794    3796    3798    3800    3820    3821    3825    3826    3827    3828    3831    3840    3841    3842    3843


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 132                                             PAGE: 0154
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

          3844    3845    3847    3849    3858    3859    3862    3864    3888    3889    3894    3895    3896    3898    3900
          3902    3903    3905    3910    3911    3912    3914    3922    3923    3924    3926    3928    3930    3931    3933
          3938    3939    3940    3942    3950    3951    3952    3954    3956    3958    3959    3961    3966    3967    3968
          3970    3978    3979    3980    3982    3984    3986    3987    3989    3994    3995    3996    3998    4006    4007
          4008    4010    4012    4014    4015    4017    4022    4023    4024    4026    4034    4035    4036    4038    4040
          4042    4043    4045    4050    4051    4052    4054    4062    4063    4064    4066    4068    4070    4071    4073
          4078    4079    4080    4082    4090    4091    4092    4094    4096    4098    4099    4101    4106    4107    4108
          4110    4129    4130    4132    4133    4134    4135    4136    4137    4142    4143    4144    4146    4147    4158
          4159    4160    4164    4178    4179    4181    4182    4183    4185    4187    4193    4196    4198    4199    4200
          4201    4202    4203    4211    4212    4213    4214    4217    4221    4223    4224    4225    4226    4227    4228
          4229    4234    4235    4236    4237    4242    4245    4246    4253    4254    4255    4256    4259    4293    4294
          4296    4297    4298    4299    4301    4304    4307    4309    4311    4313    4315    4316    4321    4322    4324
          4326    4328    4330    4336    4337    4339    4341    4343    4345    4348    4361    4362    4364    4365    4366
          4367    4369    4371    4373    4374    4376    4377    4379    4381    4382    4390    4392    4410    4411    4412
          4413    4414    4416    4418    4428    4430    4432    4433    4438    4440    4450    4452    4454    4455    4460
          4462    4472    4474    4476    4477    4482    4484    4494    4496    4498    4499    4504    4506    4516    4518
          4520    4521    4526    4528    4538    4540    4542    4543    4548    4550    4560    4562    4564    4565    4570
          4572    4582    4584    4586    4587    4592    4594    4604    4606    4608    4609    4614    4616    4626    4628
          4630    4631    4636    4638    4648    4650    4652    4653    4658    4660    4670    4672    4674    4675    4693
          4694    4695    4696    4697    4701    4702    4703    4704    4705    4714    4716    4718    4719    4738    4739
          4740    4741    4742    4746    4747    4748    4749    4750    4759    4761    4763    4764    4783    4784    4785
          4786    4787    4791    4792    4793    4794    4795    4804    4806    4808    4809    4825    4826    4827    4834
          4836    4852    4865    4866    4867    4874    4876    4892    4905    4906    4907    4914    4916    4919    4933
          4948    4949    4950    4953    4958    4988    4989    4990    4996    4997    5001    5013    5014    5015    5021
          5022    5026    5038    5039    5040    5046    5047    5051    5063    5064    5065    5071    5072    5077    5090
          5091    5096    5097    5122    5123    5124    5126    5129    5130    5131    5132    5136    5137    5152    5175
          5176    5177    5179    5180    5181    5182    5183    5185    5190    5192    5194    5195    5218    5219    5220
          5222    5223    5224    5225    5226    5228    5233    5235    5237    5238    5261    5262    5263    5265    5266
          5267    5268    5269    5271    5277    5279    5281    5283    5284    5307    5308    5309    5311    5312    5313
          5314    5315    5317    5323    5325    5327    5329    5330    5354    5355    5357    5358    5359    5361    5362
          5364    5366    5368    5369    5370    5376    5379    5380    5381    5390    5392    5394    5397    5402    5403
          5413    5414    5416    5418    5419    5421    5423    5424    5429    5430    5431    5432    5433    5434    5436
          5442    5447    5448    5449    5457    5458    5459    5460    5471    5472
MOVB       381     386     432     433     485     486     572     588     616     617     701     702     705     710     720
           725     773     806    1100    1104    1105    1109    1110    1112    1113    1164    1165    1172    1191    2406
          2412    2421    2427    2449    2450    2509    2510    2520    2553    2554    2569    2570    2606    2607    2633
          2634    2673    2674    2675    3829    3830    4191    4192    4257    4258    5473    5474
