DMC-11

BASIC W/R MICRO PRO

CZDMCC0

AH-8545C-MC

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

```
.REM    &
```

IDENTIFICATION

PRODUCT CODE:	AC-8544C-MC
PRODUCT NAME:	CZDMCCO BSC W/R MICRO-PROC TST
DATE:	SEPTEMBER 1978
MAINTAINER:	DIAGNOSTICS

1. ABSTRACT

THE FUNCTION OF THE DMC11 DIAGNOSTICS IS TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERFIY THAT THERE ARE NO MALFUNCTIONS AND THE ALL OPERATIONS OF THE DMC11 ARE CORRECT IN ITS ENVIRONMENT.

PARAMETERS MUST BE SET UP TO ALERT THE DIAGNOSTICS TO THE DMC11 CONFIGURATION. THESE PARAMETERS ARE CONTAINED IN THE STATUS TABLE AND ARE GENERATED IN TWO WAYS: 1) MANUAL INPUT - THE OPERATOR ANSWERS QUESTIONS. 2) AUTOSIZING - THE PROGRAM DETERMINES THE PARAMETERS AUTOMATICALLY.

CZDMC TESTS THE DMC11 MICRO-PROCESSOR (M8200-YA OR M8200-YB). IT PERFORMS WRITE/READ TESTS ON THE DMC UNIBUS REGISTERS, CHECKS THE MICRO-PROCESSOR OPERATION, CHECKS OUT MAIN MEMORY, SCRATCH PAD MEMORY, THE ALU FUNCTIONS AS WELL AS INTERRUPTS AND NPR OPERATION. CZDMC PERFORMS NO TESTS ON THE LINE UNIT OR ANY CROM DEPENDENT TESTS. IT DOES NOT REQUIRE A LINE UNIT TO RUN. NOTE: THIS DIAGNOSTIC WILL RUN ON A KMC11 (M8204), HOWEVER IT IS NOT ADVISED THAT THIS DIAGNOSTIC BE USED TO CHECK A KMC11, RATHER YOU SHOULD CHECK A KMC11 WITH THE KMC11 DIAGNOSTIC PACKAGE.

CURRENTLY THERE ARE FIVE OFF LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO INSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE.

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE FIVE DIAGNOSTICS ARE:

1. CZDMC [REV] BASIC W/R AND MICRO-PROCESSOR TESTS
2. CZDME [REV] DDCMP MODE LINE UNIT TESTS
3. DZDMF [REV] BITSTUFF LINE UNIT TESTS
4. DZDMG [REV] JUMP AND CROM TESTS
5. DZDMH [REV] FREE-RUNNING TESTS (HEAT TEST TAPE)

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (EXCEPT AN LSI-11) WITH MINIMUM 8K MEMORY
ASR 33 (OR EQUILIVALENT)
DMC11-AR (M8200-YA) OR A DMC11-AL (M8200-YB)

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP LOADER RESIDE. LOCATIONS 1500 THRU 1640; CONTAIN THE ''STATUS TABLE'' INFORMATION WHICH IS GENERATED AT START OF DIAGNOSTICS BY MANUAL INPUT (QUESTIONS) OR AUTOMATICALLY (AUTO-SIZING). THIS AREA IS AN OVERLAY AREA AND SHOULD NOT BE ALTERED BY THE OPERATOR.

3. LOADING PROCEDURE

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 PROCEDURE

A. SET SWITCH REGISTER TO 000200
B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
C. SET SWR TO ZERO FOR 'AUTO SIZING' OR SWR BIT0=1 FOR MANUAL INPUT (QUESTIONS) OR SWR BIT7=1 TO USE EXISTING PARAMETERS SET UP BY A PREVIOUS START OR A PREVIOUSLY RUN DMC11 DIAGNOSTIC.
D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

MAP OF DMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
001500	160010	145310	177777	000000
001510	160020	145320	177777	000000

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC. THE ABOVE IS ONLY AN EXAMPLE. THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. IN THIS EXAMPLE THE TABLE CONTAINS THE INFORMATION AND STATUS OF TWO DMC11'S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

IF THE DIAGNOSTIC WAS STARTED WITH SW00-1 INDICATING MANUAL PARAMETER INPUT THEN THE FOLLOWING SHOWS AN EXAMPLE OF THE QUESTIONS ASKED AND SOME EXAMPLE ANSWERS:

```
HOW MANY DMC11'S TO BE TESTED?1

01
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
DOES MICRO-PROCESSOR HAVE CRAM?  (Y OR N)N
WHICH LINE UNIT?  IF NONE TYPE 'N'',  IF  M8201  TYPE  ''1'',  IF
M8202 TYPE '2''?1
IS THE LOOP BACK CONNECTOR ON?Y
SWITCH PAC#1 (DDCMP LINE#)?377
SWITCH PAC#2 (BM873 BOOT ADD)?377
```

FOLLOWING THE QUESTIONS THE STATUS MAP IS PRINTED OUT AS DESCRIBED ABOVE. THE INFORMATION IN THE MAP REFLECTS THE ANSWERS TO THE QUESTIONS. IF THE DIAGNOSTIC WAS STARTED WITH SW00=0 AND SW07-0 (AUTO-SIZING) THEN NO QUESTIONS ARE ASKED AND ONLY THE STATUS-MAP IS PRINTED OUT. IF AUTO-SIZING IS USED THE STATUS INFORMATION MUST BE VERIFIED TO BE CORRECT (MATCH THE HARDWARE). IF IT DOES NOT MATCH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED WITH SW00=1 AND THE QUESTIONS

```
                  ANSWERED.


          4.1     CONTROL SWITCH SETTINGS

                  SW 15 SET:  HALT ON ERROR
                  SW 14 SET:  LOOP ON CURRENT TEST
                  SW 13 SET:  INHIBIT ERROR PRINT OUT
                  SW 12 SET:  INHIBIT TYPE OUT/ABELL ON ERROR.
                  SW 11 SET:  INHIBIT ITERATIONS.  (QUICK PASS)
                  SW 10 SET:  ESCAPE TO NEXT TEST ON ERROR
                  SW 09 SET:  LOOP WITH CURRENT DATA
                  SW 08 SET:  CATCH ERROR AND LOOP ON IT
                  SW 07 SET:  USE PREVIOUS STATUS TABLE.
                  SW 06 SET:  HALT IN    ROMCLK    ROUTINE    BEFORE    CLOCKING
                              MICRO-PROCESSOR
                  SW 05 SET:  RESERVED
                  SW 04 SET:  RESERVED
                  SW 03 SET:  RESELECT DMC11'S DESIRED ACTIVE
                  SW 02 SET:  LOCK ON SELECTED TEST
                  SW 01 SET:  RESTART PROGRAM AT SELECTED TEST
                  SW 00 SET:  BUILD NEW STATUS TABLE FROM QUESTIONS.  (IF SW07=0
                                AND  SW00 0  A  NEW  STATUS  TABLE  IS  BUILT BY
                                AUTO-SIZING)

                  SWITCH 06 AND 08-15 ARE DYNAMIC AND CAN BE CHANGED  AS  NEEDED
                  WHILE THE DIAGNOSTIC IS RUNNING.  SWITCHES 00-03 AND SWITCH 07
                  ARE STATIC, AND ARE USED ONLY ON STARTING  OR  RESTARTING  THE
                  DIAGNOSTIC.
```

```
4.1.2   SWITCH REGISTER OPTIONS (AT START UP)

        SW 01    RESTART PROGRAM AT  SELECTED  TEST.   IT  IS  STRONGLY
                 SUGGESTED  THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE
                 TRYING TO SELECT A TEST, THE REASON BEING IS THAT  THE
                 PROGRAM  HAS  TO  CLEAR  AREAS  AND SET UP PARAMETERS.
                 WHEN THIS SWITCH IS USED THE DIAGNOSTIC WILL ASK  TEST
                 NO.?   ANSWER BY TYPING THE NUMBER OF THE TEST DESIRED
                 AND CARRIGE RETURN TO BEGIN EXECUTION AT THE  SELECTED
                 TEST.

        SW 02    LOCK ON SELECTED TEST.  THIS  SWITCH  WHEN  USED  WITH
                 SW01  WILL CAUSE THE PROGRAM TO CONSTANTLY LOOP ON THE
                 SELECTED TEST.  HITTING ANY KEY ON  THE  CONSOLE  WILL
                 LET  IT  ADVANCE TO THE NEXT TEST AND LOOP UNTIL A KEY
                 IS HIT AGAIN.  IF  SW02=0  WHEN  SW01  IS  USED.   THE
                 PROGRAM  WILL  BEGIN AT THE SELECTED TEST AND CONTINUE
                 NORMAL OPERATIONS.

        SW 03    RESELECT DMC11'S DESIRED ACTIVE.  PLEASE NOTE  THAT  A
                 MESSAGE  IS  TYPED OUT FOR SETTING THE SWITCH REGISTER
                 EQUAL TO DMC11'S ACTIVE.  THIS MEANS IF THE SYSTEM HAS
                 FOUR  DMC11S;   BITS  00,01,02,03  WILL  BE SET IN LOC
                 'DMACTV'  FROM  THE  SWITCH  REGISTER.    USING   THIS
                 SWITCH(SW00)  ALTERS  THAT  LOCATION;THEREFORE IF FOUR
                 DMC11S ARE IN THE  SYSTEM  ***DO  NOT***  SET  SWITCHS
                 GREATER  THAN SW 03 IN THE UP POSITION.  THIS WOULD BE
                 A FATAL ERROR.  DO NOT SELECT MORE ACTIVE DMC11S  THAN
                 THERE IS INFORMATION ON IN THE STATUS TABLE.

        METHOD: A:       LOAD ADDRESS 200
                B:       START WITH SW 00=1
                C:       PROGRAM WILL TYPE ME[illegible]E
                D:       SET A SWITCH  FOR  EACH  DMC  DESIRED  ACTIVE.
                         EXAMPLE:  IF YOU HAVE 4 DMC'S BUT ONLY WANT TO
                         RUN THE FIRST AND THE LAST SET SWR BITS 0  AND
                         3 = 1.  PRESS CONTINUE
                E:       NUMBER (IF  VALID)  WILL  BE  IN  DATA  LIGHTS
                         (EXCLUDING 11/05)
                F:       SET WITH ANY OTHER  SWITCH  SETTINGS  DESIRED.
                         PRESS CONTINUE.
```

```
4.1.3   DYNAMIC SWITCHES

        ERROR SWITCHES

        1.       SW 12   DELETE PRINT OUT/BELL ON ERROR.
        2.       SW 13   DELETE ERROR PRINTOUT.
        3.       SW 15   HALT ON THE ERROR.
        4.       SW 08   GOTO BEGINNING OF THE TEST(ON ERROR).
        5.       SW 10   GOTO NEXT TEST(ON ERROR).

        SCOPE SWITCHES

        1.  SW06     HALT   IN   ROMCLK   ROUTINE   BEFORE   CLOCKING
                     MICRO-PROCESSOR   INSTRUCTION.   THIS  ALLOWS  THE
                     OPERATOR TO SCOPE A MICRO-PROCESSOR INSTRUCTION IN
                     THE  STATIC  STATE  BEFORE  IT  IS  CLOCKED.   HIT
                     CONTINUE TO RESUME RUNNING.
        2.  SW09     (IF ENABLED BY 'SCOP1') ON AN ERROR;  IF AN '*' IS
                     PRINTED  IN FRONT OF THE TEST NO.  (EX.  *TEST NO.
                     10  )  SW09  IS  INCORPORATED  IN  THAT  TEST  AND
                     THEREFORE  SW09 IS USUALLY THE BEST SWITCH FOR THE
                     SCOPE LOOP (SW14=0, SW10=0, SW09-1,  SW08-0).   IF
                     SW09  IS  NOT ENABELED;  AND THERE IS A HARD ERROR
                     (CONSTANT);  SW08  IS  BEST.   (SW14=1,0,  SW10-0,
                     SW09=0, SW08=1).  FOR INTERMITTEMT ERRORS;  SW14-1
                     WILL LOOP ON TEST  REGUARDLESS  OF  ERROR  OR  NOT
                     ERROR.  (SW14=1, SW10=0, SW09 0, SW08-1,0)
        3.  SW11     INHIBIT INTERATIONS.
        4.  SW14     LOOP ON CURRENT TEST.

4.2     STARTING ADDRESS

        STARTING ADDRESS IS AT 000200  THERE  ARE  NO  OTHER  STARTING
        ADDRESSES FOR THE DMC11 DIAGNOSTICS.  (SEE SECTION 4.0)

        NOTE:    IF ADDRESS 000042 IS NON-ZERO THE PROGRAM  ASSUMES  IT
                 IS   UNDER   ACT11   OR   XXDP  CONTROL  AND  WILL  ACT
                 ACCORDINGLY AFTER ALL AVAILABLE DMC11'S ARE TESTED THE
                 PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

5.      OPERATING PROCEDURE

        WHEN PROGRAM IS INITIALLY STARTED  MESSAGES  AS  DESCRIBED  IN
        SECTION  4.0  WILL  BE PRINTED, AND PROGRAM WILL BEGIN RUNNING
        THE DIAGNOSTIC
```

```
5.2     PROGRAM AND/OR OPERATOR ACTION

        THE TYPICAL APPROACH SHOULD BE

        1.      HALT ON ERROR (VIA SW 15 1) WHEN EVER AN ERROR OCCURS.
        2.      CLEAR SW 15.
        3.      SET SW 14:  (LOOP ON THIS TEST)
        4.      SET SW 13:  (INHIBIT ERROR PRINT OUT)

        THE TEST NUMBER AND PC WILL BE  TYPED  OUT  AND  POSSIBILY  AN
        ERROR  MESSAGE (THIS DEPENDS ON THE TEST) TO GIVE THE OPERATOR
        AN IDEA AS TO THE SOURCE OF THE PROBLEM.  IF IT  IS  NECESSARY
        TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT;  LOOK IN
        THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND  THEN
        NOTE THE PC OF THE ERROR REPORT THIS WAY THE EXACT FUNCTION OF
        THE TEST CAN BE DETERMINED.

6.      ERRORS

        AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND
        PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13 0 AND SW
        12=0).  IN MOST CASES ADDITIONAL INFORMATION WILL BE  SUPPLIED
        IN THE THE ERROR MESSAGE TO GIVE THE OPERATOR AN INDICATION OF
        THE ERROR.

6.2     ERROR RECOVERY

        IF FOR SOME REASON THE  DMC11  SHOULD  'HANG  THE  BUS'  (GAIN
        CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED)
        AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR  TO  REGAIN
        CONTROL  OF  CPU.   IF  THIS  SHOULD HAPPEN;  LOOK IN LOCATION
        'TSTNO' (ADDRESS 1226)FOR THE NUMBER  OF  THE  TEST  THAT  WAS
        RUNNING  AT  THE  TIME OF THE CATASTROPHIC ERROR.  IN THIS WAY
        THE OPERATOR WILL HAVE AN IDEA AS TO WHAT THE DMC11 WAS  DOING
        AT THE TIME OF THE ERROR.

7.      RESTRICTIONS

7.1     STARTING RESTRICTIONS

        SEE SECTION 4.  (PLEASE)
        STATUS TABLE SHOULD BE VERIFIED REGUARDLESS OF HOW PROGRAM WAS
        STARTED.   ALSO IT IS IMPORTANT TO USE THIS LISTING ALONG WITH
        THE INFORMATION  PRINTED  ON  THE  TTY  TO  COMPLETLY  ISOLATE
        PROBLEMS.
```

```
7.2     OPERATING RESTRICTIONS

        THE FIRST TIME A DMC11 DIAGNOSTIC IS LOADED INTO CORE AND  RUN
        THE STATUS TABLE MUST BE SET UP.  THIS IS DONE BY MANUAL INPUT
        (SW00=1) OR BY AUTOSIZING  (SW00=0  AND  SW07=0).   THEREAFTER
        HOWEVER  THE  STATUS  TABLE  NEED  NOT  BE SETUP BY SUBSEQUENT
        RESTARTS OR EVEN LOADING THE NEXT DMC DIAGNOSTIC  BECAUSE  THE
        STATUS  TABLE  IS  OVERLAYED.   THE  CURRENT PARAMETERS IN THE
        STATUS TABLE ARE USED WHEN SW07=1 ON START UP.

7.3     HARDWARE CONFIGURATION RESTRICTIONS

        DMC11(M8200)- JUMPER W1 MUST BE IN, AND SWITCH 7 OF  E76  MUST
        BE IN THE OFF POSITION.

        KMC(M8204)- JUMPER W1 MUST BE IN.

8.      MISCELLANEOUS

8.1     EXECUTION TIME

        ALL DMC11 DEVICE DIAGNOSTICS WILL GIVE AN 'END  PASS'  MESSAGE
        (PROVIDING  NO  ERRORS  AND  SW12=0)  WITHIN  4 MINS.  THIS IS
        ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST
        POSSIBLE EXECUTION.  THE ACTUAL EXECUTION TIME DEPENDS GREATLY
        ON THE PDP11 CPU CONFIGURATION AND THE AMOUNT OF MEMORY IN THE
        SYSTEM.

8.2     PASS COMPLETE

        NOTE:  EVERY TIME THE PROGRAM IS STARTED;  THE TESTS WILL  RUN
        AS  IF  SW11  (DELETE  ITERATIONS)  WAS  UP  ( 1).  THIS IS TO
        'VERIFY NO HARD ERRORS' AS SOON AS  POSSIBLE.   THEREFORE  THE
        FIRST  PASS  -EACH  TIME  PROGRAM IS STARTED- WILL BE A 'QUICK
        PASS' UNTIL ALL  DMC11'S  IN  SYSTEM  ARE  TESTED.   WHEN  THE
        DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS AN EXAMPLE OF
        THE PRINT OUT TO BE EXPECTED.

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

        NOTE:    THE PASS COUNT AND ERROR COUNTS  ARE  CUMMULITIVE  FOR
                 EACH  DMC11  THAT IS RUNNING, AND ARE SET TO ZERO ONLY
                 WHEN THE DIAGNOSTIC IS STARTED.   THEREFORE  AFTER  AN
                 OVERNIGHT RUN FOR EXAMPLE, THE TOTAL PASSES AND ERRORS
                 FOR EACH DMC11 SINCE THE DIAGNOSTIC  WAS  STARTED  ARE
                 REFLECTED IN PASSES:  AND ERRORS:.
```

8.4 KEY LOCATIONS

RETURN (1214) CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT (1216) CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PEFORMED.

TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW BEING PEFORMED.

RUN (1316) THE BIT IN 'RUN' ALWAYS POINTS TO THE DMC11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) 1302/0000000001000000 MEANS THAT DMC11 NO.06 IS THE DMC11 NOW RUNNING.

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

THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 16 (DECIMAL) DMC11S SEQUENTIALY. THEY CONTAIN THE CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DMC11.

DMACTV (1306) EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DMC11 WILL BE TESTED IN TURN. EXAMPLE: (DMACTV) 1276/0000000000011111 MEANS THAT DMC11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DMACTV) 1276/0000000000010001 MEANS THAT DMC11 NO. 00,04 WILL BE TESTED.

DMCSR (1404) CONTAINS THE CSR OF THE CURRENT DMC11 UNDER TEST.

8.4A 'STATUS TABLE' (1500-1640)

THE TABLE IS FILLED BY AUTO SIZING OR BY THE MANUAL PARAMETER INPUT (QUESTIONS) AS DESCRIBED PREVIOUSLY. ALSO IF DESIRED BY USER; THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION.

THE EXAMPLE STATUS MAP SHOWN BELOW CONTAINS INFORMATION FOR TWO DMC11'S. THE TABLE CAN CONTAIN UP TO 16 DMC11'S. FOLLOWING THE MAP IS A DESCRIPTION OF THE BITS FOR EACH MAP ENTRY

MAP OF DMC11 STATUS

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

```
EACH MAP ENTRY CONTAINS  4  WORDS  WHICH  CONTAIN  THE  STATUS
INFORMATION  FOR  1  DMC11.  THE PC SHOWS WHERE IN CORE MEMORY
THE FIRST OF THE 4 WORDS IS.  IN THE EXAMPLE ABOVE  THE  FIRST
DMC'S STATUS IS IN LOCATIONS, 1500, 1502, 1504, AND 1506.  THE
SECOND DMC STATUS IS LOCATED AT 1510, 1512,  1514,  AND  1516.
THE  INFORMATION  CONTAINED IN EACH 4 WORD ENTRY IS DEFINED AS
FOLLOWS:

CSR:     CONTAINS DMC11 CSR ADDRESS

STAT1:   BITS 00-08 IS DMC11 VECTOR ADDRESS
         BIT15=1 MICRO-PROCESSOR HAS CRAM
         BIT15=0 MICRO-PROCESSOR HAS CROM
         BIT14=1 TURNAROUND CONNECTOR IS ON
         BIT14=0 NO TURNAROUND CONNECTOR
         BIT13=0 LINE UNIT IS AN M8201
         BIT13=1 LINE UNIT IS AN M8202
         BIT12=1 NO LINE UNIT
         BITS 09-11 IS DMC11 BR PRIORITY LEVEL

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

STAT3:   BIT0=1 RUN FREE RUNNING TESTS ON KMC11
         BIT1=0 DMC11-AR (LOW SPEED)
         BIT1=1 DMC11-AL (HIGH SPEED)
```

8.5 METHOD OF AUTO SIZING

8.5.1 FINDINC THE CONTROL STATUS REGISTER.

THE AUTO-SIZING ROUTINE FINDS A DMC11 AS FOLLOWS: IT STARTS AT ADDRESS 160000 AND TESTS ALL ADDRESS IN INCREMENTS OF 10 UP TO AND INCLUDING ADDRESS 167760. IF THE ADDRESS DOES NOT TIME OUT, THE FOLLOWING IS DONE, THE FIRST CROM ADDRESS IS WRITTEN TO A 125252 THEN IT IS READ BACK. IF IT CONTAINS A -1 OR 125252 OR 626 OR 16520 A DMC11 OR KMC11 HAS BEEN FOUND, IF NOT, THE ADDRESS IS UPDATED BY 10 AND THE SEARCH CONTINUES. A -1 INDICATES A DMC11 WITH NO CROM, A 125252 INDICATES A KMC11 WITH CRAM, A 626 INDICATES A DMC11-AL AND A 16520 INDICATES A DMC11-AR. FURTHER TESTS ARE PERFORMED AT THIS POINT TO DETERMINE WHICH LINE UNIT, IF ANY, IS INSTALLED, IF A LOOP-BACK CONNECTOR IS INSTALLED AND VARIOUS SWITCH SETTINGS ON THE LINE UNIT. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. ALL DMC11'S IN THE SYSTEM WILL BE FOUND BY THE AUTO-SIZER. IF IT DOES NOT FIND A DMC11 THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS ANSWERED.

8.5.2 FINDING THE VECTOR AND BR LEVEL

THE VECTOR AREA (ADDRESS 300-776) IS FILLED WITH THE INSTRUCTION IOT AND '.+2' (NEXT ADDRESS). THE PROCESSOR STATUS IS STARTED AT 7 AND THE DMC IS PROGRAMMED TO INTERRUPT. THE PS IS LOWERED BY 1 UNTIL THE DMC INTERRUPTS, A DELAY IS MADE AND IF NO INTERUPT OCCURES AT PS LEVEL 3 (BECAUSE OF A BAD DMC11) THE PROGRAM ASSUMES VECTOR ADDRESS 300 AT BR LEVEL 5 AND THE PROBLEM SHOULD BE FIXED IN THE DIAGNOSTIC. ONCE THE PROBLEM IS FIXED; THE PROGRAM SHOULD BE RE-SETUP AGAIN TO GET CORRECT VECTOR. IF AN INTERUPT OCCURED; THE ADDRESS TO WHICH THE DMC11 INTERUPTED TO IS PICKED UP AND REPORTED AS THE VECTOR. NOTE: IF THE VECTOR REPORTED IS NOT THE VECTOR SET UP BY YOU; THERE IS A PROBLEM AND AUTO SIZING SHOULD NOT BE DONE.

8.6 SOFTWARE SWITCH REGISTER

IF THE DIAGNOSTIC IS RUN ON AN 11/04 OR OTHER CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED TO ALLOW USER THE SAME SWITCH OPTIONS AS DESCRIBED PREVIOUSLY. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THIS SOFTWARE SWITCH REGISTER IS USED.

CONTROL:

TO OBTAIN CONTROL AT ANY ALLOWABLE TIME DURING EXECUTION OF THE DIAGNOSTIC THE OPERATOR TYPES A CTRL C ON THE CONSOLE TERMINAL KEYBOARD. AS SOON AS THE CTRL G IS RECOGNIZED, BY THE DIAGNOSTIC, THE FOLLOWING MESSAGE WILL BE DISPLAYED:

SWR=XXXXXX NEW?

WHERE XXXXXX IS THE CURRENT CONTENTS OF THE SOFTWARE SWITCH REGISTER IN OCTAL. THE SOFTWARE CONTROL ROUTINE WILL THEN AWAIT OPERATOR ACTION. AT WHICH TIME THE OPERATOR IS REQUIRED TO TYPE ONE OR MORE OF THE LEGAL CHARACTERS: 1) 0 - 7, 2) LINE FEED(<LF>), 3) CARRIAGE RETURN(<CR>), OR 4) CONTROL-U (CTRL U). NO CHECK IS MADE FOR LEGALITY. IF THE INPUT CHARACTER IS NOT A <LF>, <CR>, OR CTRL U IT IS ASSUMED TO BE AN OCTAL DIGIT.

TO CHANGE THE CONTENTS OF THE SSR THE OPERATOR SIMPLY TYPES THE NEW DESIRED VALUE IN OCTAL - LEADING ZEROS NEED NOT BE TYPED. AND TERMINATES THE INPUT STRING WITH A <CR> OR <LF> DEPENDING ON THE PROGRAM ACTION DESIRED AS DESCRIBED BELOW. THE INPUT VALUE WILL BE TRUNCATED TO THE LAST 6 DIGITS TYPED. AT LEAST ONE DIGIT MUST BE TYPED ON ANY GIVEN INPUT STRING PRIOR TO THE TERMINATOR BEFORE A CHANGE TO THE SSR WILL OCCUR.

WHEN THE INPUT STRING IS TERMINATED WITH A <CR> THE DIAGNOSTIC WILL CONTINUE EXECUTION FROM THE POINT AT WHICH IT WAS INTERRUPTED. IF A <CR> IS THE ONLY THING TYPED THE PROGRAM WILL CONTINUE WITHOUT CHANGING THE SSR. THE <LF> DIFFERS FROM THE <CR> BY RESTARTING THE PROGRAM AS IF IT WERE RESTARTED AT ADDRESS 200.

IF A CTRL U IS TYPED AT ANY POINT IN THE INPUT STRING PRIOR TO THE TERMINATOR THE INPUT VALUE WILL BE DISREGARDED AND THE PROMPT DISPLAYED (SWR XXXXXX NEW?).

TO SET THE SSR FOR THE STARTING SWITCHES, FIRST LOAD THE DIAGNOSTIC, THEN HIT CTRL G, THEN START THE DIAGNOSTIC.

```
                        ;*AC-8544C-MC   CZDMCCO BASIC DMC11 CONTROLLER TEST
                        ;*COPYRIGHT 1976,1978, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*----------------------------------------------------------------------

                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;SWR=0  AUTOSIZE DMC11
                        ;SW07=1  USE CURRENT DMC11 PARAMETERS
                        ;SW00=1  INPUT NEW DMC11 PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "AC-8544C-MC   CZDMCCO BASIC DMC11 CONTROLLER TEST"
                        ;PROGRAM WILL TYPE STATUS MAP
                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                        ;AND THEN RESUME TESTING
                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE




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

        100000          SW15=100000             ; 1,HALT ON ERROR
        040000          SW14=40000              ; 1,LOOP ON CURRENT TEST
        020000          SW13=20000              ; 1,INHIBIT ERROR TYPEOUT
        010000          SW12=10000              ; 1,DELETE TYPEOUT/BELL ON ERROR.
        004000          SW11=4000               ; 1,INHIBIT ITERATIONS
        002000          SW10=2000               ; 1,ESCAPE TO NEXT TEST ON ERROR
        001000          SW09=1000               ; 1,LOOP WITH CURRENT DATA
        000400          SW08=400                ; 1,LOOP ON ERROR
        000200          SW07=200                ; 1,USE CURRENT DMC11 PARAMETERS, =0,AUTOSIZE DMC11
        000100          SW06=100                ; 1, HALT BEFORE CLOCKING MICRO-PROCESSOR INSTRUCTION
        000040          SW05=40
        000020          SW04=20
        000010          SW03=10                 ;RESELECT DMC11'S TO BE TESTED (ACTIVE)
        000004          SW02=4                  ;LOCK ON TEST SELECT
        000002          SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
        000001          SW00=1                  ;INPUT DMC11 PARAMETERS
```

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

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

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

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

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

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

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

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

```

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

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

         000024                         . 24
000024   005336                                 .PFAIL                  ;POWER FAIL HANDLER
000026   000340                                 340                     ;SERVICE AT LEVEL 7
000030   004750                                 .HLT                    ;ERROR HANDLER
000032   000340                                 340                     ;SERVICE AT LEVEL 7
000034   004716                                 .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036   000340                                 340                     ;SERVICE AT LEVEL 7
         000040                         . 40
000040   000000                                 0                       ;SAVE FOR ACT-11 OR XXDP
000042   000000                                 0                       ;RETURN ADDRESS IF UNDER ACT-11 OR XXDP
000044   000000                                 0                       ;SAVE FOR ACT-11 OR XXDP
000046   003522                                 $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
         000052                         . 52
000052   000000                                 0                       ;ACT-11 PROGRAM CHARACTERISTICS

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

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


         001000                         .=1000
001000   005377   041501   034055       MTITLE: .ASCII  <377><12>/AC-8544C-MC/<377>
001016   055103   046504   041503               .ASCIZ  /CZDMCCO BASIC DMC11 CONTROLLER TEST/<377>

         001200                         .=1200

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

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

```

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

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

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

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

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

 001236  000000                  STRTSW: 0                       ;SWITCHES AT START OF PROGRAM
 001240  000000                  STAT:   0                       ;DM STATUS WORD STORAGE
 001242  000000                  CLKX:   0
 001244  000000                  MASKX:  0
 001246  000000                  TEMP1:  0                       ;TEMPORARY STORAGE
 001250  000000                  TEMP2:  0                       ;TEMPORARY STORAGE
 001252  000000                  TEMP3:  0                       ;TEMPORARY STORAGE
 001254  000000                  TEMP4:  0                       ;TEMPORARY STORAGE
 001256  000000                  TEMP5:  0                       ;TEMPORARY STORAGE
 001260  000000                  SAVR0:  0                       ;R0 STORAGE
 001262  000000                  SAVR1:  0                       ;R1 STORAGE
 001264  000000                  SAVR2:  0                       ;R2 STORAGE
 001266  000000                  SAVR3:  0                       ;R3 STORAGE
 001270  000000                  SAVR4:  0                       ;R4 STORAGE
 001272  000000                  SAVR5:  0                       ;R5 STORAGE
 001274  000000                  SAVSP:  0                       ;STACK POINTER STORAGE
 001276  000000                  SAVPC:  0                       ;PROGRAM COUNTER STORAGE
 001300  000000                  ZERO:   0
 001302  000001                  ONE:    1
 001304  000000                  MEMLIM: 0                       ;HIGHEST LOCATION FOR NPR'S
 001306  000001                  DMACTV: .BLKW 1                 ;DMC11'S SELECTED ACTIVE.
 001310  000001                  DMNUM:  .BLKW 1                 ;OCTAL NUMBER OF DMC11'S.
 001312  000001                  SAVACT: .BLKW 1                 ;ORIGINAL ACTV  DEVICES
 001314  000001                  SAVNUM: .BLKW 1                 ;WORKABLE NUMBER
 001316  000000                  RUN:    0                       ;POINTER TO RUNNING DEVICE.
                                 .EVEN
 001320  001472                  CREAM:  DM.MAP-6                ;TABLE POINTER.
 001322  001676                  MILK:   CNT.MAP-4               ;TABLE POINTER
```

```


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

 001324      000                 INIFLG: .BYTE   0              ;PROGRAM INITIALIZATION FLAG
 001325      000                 ERRFLG: .BYTE   0              ;ERROR OCCURED FLAG
 001326      000                 LOKFLG: .BYTE   0              ;LOCK ON CURRENT TEST FLAG
 001327      000                 QV.FLG: .BYTE   0              ;QUICK VERIFY FLAG.
                                                                ;ON FIRST PASS OF EACH DMC11 ITERATIONS WILL BE SUPPRESS
                                 .EVEN

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

                                 ;;*****************************************************************
                                 ;-----------------------------------------------------------------
 001330                          .TRPTAB:
             104400              SCOPE=TRAP+0           ;CALL TO SCOPE LOOP AND ITERATION HANDLER
 001330      003576                      .SCOPE
             104401              SCOP1=TRAP+1           ;CALL TO LOOP ON CURRENT DATA HANDLER
 001332      003736                      .SCOP1
             104402              TYPE=TRAP+2            ;CALL TO TELETYPE OUTPUT ROUTINE
 001334      003766                      .TYPE
             104403              INSTR=TRAP+3           ;CALL TO ASCII STRING INPUT ROUTINE
 001336      004050                      .INSTR
             104404              INSTER=TRAP+4          ;CALL TO INPUT ERROR HANDLER
 001340      004154                      .INSTER
             104405              PARAM=TRAP+5           ;CALL TO NUMERICAL DATA INPUT ROUTINE
 001342      004174                      .PARAM
             104406              SAV05=TRAP+6           ;CALL TO REGISTER SAVE ROUTINE
 001344      004374                      .SAV05
             104407              RES05=TRAP+7           ;CALL TO REGISTER RESTORE ROUTINE
 001346      004434                      .RES05
             104410              CONVRT=TRAP+10         ;CALL TO DATA OUTPUT ROUTINE
 001350      004466                      .CONVRT
             104411              CNVRT=TRAP+11          ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
 001352      004472                      .CNVRT
             104412              MSTCLR=TRAP+12         ;CALL TO ISUE A MASTER CLEAR
 001354      005466                      .MSTCLR
             104413              DELAY=TRAP+13          ;CALL TO DELAY
 001356      005436                      .DELAY
             104414              ROMCLK=TRAP+14         ;CALL TO CLOCK ROM ONCE
 001360      005504                      .ROMCLK
             104415              DATACLK=TRAP+15        ;CALL TO CLK DATA
 001362      005552                      .DATACLK
             104416              TIMER=TRAP+16          ;CALL TO DELAY A CLOCK TICK
 001364      005616                      .TIMER