NOP        399     454     455     492     493     494     495    1226    1227    2873    3335    3346    4835    4837    4844
          4875    4877    4884    4915    4917    4925    4998    5023    5048    5073
RESET      469     489    1128    2725
ROL        930     931     932    2111    2149    2469    2480    2631    4156    4384
ROLB       990     998    1167
ROR        712     714     716    1192    5405    5437    5446
RORB       903
RTI        557     575     605     657     677     687     737     835     874     888     892     896     909    1179    1208
          1237    4973    4977    5002    5027    5052    5078    5479    5483
RTS        947    1117    1209    3302    3536    5426    5461    5475
SEC       2479    2630
SUB        750     782    2689
SWAB      5239    5331
TRAP       191     193     195     197     199     201     203     205     207     209     211     213     215     217     219
TST        398     420     449     554     794     798     807     812     816     821    1065    1087    1097    1137    1141
          1145    1225    2626    2733    2738    2744    2748    2752    2758    2759    2762    2798    2803    2809    2813
          2817    2823    2824    2827    4149    4964    5093    5100    5106    5107    5134
TSTB       392     408     527     533     567     570     586     592     723     771     792     919     926     940     982
          1162
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 133                                                        PAGE: 0155
DZDVAB.P11       CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.ASCIZ     119     949     956     958    5486
.BLKB      165     166     167     168     169
.BLKW      104     105     106     281     282     283     284     285     286     287     288     289     290     292     293
           294     295     296     297     298     299     300     301     303     304     305     306     307     308     309
           310     311     312     314     315     316     317     318     319     320     321     322     323     325     326
           327     328     329     330     331     332     333     334     336     337     338     339     340     341     342
           343     344     345     347     348     349     350     351     352     353     354     355     356     358     359
           360     361     362     363     364     365     366     367     910    4708    4709    4711    4712    4753    4754
          4756    4757    4798    4799    4801    4802
.BYTE      176     177     178     179     245     246     247     248     250     251     252     253     499     502     505
           508     837     840     867     953     959     961    1074    1075    2697    2698    2699    2700    2701    4832
          4842    4849    4872    4882    4889    4912    4923    4930    4957    4995    5020    5045    5070    5490    5492
          5495    5497    5499    5501    5503    5506    5508    5510    5513    5515    5517    5519    5523    5525    5527
          5530    5532    5534    5536    5538
.ENABL      15
.END      5582
.ENDC     1211    1212    1215    1220    1241    1246    1251    1310    1314    1319    1336    1340    1345    1362    1366
          1371    1388    1392    1397    1414    1418    1423    1440    1446    1451    1473    1477    1482    1499    1503
          1508    1525    1529    1534    1551    1555    1560    1577    1581    1586    1603    1607    1612    1629    1635
          1640    1661    1664    1669    1683    1688    1693    1697    1705    1709    1714    1733    1737    1742    1761
          1765    1770    1789    1793    1798    1817    1821    1826    1845    1849    1854    1873    1877    1882    1899
          1903    1908    1925    1929    1934    1951    1955    1960    1977    1981    1986    2003    2007    2012    2029
          2033    2038    2055    2059    2064    2081    2085    2091    2118    2122    2128    2156    2163    2168    2174
          2182    2186    2191    2208    2212    2217    2234    2238    2243    2260    2264    2269    2286    2290    2295
          2312    2316    2321    2338    2342    2347    2364    2368    2373    2390    2398    2403    2435    2440    2445
          2494    2499    2504    2537    2542    2548    2648    2658    2664    2706    2711    2716    2727    2772    2777
          2782    2792    2837    2852    2854    2859    2864    2883    2895    2900    2907    2916    2925    2934    2943
          2952    2961    2965    2970    2983    2991    2996    3003    3012    3021    3030    3041    3048    3053    3092
          3103    3108    3161    3165    3170    3305    3312    3317    3363    3370    3375    3422    3426    3431    3540
          3552    3557    3584    3596    3601    3771    3777    3782    3811    3817    3822    3871    3885    3890    4119