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

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

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

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

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

                          ; EMP STORAGE
                          ;------------

001416  000000            TEMP:   0
        001460            . .+40

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

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

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

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

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

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

```
001550  000001      DMCR05: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 05
001552  000001      DMS105: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 05
001554  000001      DMS205: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 05
001556  000001      DMS305: .BLKW   1       ;3RD STATUS WORD

001560  000001      DMCR06: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 06
001562  000001      DMS106: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 06
001564  000001      DMS206: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 06
001566  000001      DMS306: .BLKW   1       ;3RD STATUS WORD

001570  000001      DMCR07: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 07
001572  000001      DMS107: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 07
001574  000001      DMS207: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 07
001576  000001      DMS307: .BLKW   1       ;3RD STATUS WORD

001600  000001      DMCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 10
001602  000001      DMS110: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 10
001604  000001      DMS210: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 10
001606  000001      DMS310: .BLKW   1       ;3RD STATUS WORD

001610  000001      DMCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 11
001612  000001      DMS111: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 11
001614  000001      DMS211: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 11
001616  000001      DMS311: .BLKW   1       ;3RD STATUS WORD

001620  000001      DMCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 12
001622  000001      DMS112: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 12
001624  000001      DMS212: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 12
001626  000001      DMS312: .BLKW   1       ;3RD STATUS WORD

001630  000001      DMCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 13
001632  000001      DMS113: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 13
001634  000001      DMS213: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 13
001636  000001      DMS313: .BLKW   1       ;3RD STATUS WORD

001640  000001      DMCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 14
001642  000001      DMS114: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 14
001644  000001      DMS214: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 14
001646  000001      DMS314: .BLKW   1       ;3RD STATUS WORD

001650  000001      DMCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 15
001652  000001      DMS115: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 15
001654  000001      DMS215: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 15
001656  000001      DMS315: .BLKW   1       ;3RD STATUS WORD

001660  000001      DMCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 16
001662  000001      DMS116: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 16
001664  000001      DMS216: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 16
001666  000001      DMS316: .BLKW   1       ;3RD STATUS WORD

001670  000001      DMCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 17
001672  000001      DMS117: .BLKW   1       ;VECTOR FOR DMC11 NUMBER 17
001674  000001      DMS217: .BLKW   1       ;DDCMP LINE# FOR DMC11 NUMBER 17
001676  000001      DMS317: .BLKW   1       ;3RD STATUS WORD

001700  000000      DM.END: 000000
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1000

```
                        FORMAT OF STATUS TABLE
                        ----------------------

   15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
  ----------------------------------------------------------------
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  I C   O   N   T   R   O   L       R   E   G   I   S   T   E   R I   CSR
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  ----------------------------------------------------------------

  ----------------------------------------------------------------
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  I * I * I * I * I * I * I * I *   .   V   E   C   T   O   R   * I   STAT1
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  ----------------------------------------------------------------

  ----------------------------------------------------------------
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  I *   B   M       A   D   D   * I *   L   I   N   E       #   * I   STAT2
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  ----------------------------------------------------------------

  ----------------------------------------------------------------
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I * I * I   STAT3
  I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I   I
  ----------------------------------------------------------------
```

```
                DEFINITION OF FORMAT
                --------------------

        CSR:    CONTAINS DMC11 CSR ADDRESS

        STAT1:  BITS 00-08 IS DMC11 VECTOR ADDRESS
                BIT15=1 MICRO-PROCESSOR HAS CRAM
                BIT15=0 MICRO-PROCESSOR HAS CROM
                BIT14=1 ???? TURNAROUND CONNECTOR IS ON
                BIT14=0 NO TURNAROUND CONNECTOR
                BIT13=0 LINE UNIT IS AN M8201
                BIT13-1 LINE UNIT IS AN M8202
                BIT12=1 NO LINE UNIT
                BITS 09-11 IS DMC11 BR PRIORITY LEVEL

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

        STAT3:  BIT0=1 DO FREE RUNNING TESTS ON KMC
                (MUST BE SET TO A ONE MANUALLY [PROGRAM DZDMI ONLY])
                KMC MUST HAVE MICRO-CODE WRITTEN FROM RUNNING
                DZDMG TEST 2 FIRST
                BIT1-1 DMC11-AL LOCAL HIGH SPEED MICRO-CODE
                BIT1 0 DMC11-AR REMOTE LOW SPEED MICRO-CODE
```

```

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

002002  012737  000340  177776  .START: MOV     #340,PS          ;LOCK OUT INTERRUPTS
002010  012706  001200                  MOV     #STACK,SP        ;SET UP STACK
002014  012737  005336  000024          MOV     #.PFAIL,@#24     ;SET UP POWER FAIL VECTOR
002022  013737  001310  001314          MOV     DMNUM,SAVNUM     ;SAVE NUMBER OF DEVICES IN SYSTEM.
002030  005037  010016                  CLR     SWFLG            ;CLEAR SOFT TYPEOUT FLAG
002034  105037  001325                  CLRB    ERRFLG           ;CLEAR ERROR FLAG
002040  105037  001327                  CLRB    QV.FLG           ;ZERO QUICK VERIFY FLAG
002044  012737  001470  001320          MOV     #DM.MAP-10,CREAM;GET MAP POINTER.
002052  012737  001676  001322          MOV     #CNT.MAP-4,MILK  ;GET PASS COUNT MAP POINTER
002060  012737  100000  001316          MOV     #BIT15,RUN       ;POINT POINTER TO FIRST DEVICE.
002066  012700  001702                  MOV     #CNT.MAP,R0      ;PASS COUNT POINTER TO R0
002072  005020                  23$:    CLR     (R0)+            ;CLEAR TABLE
002074  022700  002002                  CMP     #CNT.MAP+100,R0  ;DONE YET?
002100  001374                          BNE     23$              ;KEEP GOING
002102  005037  001234                  CLR     LSTERR           ;CLEAR LAST ERROR POINTER
002106  012737  000001  001226          MOV     #1,TSTNO         ;SET UP FOR TEST 1
002114  012737  002002  001214          MOV     #.START,RETURN   ;SET UP FOR POWER FAIL BEFORE
                                                                 ;TESTING STARTS
002122  013746  000006                  MOV     @#6,-(SP)        ;SAVE CURRENT VECTORS
002126  013746  000004                  MOV     @#4,-(SP)        ;
002132  012737  002166  000004          MOV     #6$,@#4          ;SET UP FOR TIMEOUT
002140  012737  177570  001202          MOV     #177570,SWR      ;SET SWR TO HARD SWR ADDRESS
002146  012737  177570  001200          MOV     #177570,DISPLAY  ;SET DISPLAY TO HARD SWR ADDRESS
002154  022777  177777  177020          CMP     #-1,@SWR         ;REFERENCE HARDWARE SWITCH REGISTER
002162  001402                          BEQ     6$+2             ;IF = -1 USE SOFT SWR ANYWAY
002164  000407                          BR      7$               ;IF IT EXISTS AND NOT   -1 USE HARD SWR
002166  022626                  6$:     CMP     (SP)+,(SP)+      ;ADJUST STACK
002170  012737  000176  001202          MOV     #SWREG,SWR       ;POINTER TO SOFT SWR
002176  012737  000174  001200          MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002204  012637  000004          7$:     MOV     (SP)+,@#4        ;RESTORE VECTORS
002210  012637  000006                  MOV     (SP)+,@#6        ;
002214  105737  001324                  TSTB    INIFLG           ;HAS INITIALIZATION BEEN PERFORMED
002220  001006                          BNE     20$              ;BR IF YES
002222  022737  003522  000042          CMP     #$ENDAD,@#42     ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002230  001402                          BEQ     20$
002232  104402  001000                  TYPE    ,MTITLE          ;TYPE TITLE MESSAGE
002236  004737  007606          20$:    JSR     PC,CKSWR         ;CHECK FOR SOFT SWR
002242  017737  176734  001236          MOV     @SWR,STRTSW      ;STORE STARTING SWITCHES
002250  005737  000042                  TST     @#42             ;IS IT RUNNING IN AUTO MODE?
002254  001402                          BEQ     .+6              ;BR IF NO
002256  005037  001236                  CLR     STRTSW           ;IF YES, CLEAR SWITCHES
002262  032737  000001  001236          BIT     #SW00,STRTSW     ;IF SW00=1, QUESTIONS ARE ASKED.
002270  001012                          BNE     17$              ;BR IF SW00=1
002272  105737  001236                  TSTB    STRTSW           ;BIT7=1??
002276  100007                          BPL     17$              ;BR IF SW07=0
002300  005737  001306                  TST     DMACTV           ;ARE ANY DEVICES SELECTED?
002304  001006                          BNE     16$              ;BR IF YES
002306  104402  007154                  TYPE,   NOACT            ;NO DEVICES SELECTED.
```

```
002312  000000                            HALT                        ;STOP THE SHOW
002314  000776                            BR      .-2                 ;DISQUALIFY CONTINUE SWITCH
002316  004737  010512            17$:    JSR     PC,AUTO.SIZE        ;GO DO THE AUTO SIZE
002322  105737  001324            14$     TSTB    INIFLG              ;FIRST TIME?
002326  001410                            BEQ     21$                 ;BR IF YES
002330  105737  001236                    TSTB    STRTSW              ;IF USING SAME PARAMETERS DONT TYPE MAP
002334  100431                            BMI     1$
002336  032737  000006  001236            BIT     #BIT1.BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
002344  001403                            BEQ     24$                 ;IF NO THEN TYPE STATUS
002346  000424                            BR      1$                  ;IF YES DO NOT TYPE STATUS
002350  005137  001324            21$:    COM     INIFLG              ;SET FLAG
002354  104402  006224            24$:    TYPE    ,XHEAD              ;TYPE HEADER
002360  012704  001500                    MOV     #DM.MAP,R4          ;SET POINTER
002364  010437  001246            5$:     MOV     R4,TEMP1            ;SET ADDRESS
002370  012437  001250                    MOV     (R4)+,TEMP2         ;SET CSR
002374  001411                            BEQ     1$                  ;ALL DONE IF ZERO
002376  012437  001252                    MOV     (R4)+,TEMP3         ;SET STAT1
002402  012437  001254                    MOV     (R4)+,TEMP4         ;SET STAT2
002406  012437  001256                    MOV     (R4)+,TEMP5         ;SET STAT3
002412  104410                            CONVRT                      ;TYPE OUT STATUS MAP
002414  007454                            XSTATQ                      ;
002416  000762                            BR      5$
002420  012700  001500            1$:     MOV     #DM.MAP,R0          ;R0 POINTS TO STATUS TABLE

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

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

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

```
002760  104402                   3$:     TYPE
002762  007330                           KMCM
002764  022626                   4$:     CMP     (SP)+,(SP)+        ;ADJUST STACK
002766  000727                           BR      OK                 ;BR TO GET OUT
002770  000001                   ERRPC:  1
002772     006     002                   .BYTE   6,2
002774  001276                           SAVPC
002776  000002                   CONTAB: 2
003000     006     004                   .BYTE   6,4
003002  003022                           XLOC
003004     006     002                   .BYTE   6,2
003006  001404                           DMCSR
003010     007                   DEVTAB: .BYTE   7                  ;DJ
003011     017                           .BYTE   17                 ;DH
003012     007                           .BYTE   7                  ;DQ
003013     007                           .BYTE   7                  ;DU
003014     007                           .BYTE   7                  ;DUP
003015     007                           .BYTE   7                  ;LK
003016     007                           .BYTE   7                  ;DMC
003017     007                           .BYTE   7                  ;DZ
003020     007                           .BYTE   7                  ;KMC
        003022                   .EVEN
003022  000000                   XLOC:   0
003024  005705                   AUDONE: TST     R5                 ;DMC?
003026  001005                           BNE     1$                 ;BR IF KMC AND ALL DONE
003030  012705  177777                   MOV     #-1,R5             ;SET R5 TO -1 (KMC)
003034  012700  001500                   MOV     #DM.MAP,R0         ;RESET R0 TO START OF TABLE
003040  000602                           BR      AUSTRT             ;GO DO KMC'S
003042  012637  000006           1$:     MOV     (SP)+,@#6          ;RESTORE LOC 6
003046  012637  000004                   MOV     (SP)+,@#4          ;RESTORE LOC 4
003052  032737  000010  001236           BIT     #SW03,STRTSW       ;SELECT SPECIFIC DEVICES??
003060  001422                           BEQ     3$                 ;BR IF NO.
003062  104402  006144                   TYPE    ,MNEW              ;TYPE THE MESSAGE.
003066  005000                           CLR     R0                 ;ZERO DATA LIGHTS
003070  000000                           HALT                       ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
003072  027737  176104  001312           CMP     @SWR,SAVACT        ;IS THE NUMBER VALID?
003100  101404                           BLOS    2$                 ;BR IF NUMBER IS OK.
003102  104402  006005                   TYPE    ,MERR3             ;TELL USER OF INVALID NUMBER.
003106  000000                           HALT                       ;STOP EVERY THING.
003110  000776                           BR      .-2                ;RESTART THE PROGRAM AGAIN.
003112  017737  176064  001306   2$:     MOV     @SWR,DMACTV        ;GET NEW DEVICE PATTERN
003120  013700  001306                   MOV     DMACTV,R0          ;SHOW THE USER WHAT HE SELECTED.
003124  000000                           HALT                       ;CONTINUE DYNAMIC SWITCHES.
003126  012700  000300           3$.     MOV     #300,R0            ;PREPARE TO CLEAR THE FLOATING
003132  012701  000302                   MOV     #302,R1            ;VECTOR AREA. 300-776
003136  010120                   4$:     MOV     R1,(R0)+           ;START PUTTING "PC+2 - HALT"
003140  005021                           CLR     (R1)+              ;IN VECTOR AREA.
003142  022021                           CMP     (R0)+,(R1)+        ;POP POINTERS
003144  022700  001000                   CMP     #1000,R0           ;ALL DONE??
003150  001372                           BNE     4$                 ;BR IF NO.

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

003152  012706  001200           .BEGIN: MOV     #STACK,SP          ;SET UP STACK
003156  013746  000006                   MOV     @#6,-(SP)          ;SAVE LOC 6
```