          4126    4131    4170    4175    4180    4273    4280    4282    4290    4295    4354    4358    4363    4402    4407
          4412    4684    4690    4695    4729    4735    4740    4774    4780    4785    4819    4822    4827    4859    4862
          4867    4899    4902    4907    4940    4945    4950    4981    4985    4990    5006    5010    5015    5031    5035
          5040    5056    5060    5065    5082    5087    5092    5114    5119    5124    5164    5172    5177    5195    5196
          5202    5207    5215    5220    5238    5240    5246    5251    5258    5263    5284    5285    5289    5292    5297
          5304    5309    5330    5332    5336    5339    5344    5351    5356    5582
.EQUIV      55
.EVEN      170     181     951     957     958     963    2702    5486
.IF          1    1211    1214    1219    1240    1245    1250    1309    1313    1318    1335    1339    1344    1361    1365
          1370    1387    1391    1396    1413    1417    1422    1439    1445    1450    1472    1476    1481    1498    1502
          1507    1524    1528    1533    1550    1554    1559    1576    1580    1585    1602    1606    1611    1628    1634
          1639    1660    1663    1668    1682    1687    1692    1695    1704    1708    1713    1732    1736    1741    1760
          1764    1769    1788    1792    1797    1816    1820    1825    1844    1848    1853    1872    1876    1881    1898
          1902    1907    1924    1928    1933    1950    1954    1959    1976    1980    1985    2002    2006    2011    2028
          2032    2037    2054    2058    2063    2080    2084    2090    2117    2121    2127    2155    2162    2167    2170
          2181    2185    2190    2207    2211    2216    2233    2237    2242    2259    2263    2268    2285    2289    2294
          2311    2315    2320    2337    2341    2346    2363    2367    2372    2389    2397    2402    2434    2439    2444
          2493    2498    2503    2536    2541    2547    2647    2657    2663    2705    2710    2715    2725    2771    2776
          2781    2791    2836    2851    2853    2858    2863    2882    2894    2899    2906    2915    2924    2933    2942
          2951    2960    2964    2969    2982    2990    2995    3002    3011    3020    3029    3040    3047    3052    3091
          3102    3107    3160    3164    3169    3304    3311    3316    3362    3369    3374    3421    3425    3430    3539
          3551    3556    3583    3595    3600    3770    3776    3781    3810    3816    3821    3870    3884    3889    4118
          4125    4130    4169    4174    4179    4272    4279    4281    4289    4294    4353    4357    4362    4401    4406
          4411    4683    4689    4694    4728    4734    4739    4773    4779    4784    4818    4821    4826    4858    4861
          4866    4898    4901    4906    4939    4944    4949    4980    4984    4989    5005    5009    5014    5030    5034
          5039    5055    5059    5064    5081    5086    5091    5113    5118    5123    5163    5171    5176    5190    5195


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 134                                                         PAGE: 0156
DZDVAB.P11       CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

          5196    5200    5202    5206    5214    5219    5233    5238    5244    5246    5250    5257    5262    5276    5284
          5285    5289    5296    5303    5308    5322    5330    5336    5343    5350    5355    5582
.IFF      1212    1214    1219    1220    1241    1245    1250    1251    1310    1313    1318    1319    1336    1339    1344
          1345    1362    1365    1370    1371    1388    1391    1396    1397    1414    1417    1422    1423    1440    1445
          1450    1451    1473    1476    1481    1482    1499    1502    1507    1508    1525    1528    1533    1534    1551
          1554    1559    1560    1577    1580    1585    1586    1603    1606    1611    1612    1629    1634    1639    1640
          1661    1663    1668    1669    1683    1687    1688    1692    1693    1695    1705    1708    1713    1714    1733
          1736    1741    1742    1761    1764    1769    1770    1789    1792    1797    1798    1817    1820    1825    1826
          1845    1848    1853    1854    1873    1876    1881    1882    1899    1902    1907    1908    1925    1928    1933
          1934    1951    1954    1959    1960    1977    1980    1985    1986    2003    2006    2011    2012    2029    2032
          2037    2038    2055    2058    2063    2064    2081    2084    2090    2091    2118    2121    2127    2128    2155
          2156    2162    2167    2168    2174    2182    2185    2190    2191    2208    2211    2216    2217    2234    2237
          2242    2243    2260    2263    2268    2269    2286    2289    2294    2295    2312    2315    2320    2321    2338