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

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

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

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

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

```
003604  032777  040000  175370            BIT     #BIT14,@SWR       ;"LOOP ON THIS TEST"?
003612  001407                    TTST:   BEQ     1$                ;BR IF NO.  (IF LOCK SW01=1; THIS LOC -240)
003614  000437                            BR      3$                ;GOTO 3$    (IF LOCK SW01=1; THIS LOC -240)
003616  005737  003734                    TST     DONE              ;WAS TKCSR DONE SET?
003622  001434                            BEQ     3$                ;BR IF NO (LOCKED ON TEST)
003624  005037  003734                    CLR     DONE              ;YES, CLEAR FLAG
003630  000415                            BR      2$                ;GO TO NEXT TEST
003632  032777  004000  175342    1$:     BIT     #SW11,@SWR        ;DELETE ITERATION?  (QUICK PASS)
003640  001011                            BNE     2$                ;BR IF YES
003642  105737  001327                    TSTB    QV.FLG            ;HAVE PASSES BEECOMPLETED?
003646  001406                            BEQ     2$                ;BR IF QUICK PASS.
003650  005237  001224                    INC     LPCNT             ;UPDATE ITERATION COUNTER
003654  023737  001224  001222            CMP     LPCNT,ICOUNT      ;ARE ALL ITERATIONS DONE??
003662  101414                            BLOS    3$                ;BR IF NOT YET
003664  105037  001325            2$:     CLRB    ERRFLG            ;PREPARE FOR NEW TEST
003670  005037  001224                    CLR     LPCNT             ;START ICOUNTER AT 0
003674  005037  001220                    CLR     LOCK
003700  012737  000020  001222            MOV     #20,ICOUNT        ;RESET ITERATIONS
003706  013737  001216  001214            MOV     NEXT,RETURN       ;GET NEXT TEST
003714  011600                    3$:     MOV     (SP),R0           ;POP R0 OFF OF THE STACK
003716  022626                            POP2SP                    ;FAKE AN "RTI"
003720  013701  001404                    MOV     DMCSR,R1          ;R1 CONTAINS BASE DMC ADDRESS
003724  000177  175264                    JMP     @RETURN           ;GO DO THE TEST
003730  001407                    BRW:    1407
003732  000437                    BRX:    437
003734  000000                    DONE:   0

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

003736  004737  007606            .SCOP1: JSR     PC,CKSWR          ;CHECK FOR SOFT SWR
003742  032777  001000  175232            BIT     #SW09,@SWR        ;IS SW09=1(SET)?
003750  001405                            BEQ     1$                ;BR IF NOT SET.
003752  005737  001220                    TST     LOCK
003756  001402                            BEQ     1$
003760  013716  001220                    MOV     LOCK,(SP)         ;GOTO THE ADDRESS IN LOCK.
003764  000002                    1$:     RTI                       ;GO BACK.

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

003766  010546                    .TYPE:  MOV     R5,-(SP)          ;SAVE R5 ON THE STACK.
003770  017605  000002                    MOV     @2(SP),R5         ;GET ADDRESS OF MESSAGE.
003774  062766  000002  000002            ADD     #2,2(SP)          ;POP OVER ADDRESS.
004002  005737  010016            4$:     TST     SWFLG             ;SOFT SWR MESSAGE?
004006  001004                            BNE     1$                ;IF YES TYPE IT OUT REGARDLESS OF SW12
004010  032777  010000  175164            BIT     #SW12,@SWR        ;INHIBIT ALL PRINT OUT??
004016  001012                            BNE     3$                ;BR IF NO PRINT OUT WANTED (SW12-1)
004020  105715                    1$:     TSTB    (R5)              ;IS NUMBER MINUS? (MSB 1(BIT7))
004022  100002                            BPL     2$                ;BR IF NUMBER IS PLUS
004024  104402  005672                    TYPE    ,MCRLF            ;TYPE A CR/LF.
004030  105777  175154            2$:     TSTB    @TPCSR            ;TTY READY?
004034  100375                            BPL     2$                ;BR IF NO.
004036  112577  175150                    MOVB    (R5)+,@TPDBR      ;PRINT CURRENT CHAR.
004042  001357                            BNE     4$                ;IF NOT ZERO KEEP PRINTING
004044  012605                    3$:     MOV     (SP)+,R5          ;END OF OUTPUT. RESTORE R5
```

```
004046  000002                          RTI                             ;GO HOME
                                        ;-------------------------

004050  010346                  .INSTR: MOV     R3,-(SP)                ;SAVE R3 ON STACK
004052  010446                          MOV     R4,-(SP)                ;SAVE R4 ON STACK
004054  017637  000004  004072          MOV     @4(SP),.MSG
004062  062766  000002  000004          ADD     #2,4(SP)
004070  104402                  .INST1: TYPE
004072  000000                  .MSG:   0
004074  012704  007502                  MOV     #INBUF,R4
004100  012703  000007                  MOV     #7,R3
004104  105777  175074          1$:     TSTB    @TKCSR
004110  100375                          BPL     1$
004112  117714  175070                  MOVB    @TKDBR,(R4)
004116  142714  000200                  BICB    #200,(R4)
004122  122427  000015                  CMPB    (R4)+,#15
004126  001417                          BEQ     INSTR2
004130  105777  175054          2$:     TSTB    @TPCSR
004134  100375                          BPL     2$
004136  017777  175044  175046          MOV     @TKDBR,@TPDBR
004144  005303                          DEC     R3
004146  001356                          BNE     1$
004150  012604                          MOV     (SP)+,R4
004152  012603                          MOV     (SP)+,R3
004154  104402  005666          .INSTE: TYPE    ,MQM
004160  010346                          MOV     R3,-(SP)
004162  010446                          MOV     R4,-(SP)
004164  000741                          BR      .INST1
004166  012604                  INSTR2: MOV     (SP)+,R4                ;RESTORE R4
004170  012603                          MOV     (SP)+,R3                ;RESTORE R3
004172  000002                          RTI

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

004174  010546                  .PARAM: MOV     R5,-(SP)
004176  010446                          MOV     R4,-(SP)
004200  016605  000004                  MOV     4(SP),R5
004204  012537  004364                  MOV     (R5)+,LOLIM
004210  012537  004366                  MOV     (R5)+,HILIM
004214  012537  004370                  MOV     (R5)+,DEVADR
004220  112537  004372                  MOVB    (R5)+,LOBITS
004224  112537  004373                  MOVB    (R5)+,ADRCNT
004230  010566  000004                  MOV     R5,4(SP)
004234  005005                  PARAM1: CLR     R5
004236  012704  007502                  MOV     #INBUF,R4
004242  122714  000015                  CMPB    #15,(R4)
004246  001420                          BEQ     PARERR
004250  121427  000060          1$:     CMPB    (R4),#60
004254  002415                          BLT     PARERR
004256  121427  000067                  CMPB    (R4),#67
004262  003012                          BGT     PARERR
004264  142714  000060                  BICB    #60,(R4)
004270  152405                          BISB    (R4)+,R5
004272  122714  000015                  CMPB    #15,(R4)
004276  001406                          BEQ     LIMITS
```

```
004300  006305                            ASL     R5
004302  006305                            ASL     R5
004304  006305                            ASL     R5
004306  000760                            BR      1$
004310  104404                   PARERR:  INSTER
004312  000750                            BR      PARAM1

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

004314  020537  004366           LIMITS:  CMP     R5,HILIM
004320  101373                            BHI     PARERR
004322  020537  004364                    CMP     R5,LOLIM
004326  103770                            BLO     PARERR
004330  133705  004372                    BITB    LOBITS,R5
004334  001365                            BNE     PARERR

                                          ;STORE NUMBER AT SPECIFIED ADDRESS

004336  013704  004370                    MOV     DEVADR,R4
004342  010524                   1$:      MOV     R5,(R4)+
004344  062705  000002                    ADD     #2,R5
004350  105337  004373                    DECB    ADRCNT
004354  001372                            BNE     1$
004356  012604                            MOV     (SP)+,R4
004360  012605                            MOV     (SP)+,R5
004362  000002                            RTI
004364  000000                   LOLIM:   0
004366  000000                   HILIM:   0
004370  000000                   DEVADR:  0
004372  000000                   LOBITS:  0
        004373                   ADRCNT=LOBITS+1

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

004374  016637  000004  001276   .SAV05:  MOV     4(SP),SAVPC         ;SAVE R7 (PC)

                                          ;SAVE R0-R5

004402  010537  001272           SV05:    MOV     R5,SAVR5            ;SAVE R5
004406  010437  001270                    MOV     R4,SAVR4            ;SAVE R4
004412  010337  001266                    MOV     R3,SAVR3            ;SAVE R3
004416  010237  001264                    MOV     R2,SAVR2            ;SAVE R2
004422  010137  001262                    MOV     R1,SAVR1            ;SAVE R1
004426  010037  001260                    MOV     R0,SAVR0            ;SAVE R0
004432  000002                            RTI                         ;LEAVE.

                                          ;RESTORE R0-R5

004434  013700  001260           .RES05:  MOV     SAVR0,R0            ;RESTORE R0
004440  013701  001262                    MOV     SAVR1,R1            ;RESTORE R1
004444  013702  001264                    MOV     SAVR2,R2            ;RESTORE R2
004450  013703  001266                    MOV     SAVR3,R3            ;RESTORE R3
004454  013704  001270                    MOV     SAVR4,R4            ;RESTORE R4
004460  013705  001272                    MOV     SAVR5,R5            ;RESTORE R5
```

```
004464 000002                          RTI                     ;LEAVE

                                       ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                       ;-------------------------------------------------------
                                       ;
004466 104402 005672           .CONVR: TYPE    ,MCRLF
004472 010046                  .CNVRT: MOV     R0,-(SP)
004474 010146                          MOV     R1,-(SP)
004476 010346                          MOV     R3,-(SP)
004500 010446                          MOV     R4,-(SP)
004502 010546                          MOV     R5,-(SP)
004504 017601 000012                   MOV     @12(SP),R1
004510 062766 000002 000012            ADD     #2,12(SP)
004516 012137 004710                   MOV     (R1)+,WRDCNT
004522 112137 004712           1$:     MOVB    (R1)+,CHRCNT
004526 112137 004713                   MOVB    (R1)+,SPACNT
004532 013137 004714                   MOV     @(R1)+,BINWRD
004536 122737 000003 004712            CMPB    #3,CHRCNT
004544 001003                          BNE     2$
004546 042737 177400 004714            BIC     #177400,BINWRD
004554 013704 004714           2$:     MOV     BINWRD,R4
004560 113705 004712                   MOVB    CHRCNT,R5
004564 012700 001416                   MOV     #TEMP,R0
004570 010403                  3$:     MOV     R4,R3
004572 042703 177770                   BIC     #177770,R3
004576 062703 000060                   ADD     #060,R3
004602 110320                          MOVB    R3,(R0)+
004604 000241                          CLC
004606 006004                          ROR     R4
004610 000241                          CLC
004612 006004                          ROR     R4
004614 000241                          CLC
004616 006004                          ROR     R4
004620 005305                          DEC     R5
004622 001362                          BNE     3$
004624 012703 007544                   MOV     #MDATA,R3
004630 114023                  4$:     MOVB    -(R0),(R3)+
004632 105337 004712                   DECB    CHRCNT
004636 001374                          BNE     4$
004640 105737 004713                   TSTB    SPACNT
004644 001405                          BEQ     6$
004646 112723 000040           5$:     MOVB    #040,(R3)+
004652 105337 004713                   DECB    SPACNT
004656 001373                          BNE     5$
004660 105013                  6$:     CLRB    (R3)
004662 104402 007544                   TYPE    ,MDATA
004666 005337 004710                   DEC     WRDCNT
004672 001313                          BNE     1$
004674 012605                          MOV     (SP)+,R5
004676 012604                          MOV     (SP)+,R4
004700 012603                          MOV     (SP)+,R3
004702 012601                          MOV     (SP)+,R1
004704 012600                          MOV     (SP)+,R0
004706 000002                          RTI
004710 000000                  WRDCNT: 0
004712 000000                  CHRCNT: 0
```

```
                004713                          SPACNT=CHRCNT+1
        004714  000000                          BINWRD: 0


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

        004716  011646                          .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
        004720  162716  000002                          SUB     #2,(SP)         ;=PC OF TRAP
        004724  017616  000000                          MOV     @(SP),(SP)      ;GET TRP
        004730  006316                          TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
        004732  042716  177001                          BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
        004736  062716  001330                          ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
        004742  017616  000000                          MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
        004746  000136                                  JMP     @(SP)+          ;GO TO SUBROUTINE

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

        004750  004737  007606                  .HLT:   JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
        004754  032777  010000  174220                  BIT     #SW12,@SWR      ;BELL ON ERROR?
        004762  001406                                  BEQ     XBX             ;BR IF NO BELL
        004764  105777  174220                          TSTB    @TPCSR          ;TTY READY.
        004770  100003                                  BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
        004772  112777  000207  174212                  MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
        005000  032777  020000  174174          XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
        005006  001105                                  BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
        005010  021637  001234                          CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
        005014  001404                                  BEQ     1$              ;BR IF YES
        005016  011637  001234                          MOV     (SP),LSTERR     ;RECORD BEING HERE
        005022  105037  001325                          CLRB    ERRFLG          ;PREPARE HEADER
        005026  104406                          1$:     SAV05                   ;SAVE ALL PROC REGISTERS
        005030  011605                                  MOV     (SP),R5         ;GET THE PC OF ERROR
        005032  162705  000002                          SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
        005036  011504                                  MOV     (R5),R4         ;GET HLT INSTRUCTION
        005040  006304                                  ASL     R4              ;MULT BY TWO
        005042  061504                                  ADD     (R5),R4         ;DOUBLE IT
        005044  006304                                  ASL     R4              ;MULT AGAIN
        005046  042704  177001                          BIC     #177001,R4      ;CLEAR JUNK
        005052  062704  036330                          ADD     #.ERRTAB,R4     ;GET POINTER
        005056  012437  005172                          MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
        005062  012437  005204                          MOV     (R4)+,DATAHD    ;GET DATA HEADRER
        005066  011437  005216                          MOV     (R4),DATABP     ;GET DATA TABLE
        005072  105737  001325                          TSTB    ERRFLG          ;TYPE HEADREER
        005076  001403                                  BEQ     TYPMSG          ;BR IF YES
        005100  005737  005216                          TST     DATABP          ;DOES DATA TABLE EXIST?
        005104  001040                                  BNE     TYPDAT          ;BR IF YES.
        005106  104402  005672                  TYPMSG: TYPE    ,MCRLF
        005112  104402  005672                          TYPE    ,MCRLF
        005116  005737  001220                          TST     LOCK
        005122  001402                                  BEQ     1$
        005124  104402  006142                          TYPE    ,MASTEK
        005130  104402  006130                  1$:     TYPE    ,XTSTN
        005134  104411  005330                          CNVRT   ,XTSTN          ;SHOW IT
```

```
005140 104402 006217                   TYPE    ,MERRPC          ;TYPE PC.
005144 104411 005322                   CNVRT   ,ERTABO          ;SHOW IT
005150 104402 005672                   TYPE    ,MCRLF           ;GIVE A CR/LF
005154 112737 177777 001325            MOVB    #-1,ERRFLG       ;NO MORE HEADER UNLESS NO DATA TABLE.
005162 005737 005172                   TST     ERRMSG           ;IS THERE AN ERROR MESSAGE?
005166 001402                          BEQ     WRKO.FM          ;BR IF NO.
005170 104402                          TYPE                     ;TYPE
005172 000000                 ERRMSG:  0                        ;    ERROR MESSAGE
005174                        WRKO.FM:                          ;
005174 005737 005204                   TST     DATAHD           ;DATA HEADER?
005200 001402                          BEQ     TYPDAT           ;BR IF NO
005202 104402                          TYPE                     ;TYPE
005204 000000                 DATAHD:  0                        ;    DATA HEADER
005206 005737 005216          TYPDAT:  TST     DATABP           ;DATA TABLE?
005212 001402                          BEQ     RESREG           ;BR IF NO.
005214 104410                          CONVRT                   ;SHOW
005216 000000                 DATABP:  0                        ;    DATA TABLE
005220 104407                 RESREG:  RES05                    ;RESTORE PROC REGISTERS
005222 022737 003522 000042   HALTS:   CMP     #$ENDAD,@#42     ;IF ACT-11 AUTOMATIC MODE, HALT..
005230 001403                          BEQ     1$
005232 005777 173744                   TST     @SWR             ;HALT ON ERROR?
005236 100005                          BPL     EXITER           ;BR IF NO HALT ON ERROR
005240 010046                 1$:      PUSHRO                   ;SAVE RO
005242 016600 000002                   MOV     2(SP),RO         ;SHOW ERROR PC IN DATA LIGHTS
005246 000000                          HALT                     ;HALT
005250 012600                          POPRO                    ;GET RO
005252 005237 001232          EXITER:  INC     ERRCNT           ;UPDATE ERROR COUNT
005256 032777 000400 173716            BIT     #SW08,@SWR       ;GOTO TOP OF TEST?
005264 001007                          BNE     1$               ;BR IF YES
005266 032777 002000 173706            BIT     #SW10,@SWR       ;GOTO NEXT TEST?
005274 001411                          BEQ     2$               ;BR IF NO
005276 013737 001216 001214            MOV     NEXT,RETURN      ;SET FOR NEXT TEST
005304 012706 001200          1$:      MOV     #STACK,SP        ;RESET SP
005310 013701 001404                   MOV     DMCSR,R1         ;SET UP R1
005314 000177 173674                   JMP     @RETURN          ;GOTO SPECIFIED TEST
005320 000002                 2$:      RTI                      ;RETURN
005322 000001                 ERTABO:  1
005324    006    002                   .BYTE   6,2
005326 001276                          SAVPC
005330 000001                 XTSTN:   1
005332    003    002                   .BYTE   3,2
005334 001226                          TSTNO
                                       ;ENTER HERE ON POWER FAILURE
                                       ;---------------------------
                                       ;

005336                        .PFAIL:
005336 012737 005350 000024            MOV     #RESTART,24              ;SET UP FOR POWER UP TRAP
005344 000000                          HALT                             ;HALT ON POWER DOWN NORMA.
005346 000777                          BR      .

                                       ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

005350                        RESTAR:
005350 012737 005336 000024            MOV     #.PFAIL,24       ;SET UP FOR POWER FAILURE