          2341    2346    2347    2364    2367    2372    2373    2390    2397    2402    2403    2435    2439    2444    2445
          2494    2498    2503    2504    2537    2541    2547    2548    2648    2657    2663    2664    2706    2710    2715
          2716    2727    2772    2776    2781    2782    2791    2837    2851    2854    2858    2863    2864    2883    2894
          2899    2900    2906    2915    2925    2934    2943    2952    2961    2964    2969    2970    2983    2990    2995
          2996    3003    3012    3021    3030    3041    3047    3052    3053    3092    3102    3107    3108    3161    3164
          3169    3170    3305    3311    3316    3317    3363    3369    3374    3375    3422    3425    3430    3431    3540
          3551    3556    3557    3584    3595    3600    3601    3771    3776    3781    3782    3811    3816    3821    3822
          3871    3884    3889    3890    4119    4125    4130    4131    4170    4174    4179    4180    4273    4279    4282
          4289    4294    4295    4354    4357    4362    4363    4402    4406    4411    4412    4684    4689    4694    4695
          4729    4734    4739    4740    4774    4779    4784    4785    4819    4821    4826    4827    4859    4861    4866
          4867    4899    4901    4906    4907    4940    4944    4949    4950    4981    4984    4989    4990    5006    5009
          5014    5015    5031    5034    5039    5040    5056    5059    5064    5065    5082    5086    5091    5092    5114
          5118    5123    5124    5164    5171    5176    5177    5195    5196    5207    5214    5219    5220    5238    5240
          5251    5257    5262    5263    5276    5285    5297    5303    5308    5309    5322    5330    5344    5350    5355
.IFT         1    1211    1214    1240    1245    1309    1313    1335    1339    1361    1365    1387    1391    1413    1417
          1439    1445    1472    1476    1498    1502    1524    1528    1550    1554    1576    1580    1602    1606    1628
          1634    1660    1663    1682    1687    1695    1704    1708    1732    1736    1760    1764    1788    1792    1816
          1820    1844    1848    1872    1876    1898    1902    1924    1928    1950    1954    1976    1980    2002    2006
          2028    2032    2054    2058    2080    2084    2117    2121    2155    2162    2170    2181    2185    2207    2211
          2233    2237    2259    2263    2285    2289    2311    2315    2337    2341    2363    2367    2389    2397    2434
          2439    2493    2498    2536    2541    2647    2657    2705    2710    2725    2771    2776    2791    2836    2851
          2853    2858    2882    2894    2906    2915    2924    2933    2942    2951    2960    2964    2982    2990    3002
          3011    3020    3029    3040    3047    3091    3102    3160    3164    3304    3311    3362    3369    3421    3425
          3539    3551    3583    3595    3770    3776    3810    3816    3870    3884    4118    4125    4169    4174    4272
          4279    4281    4289    4353    4357    4401    4406    4683    4689    4728    4734    4773    4779    4818    4821
          4858    4861    4898    4901    4939    4944    4980    4984    5005    5009    5030    5034    5055    5059    5081
          5086    5113    5118    5163    5171    5190    5196    5206    5214    5233    5238    5250    5257    5276    5285
          5296    5303    5322    5330    5343    5350
.IFTF        1
.IIF      1219    1220    1250    1251    1318    1319    1321    1323    1328    1344    1345    1347    1349    1354    1370
          1371    1373    1375    1380    1396    1397    1399    1401    1406    1422    1423    1425    1427    1432    1450
          1451    1481    1482    1484    1486    1491    1507    1508    1510    1512    1517    1533    1534    1536    1538
          1543    1559    1560    1562    1564    1569    1585    1586    1588    1590    1595    1611    1612    1614    1616
          1621    1639    1640    1668    1669    1692    1693    1713    1714    1716    1718    1724    1741    1742    1744
          1746    1752    1769    1770    1772    1774    1780    1797    1798    1800    1802    1808    1825    1826    1828
          1830    1836    1853    1854    1856    1858    1864    1881    1882    1884    1886    1891    1907    1908    1910
          1912    1917    1933    1934    1936    1938    1943    1959    1960    1962    1964    1969    1985    1986    1988
          1990    1995    2011    2012    2014    2016    2021    2037    2038    2040    2042    2047    2063    2064    2066
          2068    2073    2090    2091    2127    2128    2167    2168    2190    2191    2193    2195    2200    2216    2217
          2219    2221    2226    2242    2243    2245    2247    2252    2268    2269    2271    2273    2278    2294    2295