005356 012706 001200                   MOV     #STACK,SP        ;RESET THE STACK POINTER
```

```
005362  013701  001404                  MOV     DMCSR,R1            ;RESTORE R1
005366  005037  001416                  CLR     TEMP                ;READY FOR TIMMER
005372  005237  001416                  INC     TEMP                ;PLUS ONE TO THE TIMER.
005376  001375                          BNE     .-4                 ;BR IF MORE TO GO
005400  104402  005675                  TYPE    ,MPFAIL             ;TYPE THE MESSAGE
005404  104411  005430                  CNVRT   ,PFTAB              ;TELL WHAT TEST TO RETURN TO.
005410  105037  001325                  CLRB    ERRFLG              ;START CLEAN
005414  005037  001234                  CLR     LSTERR              ;************
005420  005011                          CLR     (R1)                ;CLEAR MAINT BITS
005422  104412                          MSTCLR                      ;START CLEAN UP OF DEVICE
005424  000177  173564                  JMP     @RETURN             ;START DOING THAT TEST AGAIN.
005430  000001                  PFTAB:  1
005432     003     002          .BYTE   3,2
005434  001226                          TSTNO

005436                          .DELAY:
005436  012777  000020  173746          MOV     #20,@DMP04
005444  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
005446  121111                          121111                      ;POKE CLOCK DELAY BIT
005450                          1$:
005450  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
005452  121224                          121224                      ;PORT4 IBUS*11
005454  032777  000020  173730          BIT     #BIT4,@DMP04        ;IS CLOCK BIT SET?
005462  001772                          BEQ     1$                  ;BR IF NO
005464  000002                          RTI

005466                          .MSTCLR:
005466  152777  000100  173712          BISB    #BIT6,@DMCSRH       ;SET MASTER CLEAR
005474  142777  000300  173704          BICB    #BIT6!BIT7,@DMCSRH  ;CLEAR MASTER CLEAR AND RUN
005502  000002                          RTI                         ;RETURN

005504                          .ROMCLK:
005504  152777  000002  173674          BISB    #BIT1,@DMCSRH       ;SET ROMI
005512  013677  173676                  MOV     @(SP)+,@DMP06       ;LOAD INSTRUCTION IN SEL6
005516  062746  000002                  ADD     #2,-(SP)            ;ADJUST STACK
005522  032777  000100  173452          BIT     #SW06,@SWR          ;HALT IF SW06 =1
005530  001401                          BEQ     1$                  ;BR IF SW06 =0
005532  000000                          HALT                        ;HALT BEFORE CLOCKING INSTRUCTION
005534  152777  000003  173644  1$:     BISB    #BIT1!BIT0,@DMCSRH  ;CLOCK INSTRUCTION
005542  142777  000007  173636          BICB    #BIT2!BIT1.BIT0,@DMCSRH   ;CLEAR ROMO, ROMI, STEP
005550  000002                          RTI

005552                          .DATACLK:
005552  013637  001416                  MOV     @(SP)+,TEMP         ;PUT TICK COUNT IN TEMP
005556  062746  000002                  ADD     #2,-(SP)            ;ADJUST STACK
005562  152777  000020  173616  1$:     BISB    #BIT4,@DMCSRH       ;SET STEP LU
005570  027777  173610  173606          CMP     @DMCSR,@DMCSR       ;WASTE TIME
005576  142777  000020  173602          BICB    #BIT4,@DMCSRH       ;CLEAR STEP LU
005604  005337  001416                  DEC     TEMP                ;DEC TICK COUNT
005610  001364                          BNE     1$                  ;BR IF NOT DONE
005612  000002                          RTI                         ;RETURN
005614  000001                  3$:     .BLKW 1

005616                          .TIMER:
005616  013637  001416                  MOV     @(SP)+,TEMP         ;MOVE COUNT TO TEMP
005622  062746  000002                  ADD     #2,-(SP)            ;ADJUST STACK
```

```
      005626                                1$:
      005626  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      005630  021364                                021364                      ;PORT4 IBUS* REG11
      005632  032777  000002  173552                BIT     #2,@DMP04           ;IS PGM CLOCK BIT CLEAR?
      005640  001772                                BEQ     1$                  ;BR IF YES
      005642                                2$:
      005642  104414                                ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      005644  021364                                021364                      ;PORT4 IBUS* REG11
      005646  032777  000002  173536                BIT     #2,@DMP04           ;IS PGM CLOCK BIT SET?
      005654  001372                                BNE     2$                  ;BR IF YES
      005656  005337  001416                        DEC     TEMP                ;DEC COUNT
      005662  001361                                BNE     1$                  ;BR IF NOT DONE
      005664  000002                                RTI                         ;RETURN

      005666  020040  000077          MQM:      .ASCIZ  /  ?/
 (2)  005672  005015     000          MCRLF:    .ASCIZ  <15><12>
 (2)  005675     377  053520  020122  MPFAIL:   .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
 (2)  005733     377  047105  020104  MEPASS:   .ASCIZ  <377>/END PASS CZDMC  /
 (2)  005755     377  000122          MR:       .ASCIZ  <377>/R/
 (2)  005760  047377  020117  042504  MERR2:    .ASCIZ  <377>/NO DEVICES PRESENT./
 (2)  006005     377  047111  052523  MERR3:    .ASCIZ  <377>/INSUFFICIENT DATA./
 (2)  006031     377  042524  052123  MTSTPC:   .ASCIZ  <377>/TEST PC-/
 (2)  006043     377  047514  045503  MLOCK:    .ASCIZ  <377>/LOCK ON SELECTED TEST/
 (2)  006072  051503  035122  000040  MCSRX:    .ASCIZ  /CSR: /
 (2)  006100  042526  035103  000040  MVECX:    .ASCIZ  /VEC: /
 (2)  006106  040520  051523  051505  MPASSX:   .ASCIZ  /PASSES: /
 (2)  006117     105  051122  051117  MERRX:    .ASCIZ  /ERRORS: /
 (2)  006130  042524  052123  047040  MTSTN:    .ASCIZ  /TEST NO: /
 (2)  006142  000052                  MASTEK:   .ASCIZ  /*/
 (2)  006144  051777  052105  051440  MNEW:     .ASCIZ  <377>/SET SWITCH REG TO DMC11'S DESIRED ACTIVE./
 (2)  006217     120  035103  000040  MERRPC:   .ASCIZ  /PC: /
 (2)  006224  020212  020040  020040  XHEAD:    .ASCII  <212>/          MAP OF DMC11 STATUS/
 (2)  006263     377  020040  020040            .ASCII  <377>/          -------------------/
 (2)  006322  020212  050040  020103            .ASCII  <212>/  PC      CSR     STAT1    STAT2    STAT3/
 (2)  006374  026777  026455  026455            .ASCIZ  <377>/------   ------  ------   ------   ------/
 (2)  006450  044377  053517  046440  NUM:      .ASCIZ  <377>/HOW MANY DMC11'S TO BE TESTED?/
 (2)  006510  041777  051123  040440  CSR:      .ASCIZ  <377>/CSR ADDRESS?/
 (2)  006526  053377  041505  047524  VEC:      .ASCIZ  <377>/VECTOR ADDRESS?/
 (2)  006547     377  051102  050040  PRIO:     .ASCIZ  <377>/BR PRIORITY LEVEL? (4,5,6,7)?/
 (2)  006606  044777  020106  046504  CRAM:     .ASCIZ  <377>/IF DMC HAS CRAM (M8204) TYPE 'Y', IF CROM (M8200) TYPE 'N' ?/
 (2)  006704  053777  044510  044103  MODU:     .ASCIZ  <377>/WHICH LINE UNIT? IF NONE TYPE 'N', IF M8201 TYPE '1', IF M8202 TYP
 (2)  007016  051777  044527  041524  LINE:     .ASCIZ  <377>/SWITCH PAC#1 (DDCMP LINE #)?/
 (2)  007054  051777  044527  041524  BM:       .ASCIZ  <377>/SWITCH PAC#2 (BM873 BOOT ADD)?/
 (2)  007114  044777  020123  044124  CONN:     .ASCIZ  <377>/IS THE LOOP BACK CONNECTOR ON?/
 (2)  007154  047377  020117  042504  NOACT:    .ASCIZ  <377>/NO DEVICES ARE SELECTED/
 (2)  007205     377  051412  051127  SWMES:    .ASCIZ  <377><12>/SWR= /
 (2)  007215     116  053505  020077  SWMES1:   .ASCIZ  /NEW? /
 (2)  007223     377  042377  041515  CONERR:   .ASCIZ  <377><377>/DMC11 FOUND AT NON-STANDARD ADDRESS  PC: /
 (2)  007277     377  054105  042520  CNERR:    .ASCIZ  <377>/EXPECTED  FOUND/
 (2)  007320  024040  046504  024503  DMCM:     .ASCIZ  / (DMC) /
 (2)  007330  024040  046513  024503  KMCM:     .ASCIZ  / (KMC) /
 (2)  007340  042377  041515  030461  SPEED:    .ASCIZ  <377>/DMC11-AR(REMOTE,LOW SPEED) OR DMC11-AL(LOCAL,HIGH SPEED) TYPE 'R'
 (2)                                            .EVEN
 (2)  007454  000005                  XSTATQ:   5
      007456     006     003                    .BYTE   6,7
      007460  001246                            TEMP1
```

```
007462     006     003                  .BYTE   6,3
007464  001250                          TEMP2
007466     006     003                  .BYTE   6,3
007470  001252                          TEMP3
007472     006     003                  .BYTE   6,3
007474  001254                          TEMP4
007476     006     002                  .BYTE   6,2
007500  001256                          TEMP5
                                .EVEN

                                        ;BUFFERS FOR INPUT-OUTPUT

007502  000000                  INBUF:  0
        007544                  . .+40
007544  000000                  MDATA:  0
        007606                  . .+40


                                        ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
                                        ;REGISTER USING THE CONSOLE TERMINAL
                                        ;-------------------------------------

007606  022737  000176  001202  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
007614  001077                          BNE     CKSWR5          ;BR IF NO
007616  105777  171362                  TSTB    @TKCSR          ;IS DONE SET?
007622  100003                          BPL     2$              ;GO ON IF NOT SET
007624  012737  177777  003734          MOV     #-1,DONE        ;IF DONE SET, SET FLAG
007632  022777  000007  171346  2$.     CMP     #7,@TKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
007640  001404                          BEQ     1$              ;BR IF YES
007642  022777  000207  171336          CMP     #207,@TKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII)
007650  001061                          BNE     CKSWR5          ;BR IF NO
007652  010246                  1$:     MOV     R2,-(SP)        ;STORE R2
007654  010346                          MOV     R3,-(SP)        ;STORE R3
007656  010446                          MOV     R4,-(SP)        ;STORE R4
007660  012737  177777  010016          MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
007666  005002                  CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
007670  012704  177777                  MOV     #-1,R4          ;SET FLAG TO ALL ONES
007674  104402  007205                  TYPE    ,SWMES          ;TYPE 'SWR= '
007700  104411                  CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
007702  010052                          SOFTSW                  ;OF SOFT SWITCH REGISTER
007704  104402  007215          CKSWR3: TYPE    ,SWMES1         ;TYPE 'NEW? '
007710  004737  010020          CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
007714  022703  000015                  CMP     #15,R3          ;WAS IT A CR?
007720  001424                          BEQ     5$              ;BR IF YES
007722  022703  000012                  CMP     #12,R3          ;WAS IT A LF?
007726  001416                          BEQ     4$              ;BR IF YES
007730  022703  000025                  CMP     #25,R3          ;WAS IT CTRL U?
007734  001754                          BEQ     CKSWR1          ;BR IF YES(START OVER)
007736  022703  000007                  CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
007742  001762                          BEQ     CKSWR4
007744  005004                          CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
007746  042703  177770                  BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
007752  006302                          ASL     R2              ;SHIFT R2 3 TIMES
007754  006302                          ASL     R2
007756  006302                          ASL     R2
007760  050302                          BIS     R3,R2           ;ADD LAST DIGIT
```

```
007762  000752                            BR      CKSWR4          ;GET NEXT CHARACTER
007764  012766  002002  000006  4$:       MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
007772  005704                  5$:       TST     R4              ;IS FLAG CLEAR?
007774  001002                            BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
007776  010277  171200                    MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
010002  005037  010016          6$:       CLR     SWFLG           ;CLEAR TYPEOUT FLAG
010006  012604                            MOV     (SP)+,R4        ;RESTORE R4
010010  012603                            MOV     (SP)+,R3        ;RESTORE R3
010012  012602                            MOV     (SP)+,R2        ;RESTORE R2
010014  000207                  CKSWR5:   RTS     PC              ;RETURN

010016  000000                  SWFLG:    0

010020  105777  171160          INCHAR:   TSTB    @TKCSR
010024  100375                            BPL     .-4
010026  017703  171154                    MOV     @TKDBR,R3
010032  105777  171152                    TSTB    @TPCSR
010036  100375                            BPL     .-4
010040  010377  171146                    MOV     R3,@TPDBR
010044  042703  000200                    BIC     #BIT7,R3
010050  000207                            RTS     PC

010052  000001                  SOFTSW:   1
010054     006     002                    .BYTE   6,2
010056  000176                            SWREG
```

```
                                                  ;
                                                  ;ROUTINE USED TO ''CYCLE'' THROUGH UP TO 16 DMC11'S
                                                  ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                  ;AND RUNS THE SPECIFIED DMC11'S.   THIS ROUTINE *MUST*
                                                  ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                  ;SETUP NECESSARY.
                                                  ;

010060  005737  001306          CYCLE:  TST     DMACTV          ;ARE ANY DMC11'S TO BE TESTED?
010064  001004                          BNE     1$              ;BR IF OK.
010066  104402  007154                  TYPE    ,NOACT          ;NO DMC11'S SELECTED!!
010072  000000                          HALT                    ;STOP THE SHOW.
010074  000776                          BR      .-2             ;DISQUALIFY CONT. SW.
010076  000241                  1$:     CLC                     ;CLEAR PROC. CARRY BIT.
010100  006137  001316                  ROL     RUN             ;UPDATE POINTER
010104  005537  001316                  ADC     RUN             ;CATCH CARRY FROM RUN
010110  062737  000004  001322          ADD     #4,MILK         ;UPDATE POINTER
010116  062737  000010  001320          ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
010124  022737  001700  001320          CMP     #DM.MAP+200,CREAM
010132  001006                          BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
010134  012737  001500  001320          MOV     #DM.MAP,CREAM   ;RESET ADDRESS POINTER.
010142  012737  001702  001322          MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
010150  033737  001316  001306  2$:     BIT     RUN,DMACTV      ;IS THIS ONE ACTIVE?
010156  001747                          BEQ     1$              ;BR IF NO
010160  013700  001320                  MOV     CREAM,R0        ;GET ADDRESS POINTER
010164  013702  001322                  MOV     MILK,R2         ;GET PASS COUNT POINTER
010170  012037  001404                  MOV     (R0)+,DMCSR     ;LOAD SYSTEM CTRL. REG
010174  011037  001374                  MOV     (R0),DMRVEC     ;LOAD VECTOR
010200  042737  177000  001374          BIC     #177000,DMRVEC  ;CLEAR UNWANTED BITS
010206  012037  001366                  MOV     (R0)+,STAT1     ;LOAD STAT1
010212  012037  001370                  MOV     (R0)+,STAT2     ;LOAD STAT2
010216  012037  001372                  MOV     (R0)+,STAT3     ;LOAD STAT3
010222  012237  001230                  MOV     (R2)+,PASCNT    ;LOAD PASS COUNT
010226  012237  001232                  MOV     (R2)+,ERRCNT    ;LOAD ERROR COUNT
010232  012700  000002                  MOV     #2,R0           ;SAVE CORE THIS WAY.
010236  013737  001404  001406          MOV     DMCSR,DMCSRH
010244  005237  001406                  INC     DMCSRH
010250  013737  001406  001410          MOV     DMCSRH,DMCTL
010256  005237  001410                  INC     DMCTL
010262  013737  001410  001412          MOV     DMCTL,DMP04
010270  060037  001412                  ADD     R0,DMP04
010274  013737  001412  001414          MOV     DMP04,DMP06
010302  060037  001414                  ADD     R0,DMP06

010306  013737  001374  001376          MOV     DMRVEC,DMRLVL   ;PTY LVL
010314  060037  001376                  ADD     R0,DMRLVL       ;
010320  013737  001376  001400          MOV     DMRLVL,DMTVEC   ;TX VEC
010326  060037  001400                  ADD     R0,DMTVEC       ;
010332  013737  001400  001402          MOV     DMTVEC,DMTLVL   ;TX LVL
010340  060037  001402                  ADD     R0,DMTLVL

010344  032737  000002  001236          BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
010352  001450                          BEQ     7$              ;BR IF NO
010354                          4$:
010354  005737  000042                  TST     @#42            ;RUNNING IN AUTO MODE?
```

```
010360  001045                                  BNE     7$              ;BR IF YES
010362  104402  005672                          TYPE    ,MCRLF
010366  104403                                  INSTR                   ;GET TEST NO.
010370  006130                                  MTSTN
010372  104405                                  PARAM
010374  000001                                  1
010376  001000                                  1000
010400  001226                                  TSTNO
010402     000                          .BYTE   0
010403     001                          .BYTE   1
010404  012700  012320                          MOV     #TST1,R0
010410  022710                          5$:     CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
010412  012737                                  MOV     (PC)+,@(PC)+
010414  001020                                  BNE     6$              ;BR IF NOT SAME
010416  023760  001226  000002                  CMP     TSTNO,2(R0)     ;DOES TSTNO MATCH?
010424  001014                                  BNE     6$              ;BR IF NO
010426  022760  001226  000004                  CMP     #TSTNO,4(R0)    ;IS LAST WORD OK?
010434  001010                                  BNE     6$              ;BR IF NO
010436  010037  001214                          MOV     R0,RETURN       ;IT IS A LEGAL TEST SO DO IT
010442  104402  005755                          TYPE    ,MR
010446  042737  000002  001236                  BIC     #SW01,STRTSW
010454  000412                                  BR      8$
010456  005720                          6$:     TST     (R0)+           ;POP R0
010460  020027  033746                          CMP     R0,#TLAST+10    ;AT END YET?
010464  001351                                  BNE     5$              ;BR IF NO
010466  104402  005666                          TYPE    ,MQM            ;YES ILLEGAL TEST NO.
010472  000730                                  BR      4$              ;TRY AGAIN

010474  012737  012320  001214          7$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
010502  013701  001404                  8$:     MOV     DMCSR,R1        ;R1 = BASE DMC11 ADDRESS
010506  000177  170502                          JMP     @RETURN         ;GO START TESTING.


                                                ;ROUTINE USED TO ''AUTO SIZE'' THE DMC11
                                                ;CSR AND VECTOR.
                                                ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
                                                ;       ADDRESS RANGE (160000:164000)
                                                ;       AND THE VECTOR MAY BE ANY WHERE IN THE
                                                ;       FLOATING VECTOR RANGE (300:770)
                                                ;
                                                ;

010512                                  AUTO.SIZE:
010512  000005                                  RESET                   ;INSURE A BUS INIT.
010514  012702  001500                  CSRMAP: MOV     #DM.MAP,R2      ;LOAD MAP POINTER.
010520  005022                          1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
010522  022702  001700                          CMP     #DM.END,R2      ;ALL DONE?
010526  001374                                  BNE     1$              ;BR IF NO
010530  005037  001310                          CLR     DMNUM           ;SET OCTAL NUMBER OF DMC11'S TO 0
010534  012702  001500                          MOV     #DM.MAP,R2      ;R2 POINTS TO DMC MAP
010540  005037  001306                          CLR     DMACTV          ;CLEAR ACTIVE
010544  032737  000001  001236                  BIT     #SW00,STRTSW    ;QUESTIONS?
010552  001002                                  BNE     .+6             ;BR IF YES
010554  000137  011252                          JMP     7$              ;IF NO SKIP QUESTIONS