          2297    2299    2304    2320    2321    2323    2325    2330    2346    2347    2349    2351    2356    2372    2373
          2375    2377    2382    2402    2403    2444    2445    2503    2504    2546    2548    2663    2664    2715    2716
          2781    2782    2863    2864    2899    2900    2969    2970    2995    2996    3052    3053    3062    3063    3072
```

```
DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 135                                                    PAGE: 0157
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

            3073    3083    3084    3107    3108    3116    3128    3140    3152    3169    3170    3316    3317    3374    3375
            3430    3431    3556    3557    3600    3601    3781    3782    3821    3822    3889    3890    4130    4131    4179
            4180    4294    4295    4362    4363    4411    4412    4694    4695    4739    4740    4784    4785    4826    4827
            4839    4866    4867    4879    4906    4907    4919    4949    4950    4989    4990    4999    5014    5015    5024
            5039    5040    5049    5064    5065    5074    5091    5092    5123    5124    5168    5169    5170    5171    5176
            5177    5182    5183    5185    5186    5195    5196    5200    5201    5211    5212    5213    5219    5220    5225
            5228    5229    5244    5255    5256    5257    5262    5263    5268    5269    5271    5284    5301    5302    5308
            5309    5314    5317    5355    5356
.IRP         191     193     195     197     199     201     203     205     207     209     211     213     215     217     219
             281    1211    1216    1219    1240    1247    1250    1308    1309    1315    1318    1335    1341    1344    1361
            1367    1370    1387    1393    1396    1413    1419    1422    1439    1447    1450    1471    1472    1478    1481
            1498    1504    1507    1524    1530    1533    1550    1556    1559    1576    1582    1585    1602    1608    1611
            1628    1636    1639    1660    1665    1668    1682    1689    1692    1703    1704    1710    1713    1732    1738
            1741    1760    1766    1769    1788    1794    1797    1816    1822    1825    1844    1850    1853    1871    1872
            1878    1881    1898    1904    1907    1924    1930    1933    1950    1956    1959    1976    1982    1985    2002
            2008    2011    2028    2034    2037    2054    2060    2063    2080    2087    2090    2117    2124    2127    2155
            2164    2167    2180    2181    2187    2190    2207    2213    2216    2233    2239    2242    2259    2265    2268
            2285    2291    2294    2311    2317    2320    2337    2343    2346    2363    2369    2372    2389    2399    2402
            2434    2441    2444    2493    2500    2503    2536    2543    2547    2647    2660    2663    2705    2712    2715
            2771    2778    2781    2836    2853    2860    2863    2882    2896    2899    2960    2966    2969    2982    2992
            2995    3040    3049    3052    3091    3104    3107    3160    3166    3169    3304    3313    3316    3362    3371
            3374    3421    3427    3430    3539    3553    3556    3583    3597    3600    3770    3778    3781    3810    3818
            3821    3870    3886    3889    4118    4127    4130    4169    4176    4179    4272    4281    4291    4294    4353
            4359    4362    4401    4408    4411    4683    4691    4694    4728    4736    4739    4773    4781    4784    4818
            4823    4826    4858    4863    4866    4898    4903    4906    4939    4946    4949    4980    4986    4989    5005
            5011    5014    5030    5036    5039    5055    5061    5064    5081    5088    5091    5113    5120    5123    5163
            5173    5176    5206    5216    5219    5250    5259    5262    5296    5305    5308    5343    5352    5355    5486
.LIST          1      15      36      55      81      83      93     117     119     193     195     197     199     201     203
             205     207     209     211     213     215     217     219     221     276     370     463     511     958    1211
            1214    1221    1240    1245    1252    1309    1313    1319    1335    1339    1345    1361    1365    1371    1387
            1391    1397    1413    1417    1423    1439    1445    1451    1472    1476    1482    1498    1502    1508    1524
            1528    1534    1550    1554    1560    1576    1580    1586    1602    1606    1612    1628    1634    1640    1660
            1663    1669    1682    1687    1693    1704    1708    1714    1732    1736    1742    1760    1764    1770    1788
            1792    1798    1816    1820    1826    1844    1848    1854    1872    1876    1882    1898    1902    1908    1924