010560  012737  000001  001256                  MOV     #1,TEMP5        ;START WITH 1
010566  104403                                  INSTR
010570  006450                                  NUM
```

```
010572  104405                          PARAM
010574  000001                          1
010576  000020                          16.
010600  001252                          TEMP3
010602     000                          .BYTE   0
010603     001                          .BYTE   1
010604  013737  001252  001310          MOV     TEMP3,DMNUM     ;DMNUM - HOW MANY
010612  104402  005672          12$:    TYPE    ,MCRLF
010616  104410                          CONVRT                  ;TYPE WHICH DMC IS BEING DONE
010620  012002                          WHICH                   ;TEMP5 IS WHICH DMC
010622  005237  001256                  INC     TEMP5
010626  104403                          INSTR
010630  006510                          CSR
010632  104405                          PARAM
010634  160000                          160000
010636  164000                          164000
010640  001254                          TEMP4
010642     000                          .BYTE   0
010643     001                          .BYTE   1
010644  013722  001254                  MOV     TEMP4,(R2)+     ;STORE CSR IN MAP
010650  104403                          INSTR
010652  006526                          VEC
010654  104405                          PARAM
010656  000000                          0
010660  000776                          776
010662  001254                          TEMP4
010664     000                          .BYTE   0
010665     001                          .BYTE   1
010666  013712  001254                  MOV     TEMP4,(R2)      ;STORE VECTOR IN MAP
010672  104402                  10$:    TYPE
010674  006547                          PRIO                    ;ASK WHAT BR LEVEL
010676  004737  012266                  JSR     PC,INTTY        ;GET RESPONSE
010702  022703  000024                  CMP     #24,R3          ;
010706  101014                          BHI     50$             ;BR IF LESS THAN 4
010710  022703  000027                  CMP     #27,R3          ;
010714  103411                          BLO     50$             ;BR IF GREATER THAN 7
010716  012704  000011                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
010722  006303                          ASL     R3              ;SHIFT R3 LEFT
010724  005304                          DEC     R4              ;DEC SHIFT COUNT
010726  001375                          BNE     .-4             ;BR IF NOT DONE
010730  042703  170777                  BIC     #170777,R3      ;BIC UNWANTED BITS
010734  050312                          BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
010736  000403                          BR      8$              ;CONTINUE
010740  104402                  50$:    TYPE
010742  005666                          MQM                     ;RESPONSE IS OUT OF LIMITS
010744  000752                          BR      10$             ;TRY AGAIN
010746  104402                  8$:     TYPE
010750  006606                          CRAM                    ;DOES DMC HAVE CRAM?
010752  004737  012266                  JSR     PC,INTTY        ;GET REPLY
010756  022703  000131                  CMP     #131,R3
010762  001427                          BEQ     9$              ;YES
010764  022703  000116                  CMP     #116,R3         ;NO
010770  001403                          BEQ     40$             ;NOT A Y OR N
010772  104402                          TYPE
010774  005666                          MQM                     ;TYPE "?"
010776  000763                          BR      8$              ;ASK AGAIN
```

```
011000  104402                    40$:    TYPE
011002  007340                            SPEED                         ;DMC11-AR OR DMC11-AL?
011004  004737  012266                    JSR     PC,INTTY              ;GET RESPONSE
011010  022703  000122                    CMP     #122,R3               ;IS IT R
011014  001414                            BEQ     16$                   ;BR IF REMOTE
011016  022703  000114                    CMP     #114,R3               ;IS IT L
011022  001403                            BEQ     41$                   ;BR IF LOCAL
011024  104402                            TYPE
011026  005666                            MQM
011030  000763                            BR      40$                   ;TRY AGAIN
011032  052762  000002  000004    41$:    BIS     #BIT1,4(R2)           ;SET BIT1 IN STAT3
011040  000402                            BR      16$                   ;CONTINUE
011042  052712  100000            9$:     BIS     #BIT15,(R2)           ;SET BIT 15 IF CRAM
011046  104402                    16$:    TYPE
011050  006704                            MODU                          ;ASK WHICH LINE UNIT
011052  004737  012266                    JSR     PC,INTTY              ;GET REPLY
011056  022703  000021                    CMP     #21,R3                ;"1"
011062  001417                            BEQ     30$
011064  022703  000022                    CMP     #22,R3                ;"2"
011070  001412                            BEQ     31$
011072  022703  000116                    CMP     #116,R3               ;"N"
011076  001403                            BEQ     32$
011100  104402                            TYPE
011102  005666                            MQM                           ;IF NOT A 1,2 OR N TYPE "?"
011104  000760                            BR      16$                   ;TRY AGIAN
011106  052722  010000            32$:    BIS     #BIT12,(R2)+          ;SET BIT 12 IN STAT2 IF NO LU
011112  022222                            CMP     (R2)+,(R2)+           ;POP OVER STAT2 AND STAT3
011114  000447                            BR      33$
011116  052712  020000            31$:    BIS     #BIT13,(R2)           ;SET BIT 13 IN STAT2 IF M8202
011122  104402                    30$:    TYPE
011124  007114                            CONN                          ;ASK IF LOOP-BACK IS ON
011126  004737  012266                    JSR     PC,INTTY              ;GET REPLY
011132  022703  000131                    CMP     #131,R3               ;Y
011136  001406                            BEQ     17$
011140  022703  000116                    CMP     #116,R3               ;N
011144  001406                            BEQ     18$
011146  104402                            TYPE
011150  005666                            MQM                           ;IF NOT Y OR N TYPE "?"
011152  000763                            BR      30$                   ;TRY AGAIN
011154  052722  040000            17$:    BIS     #BIT14,(R2)+          ;TURNAROUND IS CONNECTED
011160  000402                            BR      19$
011162  042722  040000            18$:    BIC     #BIT14,(R2)+          ;NO TURNAROUND
011166                            19$:
011166  104403                            INSTR
011170  007016                            LINE
011172  104405                            PARAM
011174  000000                            0
011176  000377                            377
011200  001254                            TEMP4
011202     000                            .BYTE   0
011203     001                            .BYTE   1
011204  113722  001254                    MOVB    TEMP4,(R2)+           ;STORE SWITCH PAC IN MAP
011210  104403                            INSTR
011212  007054                            BM
011214  104405                            PARAM
011216  000000                            0
```

```
011220  000377                                    377
011222  001254                                    TEMP4
011224     000                                    .BYTE   0
011225     001                                    .BYTE   1
011226  113722  001254                            MOVB    TEMP4,(R2)+          ;STORE SWITCH PAC IN MAP
011232  005722                                    TST     (R2)+                ;POP OVER STAT3
011234  005337  001252                    33$:    DEC     TEMP3                ;DEC DMC COUNT
011240  001402                                    BEQ     34$                  ;BR IF DONE
011242  000137  010612                            JMP     12$                  ;JUMP IF NOT
011246  000137  011702                    34$:    JMP     13$                  ;CONTINUE
011252  012701  160000                    7$:     MOV     #160000,R1           ;SET FOR FIRST ADDRESS TO BE TESTED
011256  012737  011774  000004                    MOV     #6$,@#4              ;SET FOR NON-EXISTANT DEVICE TIME OUT
011264  005011                            2$:     CLR     (R1)                 ;CLEAR SEL0
011266  005711                                    TST     (R1)                 ;IF DMC11 DMCSR S/B 0
011270  001172                                    BNE     3$                   ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DMC11
011272  005061  000006                            CLR     6(R1)                ;CLEAR SEL6
011276  005761  000006                            TST     6(R1)                ;IF DMC11 THEN DMRIC S/B  0.
011302  001165                                    BNE     3$                   ;BR IF NOT DMC11
011304  012711  002000                            MOV     #BIT10,(R1)          ;SET ROMO
011310  005061  000004                            CLR     4(R1)                ;CLEAR SEL4
011314  012761  125252  000006                    MOV     #125252,6(R1)        ;WRITE THIS TO SEL6
011322  052711  020000                            BIS     #BIT13,(R1)          ;WRITE IT!
011326  022761  125252  000004                    CMP     #125252,4(R1)        ;WAS IT WRITTEN?
011334  001004                                    BNE     21$                  ;IF NO IT IS NOT CRAM
011336  052762  100000  000002                    BIS     #BIT15,2(R2)         ;SET BIT15 IF CRAM
011344  000431                                    BR      22$
011346  012711  001000                    21$:    MOV     #BIT9,(R1)           ;SET ROMI
011352  012761  100430  000006                    MOV     #100430,6(R1)        ;PUT INSTRUCTION IN SEL6
011360  012711  001400                            MOV     #BIT9!BIT8,(R1)      ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
011364  012711  002000                            MOV     #BIT10,(R1)          ;SET ROMO
011370  022761  016472  000006                    CMP     #016472,6(R1)        ;IS IT LOCAL CROM?
011376  001411                                    BEQ     23$                  ;BR IF YES
011400  022761  016461  000006                    CMP     #016461,6(R1)        ;IS IT REMOTE CROM?
011406  001410                                    BEQ     22$                  ;BR IF YES
011410  022761  177777  000006                    CMP     #-1,6(R1)            ;NO CROM?
011416  001404                                    BEQ     22$                  ;BR IF YES
011420  000516                                    BR      3$                   ;NOT A DMC
011422  052762  000002  000006            23$:    BIS     #BIT1,6(R2)          ;SET BIT 1 IN STAT3
                                          ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DMC11 CSR ADDRESS.
011430  010122                            22$:    MOV     R1,(R2)+             ;STORE CSR IN CORE TABLE.
011432  012711  001000                    15$:    MOV     #BIT9,(R1)           ;CLEAR LINE UNIT LOOP
011436  005061  000004                            CLR     4(R1)                ;CLEAR PORT4
011442  012761  122113  000006                    MOV     #122113,6(R1)        ;LOAD INSTRUCTION (CLR DTR)
011450  052711  000400                            BIS     #BIT8,(R1)           ;CLOCK INSTRUCTION
011454  012761  021264  000006                    MOV     #021264,6(R1)        ;LOAD INSTRUCTION
011462  052711  000400                            BIS     #BIT8,(R1)           ;CLOCK INSTRUCTION
011466  122761  000377  000004                    CMPB    #377,4(R1)           ;IS IT ALL ONES?
011474  001003                                    BNE     .+10                 ;BR IF NO
011476  052712  010000                            BIS     #BIT12,(R2)          ;IF YES, NO LINE UNIT, SET STATUS BIT
011502  000436                                    BR      20$
011504  032761  000002  000004                    BIT     #BIT1,4(R1)          ;IS SWITCH A ONE?
011512  001403                                    BEQ     .+10                 ;BR IF M8201
011514  052712  060000                            BIS     #BIT13!BIT14,(R2)    ;M8202 ASSUME CONNECTOR
011520  000427                                    BR      20$                  ;CONNECTOR ON)
011522  032761  000010  000004                    BIT     #BIT3,4(R1)          ;IS MRDY SET
011530  001023                                    BNE     20$                  ;BR IF M8201 NO CONNECTOR ON LINE
```

```
011532 012761 000100 000004          MOV     #BIT6,4(R1)        ;LOAD PORT4
011540 012761 122113 000006          MOV     #122113,6(R1)      ;LOAD INSTRUCTION
011546 052711 000400                 BIS     #BIT8,(R1)         ;CLOCK INSTRUCTION(SET DTR)
011552 012761 021264 000006          MOV     #021264,6(R1)      ;LOAD INSTRUCTION
011560 052711 000400                 BIS     #BIT8,(R1)         ;CLOCK INSTRUCTION(READ MODEM REG)
011564 032761 000010 000004          BIT     #BIT3,4(R1)        ;IS MRDY SET NOW?
011572 001402                        BEQ     20$                ;BR IF NO CONNECTOR
011574 052712 040000                 BIS     #BIT14,(R2)        ;SET STATUS BIT FOR CONNECTOR
011600 005722                20$:    TST     (R2)+              ;POP POINTER
011602 012761 021324 000006          MOV     #021324,6(R1)      ;PUT INSTRUCTION IN PORT6
011610 012711 001400                 MOV     #BIT9!BIT8,(R1)    ;PORT4_LU 15
011614 156122 000004                 BISB    4(R1),(R2)+        ;STORE DDCMP LINE # IN TABLE
011620 012761 021344 000006          MOV     #021344,6(R1)      ;PORT6_INSTRUCTION
011626 012711 001400                 MOV     #BIT8,BIT9,(R1)    ;CLOCK INSTR.
011632 156122 000004                 BISB    4(R1),(R2)+        ;STORE BM873 ADD IN TABLE
011636 005722                        TST     (R2)+              ;POP OVER STAT3
011640 005011                        CLR     (R1)               ;CLEAR ROMI
011642 005237 001310                 INC     DMNUM              ;UPDATE DEVICE COUNTER
011646 022737 000020 001310          CMP     #20,DMNUM          ;ARE MAX. NO. OF DEV FOUND?
011654 001412                        BEQ     13$                ;YES DON'T LOOK FOR ANY MORE.
011656 005011                3$:     CLR     (R1)               ;CLEAR BIT 10
011660 005061 000006                 CLR     6(R1)              ;CLEAR SEL 6
011664 062701 000010         14$:    ADD     #10,R1             ;UPDATE CSR POINTER ADDRESS
011670 022701 164000                 CMP     #164000,R1
011674 001402                        BEQ     13$                ;BR IF DONE
011676 000137 011264                 JMP     2$                 ;JUMP IF NOT
011702 005037 001306         13$:    CLR     DMACTV
011706 005737 001310                 TST     DMNUM              ;WERE ANY DMC11'S FOUND AT ALL?
011712 001423                        BEQ     5$                 ;ERROR AUTO SIZER FOUND NO DMC11'S IN THIS SYS.
011714 013701 001310                 MOV     DMNUM,R1
011720 010137 001314                 MOV     R1,SAVNUM          ;SAVE NUMBER OF DEVICES
011724 000241                4$:     CLC
011726 006137 001306                 ROL     DMACTV             ;GENERATE ACTIVE REGISTER OF DEVICES.
011732 005237 001306                 INC     DMACTV             ;SET THE BIT
011736 005301                        DEC     R1
011740 001371                        BNE     4$                 ;BR IF MORE TO GENERATE
011742 012737 000006 000004          MOV     #6,@#4             ;RESTORE TRAP VECTOR
011750 013737 001306 001312          MOV     DMACTV,SAVACT      ;SAVE ACTIVE REGISTER
011756 000137 012010                 JMP     VECMAP             ;GO FIND THE VECTOR NOW.
011762 104402 005760         5$:     TYPE    ,MERR2             ;NOTIFY OPR THAT NO DMC11'S FOUND.
011766 005000                        CLR     R0                 ;MAKE DATA LIGHTS ZERO
011770 000000                        HALT                       ;STOP THE SHOW
011772 000776                        BR      .-2                ;DISABLE CONT. SW.
011774 012716 011664         6$:     MOV     #14$,(SP)          ;ENTERED BY NON-EXISTANT TIME-OUT.
012000 000002                        RTI                        ;RETURN TO MAINSTREAM

012002 000001                WHICH:  1
012004    002    002                 .BYTE   2,2
012006 001256                        TEMPS

012010 032737 000001 001236  VECMAP: BIT     #SW00,STRTSW
012016 001114                        BNE     5$
012020 012737 000340 000022          MOV     #340,@#22          ;SET IOT TRAP PRIO TO 7
012026 012737 012202 000020          MOV     #4$,@#20           ;SET IOT TRAP VECTOR
012034 012702 001500                 MOV     #DM.MAP,R2         ;SET SOFTWARE POINTER
012040 012700 000300                 MOV     #300,R0            ;FLOATING VECTORS START HERE.
```

```
012044  012701  000302                     MOV     #302,R1            ;PC OF IOT INSTR.
012050  010120                     1$:     MOV     R1,(R0)+           ;START FILLING VECTOR AREA
012052  012721  000004                     MOV     #4,(R1)+           ;WITH .+2; IOT
012056  022021                             CMP     (R0)+,(R1)+        ;ADD 2 TO R0 +R1
012060  020127  001000                     CMP     R1,#1000
012064  101771                             BLOS    1$                 ;BR IF MORE TO FILL
012066  013737  001306  001246             MOV     DMACTV,TEMP1       ;STORE TEMPORALLY
012074  006037  001246             2$:     ROR     TEMP1              ;BRING OUT A BIT
012100  103063                             BCC     5$                 ;BR IF ALL DONE
012102  012704  000012                     MOV     #12,R4             ;R4 IS INDEX REGISTER
012106  016437  012252  177776             MOV     BRLVL(R4),PS       ;SET PS TO 7
012114  011201                             MOV     (R2),R1
012116  012761  000200  000004             MOV     #200,4(R1)
012124  012711  001000                     MOV     #BIT9,(R1)         ;SET ROMI
012130  012761  121111  000006             MOV     #121111,6(R1)      ;PUT INSTRUCTION IN PORT6
012136  012711  001400                     MOV     #BIT9.BIT8,(R1)    ;FORCE AN INTERRUPT
012142  105200                     7$:     INCB    R0                 ;STALL
012144  001376                             BNE     .-2                ;FOR TIME TO INTERUPT
012146  162704  000002                     SUB     #2,R4              ;GET NEXT LOWEST PS LEVEL
012152  001404                             BEQ     6$                 ;BR IF R4 = 0
012154  016437  012252  177776             MOV     BRLVL(R4),PS       ;MOVE NEXT LOWER LEVEL IN PS
012162  000767                             BR      7$                 ;BR TO DELAY
012164  052762  005300  000002     6$:     BIS     #5300,2(R2)        ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX DMC11 LATER
012172  005011                     3$:     CLR     (R1)               ;CLEAR ROMI
012174  062702  000010                     ADD     #10,R2             ;POP SOFTWARE POINTER
012200  000735                             BR      2$                 ;KEEP GOING
012202  051662  000002             4$:     BIS     (SP),2(R2)         ;GET VECTOR ADDRESS
012206  042762  000007  000002             BIC     #7,2(R2)           ;CLEAR JUNK
012214  016405  012254                     MOV     BRLVL+2(R4),R5     ;GET BR LEVEL OF DMC11
012220  006305                             ASL     R5                 ;SHIFT LEVEL 4 PLACES
012222  006305                             ASL     R5                 ;TO THE LEFT FOR THE
012224  006305                             ASL     R5                 ;STATUS TABLE
012226  006305                             ASL     R5
012230  042705  170777                     BIC     #170777,R5         ;CLEAR UNWANTED BITS
012234  050562  000002                     BIS     R5,2(R2)           ;PUT BR LEVEL IN STATUS TABLE
012240  022626                             CMP     (SP)+,(SP)+        ;POP IOT JUNK OFF STACK
012242  012716  012172                     MOV     #3$,(SP)           ;SET FOR RETURN
012246  000002                             RTI
012250  000207                     5$:     RTS     PC                 ;ALL DONE WITH ''AUTO SIZING''

012252  000000                     BRLVL:  0       ;LEVEL 0
012254  000000                             0       ;LEVEL 0
012256  000200                             200     ;LEVEL 4
012260  000240                             240     ;LEVEL 5
012262  000300                             300     ;LEVEL 6
012264  000340                             340     ;LEVEL 7


012266  105777  166712             INTTY:  TSTB    @TKCSR             ;WAIT FOR DONE
012272  100375                             BPL     .-4
012274  017703  166706                     MOV     @TKDBR,R3          ;PUT CHAR IN R3
012300  105777  166704                     TSTB    @TPCSR             ;WAIT UNTIL PRINTER IS READY
012304  100375                             BPL     .-4
012306  010377  166700                     MOV     R3,@TPDBR          ;ECHO CHAR
012312  042703  000240                     BIC     #BIT7.BIT5,R3      ;MASK OFF LOWER CASE
012316  000207                             RTS     PC                 ;RETURN
```