            1928    1934    1950    1954    1960    1976    1980    1986    2002    2006    2012    2028    2032    2038    2054
            2058    2064    2080    2084    2091    2117    2121    2128    2155    2162    2168    2181    2185    2191    2207
            2211    2217    2233    2237    2243    2259    2263    2269    2285    2289    2295    2311    2315    2321    2337
            2341    2347    2363    2367    2373    2389    2397    2403    2434    2439    2446    2493    2498    2505    2536
            2541    2549    2647    2657    2665    2705    2710    2716    2771    2776    2782    2836    2851    2853    2858
            2864    2882    2894    2900    2960    2964    2970    2982    2990    2996    3040    3047    3054    3091    3102
            3108    3160    3164    3171    3304    3311    3317    3362    3369    3375    3421    3425    3432    3539    3551
            3557    3583    3595    3601    3770    3776    3782    3810    3816    3822    3870    3884    3890    4118    4125
            4131    4169    4174    4180    4272    4279    4281    4289    4295    4353    4357    4363    4401    4406    4412
            4683    4689    4695    4728    4734    4740    4773    4779    4785    4818    4821    4827    4858    4861    4867
            4898    4901    4907    4939    4944    4950    4980    4984    4990    5005    5009    5015    5030    5034    5040
            5055    5059    5065    5081    5086    5092    5113    5118    5125    5163    5171    5178    5206    5214    5221
            5250    5257    5264    5296    5303    5310    5343    5350    5356    5486
.MACRO         1
.NLIST         1      15      36      55      81      83      93     117     119     193     195     197     199     201     203
             205     207     209     211     213     215     217     219     221     276     370     463     511     958    1211
            1214    1221    1240    1245    1252    1309    1313    1319    1335    1339    1345    1361    1365    1371    1387
            1391    1397    1413    1417    1423    1439    1445    1451    1472    1476    1482    1498    1502    1508    1524
            1528    1534    1550    1554    1560    1576    1580    1586    1602    1606    1612    1628    1634    1640    1660
            1663    1669    1682    1687    1693    1704    1708    1714    1732    1736    1742    1760    1764    1770    1788
            1792    1798    1816    1820    1826    1844    1848    1854    1872    1876    1882    1898    1902    1908    1924
            1928    1934    1950    1954    1960    1976    1980    1986    2002    2006    2012    2028    2032    2038    2054


DZDVA-B MACY11 27(732)  18-MAR-76  15:06  PAGE 136                                                    PAGE: 0158
DZDVAB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

            2058    2064    2080    2084    2091    2117    2121    2128    2155    2162    2168    2181    2185    2191    2207
            2211    2217    2233    2237    2243    2259    2263    2269    2285    2289    2295    2311    2315    2321    2337
            2341    2347    2363    2367    2373    2389    2397    2403    2434    2439    2446    2493    2498    2505    2536
            2541    2549    2647    2657    2665    2705    2710    2716    2771    2776    2782    2836    2851    2853    2858
            2864    2882    2894    2900    2960    2964    2970    2982    2990    2996    3040    3047    3054    3091    3102
            3108    3160    3164    3171    3304    3311    3317    3362    3369    3375    3421    3425    3432    3539    3551
            3557    3583    3595    3601    3770    3776    3782    3810    3816    3822    3870    3884    3890    4118    4125
            4131    4169    4174    4180    4272    4279    4281    4289    4295    4353    4357    4363    4401    4406    4412
            4683    4689    4695    4728    4734    4740    4773    4779    4785    4818    4821    4827    4858    4861    4867
            4898    4901    4907    4939    4944    4950    4980    4984    4990    5005    5009    5015    5030    5034    5040
            5055    5059    5065    5081    5086    5092    5113    5118    5125    5163    5171    5178    5206    5214    5221
            5250    5257    5264    5296    5303    5310    5343    5350    5356    5486
.PAGE         36      83     172     224     276     370     463     973    1211    2434    4817    5388
.REM         370
.REPT         93
.SBTTL         1      36      83     117     370     463     511    1211
.TITLE        15
.WORD        955    1299    1300    1301    1302    1303    1304    1305    1306


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *DZDVAB,DZDVAB/SOL/CRF=DZDVAB.MAC,DZDVAB.P11
 RUN-TIME: 47 67 12 SECONDS
 RUN-TIME RATIO: 351/128=2.7
 CORE USED:  34K  (67 PAGES)
```