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

                                                ;  TEST 1
                                                ;--------------
012320  012737  000001  001226  TST1:   MOV     #1,TSTNO
012326  012737  012426  001216          MOV     #TST2,NEXT
012334  012737  012366  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
012342  013701  001404                  MOV     DMCSR,R1        ;R1 CONTAINS BASE DMC11 ADDRESS
012346  012700  000004                  MOV     #4,R0           ;4 REGISTERS TO BE TESTED
012352  012737  012420  000004          MOV     #2$,4           ;SET UP TIMEOUT TRAP
012360  012737  000340  000006          MOV     #340,6          ;LEVEL 7
012366  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTER
012370  000240                          NOP
012372  104401                          SCOP1                   ;SW09=1?
012374  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
012400  005300                          DEC     R0              ;DEC REGISTER COUNT
012402  001371                          BNE     1$              ;BR IF NOT LAST REGISTER
012404  012737  000006  000004          MOV     #6,4            ;RESTORE LOC 4
012412  005037  000006                  CLR     6               ;RESTORE LOC 6
012416  104400                          SCOPE                   ;SCOPE THIS TEST
012420  011602                  2$:     MOV     (SP),R2         ;GET PC OF TRAP
012422  104001                          HLT     1               ;TIME-OUT ERROR
012424  000002                          RTI


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

                                                ;  TEST 2
                                                ;--------------
012426  012737  000002  001226  TST2:   MOV     #2,TSTNO
012434  012737  012456  001216          MOV     #TST3,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
012442  005011                          CLR     (R1)            ;CLEAR DMCSR
012444  005005                          CLR     R5              ;CLEAR "EXPECTED"
012446  011104                          MOV     (R1),R4         ;PUT DMCSR IN "FOUND"
012450  001401                          BEQ     1$              ;BR IF CLEARED
012452  104002                          HLT     2               ;ERROR DMCSR NOT CLEARED
012454  104400                  1$:     SCOPE                   ;SCOPE THIS TEST

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

                                                ;  TEST 3
```

```
                                                ;--------------
012456  012737  000003  001226  TST3:   MOV     #3,TSTNO
012464  012737  012606  001216          MOV     #TST4,NEXT
012472  012737  012506  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
012500  104412                          MSTCLR                  ;MASTER CLEAR DMC11
012502  012700  000001                  MOV     #1,R0           ;START PATTERN AT 1
012506  005011                  1$:     CLR     (R1)            ;CLEAR REGISTER
012510  010005                          MOV     R0,R5           ;PUT DATA IN 'EXPECTED''
012512  010011                          MOV     R0,(R1)         ;WRITE DMC REGISTER WITH PATTERN
012514  011104                          MOV     (R1),R4         ;READ DMC REGISTER INTO 'FOUND''
012516  020504                          CMP     R5,R4           ;IS DATA CORRECT
012520  001401                          BEQ     2$              ;BR IF YES
012522  104002                          HLT     2               ;DATA ERROR
012524  104401                  2$:     SCOP1                   ;SW09=1?
012526  005721                          TST     (R1)+           ;NEXT REGISTER
012530  005200                          INC     R0              ;INCREMENT DATA PATTERN
012532  022700  000005                  CMP     #5,R0           ;LAST REGISTER?
012536  001363                          BNE     1$              ;BR IF NO
012540  013701  001404                  MOV     DMCSR,R1        ;BASE DMC11  ADDRESS TO R1
012544  012700  000001                  MOV     #1,R0           ;RESTART PATTERN AT 1
012550  012737  012556  001220          MOV     #3$,LOCK        ;NEW SCOP1
012556  010005                  3$:     MOV     R0,R5           ;PUT DATA IN 'EXPECTED''
012560  011104                          MOV     (R1),R4         ;READ DMC REGISTER INTO 'FOUND''
012562  020504                          CMP     R5,R4           ;IS DATA CORRECT
012564  001401                          BEQ     4$              ;BR IF YES
012566  104002                          HLT     2               ;DUAL ADDRESSING ERROR
012570  104401                  4$:     SCOP1                   ;SW09-1?
012572  005721                          TST     (R1)+           ;NEXT REGISTER
012574  005200                          INC     R0              ;INCREMENT PATTERN
012576  022700  000005                  CMP     #5,R0           ;LAST REGISTER?
012602  001365                          BNE     3$              ;BR IF NO
012604  104400                          SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 4
                                        ;--------------
012606  012737  000004  001226  TST4:   MOV     #4,TSTNO
012614  012737  012704  001216          MOV     #TST5,NEXT
012622  012737  012632  001220          MOV     #1$,LOCK
012630  104412                          MSTCLR                  ;MASTER CLEAR DMC11
012632  013701  001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
012636  012705  000001                  MOV     #BIT0,R5        ;PUT DATA IN 'EXPECTED''
012642  010511                          MOV     R5,(R1)         ;WRITE BIT 0
012644  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012646  020504                          CMP     R5,R4           ;IS DATA CORRECT
012650  001401                          BEQ     2$              ;BR IF YES
012652  104002                          HLT     2               ;DATA ERROR
012654  104401                  2$:     SCOP1                   ;SW09 UP?
012656  012737  012664  001220          MOV     #3$,LOCK        ;NEW SCOP1
```

```
012664  042711  000001            3$:     BIC     #BIT0,(R1)      ;CLEAR BIT 0
012670  005005                            CLR     R5              ;CLEAR 'EXPECTED'
012672  011104                            MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012674  001402                            BEQ     4$              ;BR IF ZERO
012676  104002                            HLT     2               ;DATA ERROR BIT0 NOT CLEARED
012700  104401                            SCOP1                   ;SW09 UP?
012702  104400                    4$:     SCOPE                   ;SCOPE THIS TEST


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

                                          ;  TEST 5
                                          ;--------------
012704  012737  000005  001226    TST5:   MOV     #5,TSTNO
012712  012737  013002  001216            MOV     #TST6,NEXT
012720  012737  012730  001220            MOV     #1$,LOCK
012726  104412                            MSTCLR                  ;MASTER CLEAR DMC11
012730  013701  001404            1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
012734  012705  000002                    MOV     #BIT1,R5        ;PUT DATA IN 'EXPECTED'
012740  010511                            MOV     R5,(R1)         ;WRITE BIT 1
012742  011104                            MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012744  020504                            CMP     R5,R4           ;IS DATA CORRECT
012746  001401                            BEQ     2$              ;BR IF YES
012750  104002                            HLT     2               ;DATA ERROR
012752  104401                    2$:     SCOP1                   ;SW09 UP?
012754  012737  012762  001220            MOV     #3$,LOCK        ;NEW SCOP1
012762  042711  000002            3$:     BIC     #BIT1,(R1)      ;CLEAR BIT 1
012766  005005                            CLR     R5              ;CLEAR 'EXPECTED'
012770  011104                            MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
012772  001402                            BEQ     4$              ;BR IF ZERO
012774  104002                            HLT     2               ;DATA ERROR BIT1 NOT CLEARED
012776  104401                            SCOP1                   ;SW09 UP?
013000  104400                    4$:     SCOPE                   ;SCOPE THIS TEST


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

                                          ;  TEST 6
                                          ;--------------
013002  012737  000006  001226    TST6:   MOV     #6,TSTNO
013010  012737  013100  001216            MOV     #TST7,NEXT
013016  012737  013026  001220            MOV     #1$,LOCK
013024  104412                            MSTCLR                  ;MASTER CLEAR DMC11
013026  013701  001404            1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
013032  012705  000004                    MOV     #BIT2,R5        ;PUT DATA IN 'EXPECTED'
013036  010511                            MOV     R5,(R1)         ;WRITE BIT 2
013040  011104                            MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013042  020504                            CMP     R5,R4           ;IS DATA CORRECT
```

```
013044  001401                          BEQ     2$              ;BR IF YES
013046  104002                          HLT     2               ;DATA ERROR
013050  104401                  2$:     SCOP1                   ;SW09 UP?
013052  012737  013060  001220          MOV     #3$,LOCK        ;NEW SCOP1
013060  042711  000004          3$:     BIC     #BIT2,(R1)      ;CLEAR BIT 2
013064  005005                          CLR     R5              ;CLEAR 'EXPECTED''
013066  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013070  001402                          BEQ     4$              ;BR IF ZERO
013072  104002                          HLT     2               ;DATA ERROR BIT2 NOT CLEARED
013074  104401                          SCOP1                   ;SW09 UP?
013076  104400                  4$:     SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 7
                                        ;--------------
013100  012737  000007  001226  TST7:   MOV     #7,TSTNO
013106  012737  013176  001216          MOV     #TST10,NEXT
013114  012737  013124  001220          MOV     #1$,LOCK
013122  104412                          MSTCLR                  ;MASTER CLEAR DMC11
013124  013701  001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
013130  012705  000040                  MOV     #BIT5,R5        ;PUT DATA IN 'EXPECTED''
013134  010511                          MOV     R5,(R1)         ;WRITE BIT 5
013136  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013140  020504                          CMP     R5,R4           ;IS DATA CORRECT
013142  001401                          BEQ     2$              ;BR IF YES
013144  104002                          HLT     2               ;DATA ERROR
013146  104401                  2$:     SCOP1                   ;SW09 UP?
013150  012737  013156  001220          MOV     #3$,LOCK        ;NEW SCOP1
013156  042711  000040          3$:     BIC     #BIT5,(R1)      ;CLEAR BIT 5
013162  005005                          CLR     R5              ;CLEAR 'EXPECTED''
013164  011104                          MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013166  001402                          BEQ     4$              ;BR IF ZERO
013170  104002                          HLT     2               ;DATA ERROR BIT5 NOT CLEARED
013172  104401                          SCOP1                   ;SW09 UP?
013174  104400                  4$:     SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 10
                                        ;--------------
013176  012737  000010  001226  TST10:  MOV     #10,TSTNO
013204  012737  013274  001216          MOV     #TST11,NEXT
013212  012737  013222  001220          MOV     #1$,LOCK
013220  104412                          MSTCLR                  ;MASTER CLEAR DMC11
013222  013701  001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
```

```
013226  012705  000100                          MOV     #BIT6,R5            ;PUT DATA IN 'EXPECTED''
013232  010511                                  MOV     R5,(R1)             ;WRITE BIT 6
013234  011104                                  MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013236  020504                                  CMP     R5,R4               ;IS DATA CORRECT
013240  001401                                  BEQ     2$                  ;BR IF YES
013242  104002                                  HLT     2                   ;DATA ERROR
013244  104401                          2$:     SCOP1                       ;SW09 UP?
013246  012737  013254  001220                  MOV     #3$,LOCK            ;NEW SCOP1
013254  042711  000100                  3$:     BIC     #BIT6,(R1)          ;CLEAR BIT 6
013260  005005                                  CLR     R5                  ;CLEAR 'EXPECTED''
013262  011104                                  MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013264  001402                                  BEQ     4$                  ;BR IF ZERO
013266  104002                                  HLT     2                   ;DATA ERROR BIT6 NOT CLEARED
013270  104401                                  SCOP1                       ;SW09 UP?
013272  104400                          4$:     SCOPE                       ;SCOPE THIS TEST


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

                                                ;  TEST 11
                                                ;---------------
013274  012737  000011  001226          TST11:  MOV     #11,TSTNO
013302  012737  013372  001216                  MOV     #TST12,NEXT
013310  012737  013320  001220                  MOV     #1$,LOCK
013316  104412                                  MSTCLR                      ;MASTER CLEAR DMC11
013320  013701  001404                  1$:     MOV     DMCSR,R1            ;PUT REGISTER ADDRESS IN R1
013324  012705  000200                          MOV     #BIT7,R5            ;PUT DATA IN 'EXPECTED''
013330  010511                                  MOV     R5,(R1)             ;WRITE BIT 7
013332  011104                                  MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013334  020504                                  CMP     R5,R4               ;IS DATA CORRECT
013336  001401                                  BEQ     2$                  ;BR IF YES
013340  104002                                  HLT     2                   ;DATA ERROR
013342  104401                          2$:     SCOP1                       ;SW09 UP?
013344  012737  013352  001220                  MOV     #3$,LOCK            ;NEW SCOP1
013352  042711  000200                  3$:     BIC     #BIT7,(R1)          ;CLEAR BIT 7
013356  005005                                  CLR     R5                  ;CLEAR 'EXPECTED''
013360  011104                                  MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013362  001402                                  BEQ     4$                  ;BR IF ZERO
013364  104002                                  HLT     2                   ;DATA ERROR BIT7 NOT CLEARED
013366  104401                                  SCOP1                       ;SW09 UP?
013370  104400                          4$:     SCOPE                       ;SCOPE THIS TEST


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

                                                ;  TEST 12
                                                ;---------------
013372  012737  000012  001226          TST12:  MOV     #12,TSTNO
```

```
013400 012737 013470 001216           MOV     #TST13,NEXT
013406 012737 013416 001220           MOV     #1$,LOCK
013414 104412                         MSTCLR                  ;MASTER CLEAR DMC11
013416 013701 001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
013422 012705 001000                  MOV     #BIT9,R5        ;PUT DATA IN 'EXPECTED''
013426 010511                         MOV     R5,(R1)         ;WRITE BIT 9
013430 011104                         MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013432 020504                         CMP     R5,R4           ;IS DATA CORRECT
013434 001401                         BEQ     2$              ;BR IF YES
013436 104002                         HLT     2               ;DATA ERROR
013440 104401                 2$:     SCOP1                   ;SW09 UP?
013442 012737 013450 001220           MOV     #3$,LOCK        ;NEW SCOP1
013450 042711 001000          3$:     BIC     #BIT9,(R1)      ;CLEAR BIT 9
013454 005005                         CLR     R5              ;CLEAR 'EXPECTED''
013456 011104                         MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013460 001402                         BEQ     4$              ;BR IF ZERO
013462 104002                         HLT     2               ;DATA ERROR BIT9 NOT CLEARED
013464 104401                         SCOP1                   ;SW09 UP?
013466 104400                 4$:     SCOPE                   ;SCOPE THIS TEST


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

                                      ;  TEST 13
                                      ;--------------
013470 012737 000013 001226   TST13:  MOV     #13,TSTNO
013476 012737 013566 001216           MOV     #TST14,NEXT
013504 012737 013514 001220           MOV     #1$,LOCK
013512 104412                         MSTCLR                  ;MASTER CLEAR DMC11
013514 013701 001404          1$:     MOV     DMCSR,R1        ;PUT REGISTER ADDRESS IN R1
013520 012705 004000                  MOV     #BIT11,R5       ;PUT DATA IN 'EXPECTED''
013524 010511                         MOV     R5,(R1)         ;WRITE BIT 11
013526 011104                         MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013530 020504                         CMP     R5,R4           ;IS DATA CORRECT
013532 001401                         BEQ     2$              ;BR IF YES
013534 104002                         HLT     2               ;DATA ERROR
013536 104401                 2$:     SCOP1                   ;SW09 UP?
013540 012737 013546 001220           MOV     #3$,LOCK        ;NEW SCOP1
013546 042711 004000          3$:     BIC     #BIT11,(R1)     ;CLEAR BIT 11
013552 005005                         CLR     R5              ;CLEAR 'EXPECTED''
013554 011104                         MOV     (R1),R4         ;READ CONTROL STATUS REGISTER
013556 001402                         BEQ     4$              ;BR IF ZERO
013560 104002                         HLT     2               ;DATA ERROR BIT11 NOT CLEARED
013562 104401                         SCOP1                   ;SW09 UP?
013564 104400                 4$:     SCOPE                   ;SCOPE THIS TEST


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

```


                                            ;  TEST 14
                                            ;--------------
013566  012737  000014  001226  TST14:      MOV     #14,TSTNO
013574  012737  013664  001216              MOV     #TST15,NEXT
013602  012737  013612  001220              MOV     #1$,LOCK
013610  104412                              MSTCLR                      ;MASTER CLEAR DMC11
013612  013701  001404          1$:         MOV     DMCSR,R1            ;PUT REGISTER ADDRESS IN R1
013616  012705  010000                      MOV     #BIT12,R5           ;PUT DATA IN "EXPECTED"
013622  010511                              MOV     R5,(R1)             ;WRITE BIT 12
013624  011104                              MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013626  020504                              CMP     R5,R4               ;IS DATA CORRECT
013630  001401                              BEQ     2$                  ;BR IF YES
013632  104002                              HLT     2                   ;DATA ERROR
013634  104401                  2$:         SCOP1                       ;SW09 UP?
013636  012737  013644  001220              MOV     #3$,LOCK            ;NEW SCOP1
013644  042711  010000          3$:         BIC     #BIT12,(R1)         ;CLEAR BIT 12
013650  005005                              CLR     R5                  ;CLEAR "EXPECTED"
013652  011104                              MOV     (R1),R4             ;READ CONTROL STATUS REGISTER
013654  001402                              BEQ     4$                  ;BR IF ZERO
013656  104002                              HLT     2                   ;DATA ERROR BIT12 NOT CLEARED
013660  104401                              SCOP1                       ;SW09 UP?
013662  104400                  4$:         SCOPE                       ;SCOPE THIS TEST


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

                                            ;  TEST 15
                                            ;--------------
013664  012737  000015  001226  TST15:      MOV     #15,TSTNO
013672  012737  013762  001216              MOV     #TST16,NEXT
013700  012737  013710  001220              MOV     #1$,LOCK
013706  104412                              MSTCLR                      ;MASTER CLEAR DMC11
013710  013701  001410          1$:         MOV     DMCTL,R1            ;PUT REGISTER ADDRESS IN R1
013714  012705  000001                      MOV     #BIT0,R5            ;PUT DATA IN "EXPECTED"
013720  010511                              MOV     R5,(R1)             ;WRITE BIT 0
013722  011104                              MOV     (R1),R4             ;READ CONTROL OUT REGISTER
013724  020504                              CMP     R5,R4               ;IS DATA CORRECT
013726  001401                              BEQ     2$                  ;BR IF YES
013730  104002                              HLT     2                   ;DATA ERROR
013732  104401                  2$:         SCOP1                       ;SW09 UP?
013734  012737  013742  001220              MOV     #3$,LOCK            ;NEW SCOP1
013742  042711  000001          3$:         BIC     #BIT0,(R1)          ;CLEAR BIT 0
013746  005005                              CLR     R5                  ;CLEAR "EXPECTED"
013750  011104                              MOV     (R1),R4             ;READ CONTROL OUT REGISTER
013752  001402                              BEQ     4$                  ;BR IF ZERO
013754  104002                              HLT     2                   ;DATA ERROR BIT0 NOT CLEARED
013756  104401                              SCOP1                       ;SW09 UP?
013760  104400                  4$:         SCOPE                       ;SCOPE THIS TEST


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

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

                                                ;  TEST 16
                                                ;--------------
013762  012737  000016  001226  TST16:  MOV     #16,TSTNO
013770  012737  014060  001216          MOV     #TST17,NEXT
013776  012737  014006  001220          MOV     #1$,LOCK
014004  104412                          MSTCLR                  ;MASTER CLEAR DMC11
014006  013701  001410          1$:     MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014012  012705  000002                  MOV     #BIT1,R5        ;PUT DATA IN 'EXPECTED''
014016  010511                          MOV     R5,(R1)         ;WRITE BIT 1
014020  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014022  020504                          CMP     R5,R4           ;IS DATA CORRECT
014024  001401                          BEQ     2$              ;BR IF YES
014026  104002                          HLT     2               ;DATA ERROR
014030  104401                  2$:     SCOP1                   ;SW09 UP?
014032  012737  014040  001220          MOV     #3$,LOCK        ;NEW SCOP1
014040  042711  000002          3$:     BIC     #BIT1,(R1)      ;CLEAR BIT 1
014044  005005                          CLR     R5              ;CLEAR 'EXPECTED''
014046  011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014050  001402                          BEQ     4$              ;BR IF ZERO
014052  104002                          HLT     2               ;DATA ERROR BIT1 NOT CLEARED
014054  104401                          SCOP1                   ;SW09 UP?
014056  104400                  4$:     SCOPE                   ;SCOPE THIS TEST


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

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

```
014154 104400                      4$:     SCOPE                     ;SCOPE THIS TEST


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

                                           ;  TEST 20
                                           ;--------------
014156 012737 000020 001226        TST20:  MOV     #20,TSTNO
014164 012737 014254 001216                MOV     #TST21,NEXT
014172 012737 014202 001220                MOV     #1$,LOCK
014200 104412                              MSTCLR                    ;MASTER CLEAR DMC11
014202 013701 001410               1$:     MOV     DMCTL,R1          ;PUT REGISTER ADDRESS IN R1
014206 012705 000100                       MOV     #BIT6,R5          ;PUT DATA IN "EXPECTED"
014212 010511                              MOV     R5,(R1)           ;WRITE BIT 6
014214 011104                              MOV     (R1),R4           ;READ CONTROL OUT REGISTER
014216 020504                              CMP     R5,R4             ;IS DATA CORRECT
014220 001401                              BEQ     2$                ;BR IF YES
014222 104002                              HLT     2                 ;DATA ERROR
014224 104401                      2$:     SCOP1                     ;SW09 UP?
014226 012737 014234 001220                MOV     #3$,LOCK          ;NEW SCOP1
014234 042711 000100               3$:     BIC     #BIT6,(R1)        ;CLEAR BIT 6
014240 005005                              CLR     R5                ;CLEAR "EXPECTED"
014242 011104                              MOV     (R1),R4           ;READ CONTROL OUT REGISTER
014244 001402                              BEQ     4$                ;BR IF ZERO
014246 104002                              HLT     2                 ;DATA ERROR BIT6 NOT CLEARED
014250 104401                              SCOP1                     ;SW09 UP?
014252 104400                      4$:     SCOPE                     ;SCOPE THIS TEST


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

                                           ;  TEST 21
                                           ;--------------
014254 012737 000021 001226        TST21:  MOV     #21,TSTNO
014262 012737 014352 001216                MOV     #TST22,NEXT
014270 012737 014300 001220                MOV     #1$,LOCK
014276 104412                              MSTCLR                    ;MASTER CLEAR DMC11
014300 013701 001410               1$:     MOV     DMCTL,R1          ;PUT REGISTER ADDRESS IN R1
014304 012705 000200                       MOV     #BIT7,R5          ;PUT DATA IN "EXPECTED"
014310 010511                              MOV     R5,(R1)           ;WRITE BIT 7
014312 011104                              MOV     (R1),R4           ;READ CONTROL OUT REGISTER
014314 020504                              CMP     R5,R4             ;IS DATA CORRECT
014316 001401                              BEQ     2$                ;BR IF YES
014320 104002                              HLT     2                 ;DATA ERROR
014322 104401                      2$:     SCOP1                     ;SW09 UP?
014324 012737 014332 001220                MOV     #3$,LOCK          ;NEW SCOP1
014332 042711 000200               3$:     BIC     #BIT7,(R1)        ;CLEAR BIT 7
014336 005005                              CLR     R5                ;CLEAR "EXPECTED"
```

```
014340 011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014342 001402                          BEQ     4$              ;BR IF ZERO
014344 104002                          HLT     2               ;DATA ERROR BIT7 NOT CLEARED
014346 104401                          SCOP1                   ;SW09 UP?
014350 104400                   4$:    SCOPE                   ;SCOPE THIS TEST


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

                                       ;  TEST 22
                                       ;--------------
014352 012737 000022 001226     TST22: MOV     #22,TSTNO
014360 012737 014450 001216            MOV     #TST23,NEXT
014366 012737 014376 001220            MOV     #1$,LOCK
014374 104412                          MSTCLR                  ;MASTER CLEAR DMC11
014376 013701 001410            1$:    MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014402 012705 010000                   MOV     #BIT12,R5       ;PUT DATA IN ''EXPECTED''
014406 010511                          MOV     R5,(R1)         ;WRITE BIT 12
014410 011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014412 020504                          CMP     R5,R4           ;IS DATA CORRECT
014414 001401                          BEQ     2$              ;BR IF YES
014416 104002                          HLT     2               ;DATA ERROR
014420 104401                   2$:    SCOP1                   ;SW09 UP?
014422 012737 014430 001220            MOV     #3$,LOCK        ;NEW SCOP1
014430 042711 010000            3$:    BIC     #BIT12,(R1)     ;CLEAR BIT 12
014434 005005                          CLR     R5              ;CLEAR ''EXPECTED''
014436 011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014440 001402                          BEQ     4$              ;BR IF ZERO
014442 104002                          HLT     2               ;DATA ERROR BIT12 NOT CLEARED
014444 104401                          SCOP1                   ;SW09 UP?
014446 104400                   4$:    SCOPE                   ;SCOPE THIS TEST


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

                                       ;  TEST 23
                                       ;--------------
014450 012737 000023 001226     TST23: MOV     #23,TSTNO
014456 012737 014546 001216            MOV     #TST24,NEXT
014464 012737 014474 001220            MOV     #1$,LOCK
014472 104412                          MSTCLR                  ;MASTER CLEAR DMC11
014474 013701 001410            1$:    MOV     DMCTL,R1        ;PUT REGISTER ADDRESS IN R1
014500 012705 020000                   MOV     #BIT13,R5       ;PUT DATA IN ''EXPECTED''
014504 010511                          MOV     R5,(R1)         ;WRITE BIT 13
014506 011104                          MOV     (R1),R4         ;READ CONTROL OUT REGISTER
014510 020504                          CMP     R5,R4           ;IS DATA CORRECT
014512 001401                          BEQ     2$              ;BR IF YES
014514 104002                          HLT     2               ;DATA ERROR
```

```
014516  104401                          2$:     SCOP1                           ;SW09 UP?
014520  012737  014526  001220                  MOV     #3$,LOCK                ;NEW SCOP1
014526  042711  020000                  3$:     BIC     #BIT13,(R1)             ;CLEAR BIT 13
014532  005005                                  CLR     R5                      ;CLEAR ''EXPECTED''
014534  011104                                  MOV     (R1),R4                 ;READ CONTROL OUT REGISTER
014536  001402                                  BEQ     4$                      ;BR IF ZERO
014540  104002                                  HLT     2                       ;DATA ERROR BIT13 NOT CLEARED
014542  104401                                  SCOP1                           ;SW09 UP?
014544  104400                          4$:     SCOPE                           ;SCOPE THIS TEST


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

                                                ;  TEST 24
                                                ;--------------
014546  012737  000024  001226          TST24:  MOV     #24,TSTNO
014554  012737  014672  001216                  MOV     #TST25,NEXT
014562  012737  014602  001220                  MOV     #64$,LOCK
014570  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
014572  013701  001412                          MOV     DMP04,R1                        ;PUT REGISTER ADDRESS IN R1
014576  012700  000001                          MOV     #1,R0                   ;START WITH BIT0
014602                                  64$:
014602  010005                                  MOV     R0,R5                   ;PUT ''EXPECTED'' IN R5
014604  010511                                  MOV     R5,(R1)                 ;WRITE PORT4 REGISTER
014606  011104                                  MOV     (R1),R4                 ;READ PORT4 REGISTER
014610  020504                                  CMP     R5,R4                   ;COMPARE EXPECTED AND FOUND
014612  001401                                  BEQ     65$                     ;BR IF OK
014614  104002                                  HLT     2                       ;WRITE/READ ERROR
014616  104401                          65$:    SCOP1                           ;LOOP TO 64$ IF SW09-1
014620  000241                                  CLC                             ;CLEAR CARRY
014622  006100                                  ROL     R0                      ;SHIFT TO NEXT BIT
014624  001366                                  BNE     64$                     ;BR IF NOT DONE YET?
014626  012737  014640  001220                  MOV     #66$,LOCK               ;NEW SCOP1
014634  012700  000001                          MOV     #1,R0                   ;START WITH BIT0
014640                                  66$:
014640  005100                                  COM     R0                      ;CHANGE TO A FLOATING ZERO
014642  010005                                  MOV     R0,R5                   ;PUT ''EXPECTED'' IN R5
014644  010511                                  MOV     R5,(R1)                 ;WRITE PORT4 REGISTER
014646  011104                                  MOV     (R1),R4                 ;READ PORT4 REGISTER
014650  020504                                  CMP     R5,R4                   ;COMPARE EXPECTED AND FOUND
014652  001401                                  BEQ     67$                     ;BR IF OK
014654  104002                                  HLT     2                       ;WRITE/READ ERROR
014656  104401                          67$:    SCOP1                           ;LOOP TO 66$ IF SW09 1
014660  005100                                  COM     R0                      ;CHANGE BACK TO A FLOATING ONE
014662  000241                                  CLC                             ;CLEAR CARRY
014664  006100                                  ROL     R0                      ;SHIFT TO NEXT BIT
014666  001364                                  BNE     66$                     ;BR IF NOT DONE YET?
014670  104400                                  SCOPE                           ;SCOPE THIS TEST


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

```
                                                ;*FLOAT A ONE THROUGH PORT6 REGISTER
                                                ;*FLOAT A ZERO THROUGH PORT6 REGISTER
                                                ;;*************************************************************


                                                ;  TEST 25
                                                ;--------------
014672  012737  000025  001226          TST25:  MOV     #25,TSTNO
014700  012737  015016  001216                  MOV     #TST26,NEXT
014706  012737  014726  001220                  MOV     #64$,LOCK
014714  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
014716  013701  001414                          MOV     DMP06,R1                ;PUT REGISTER ADDRESS IN R1
014722  012700  000001                          MOV     #1,R0           ;START WITH BIT0
014726                                  64$:
014726  010005                                  MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
014730  010511                                  MOV     R5,(R1)         ;WRITE PORT6 REGISTER
014732  011104                                  MOV     (R1),R4         ;READ PORT6 REGISTER
014734  020504                                  CMP     R5,R4           ;COMPARE EXPECTED AND FOUND
014736  001401                                  BEQ     65$             ;BR IF OK
014740  104002                                  HLT     2               ;WRITE/READ ERROR
014742  104401                          65$:    SCOP1                   ;LOOP TO 64$ IF SW09=1
014744  000241                                  CLC                     ;CLEAR CARRY
014746  006100                                  ROL     R0              ;SHIFT TO NEXT BIT
014750  001366                                  BNE     64$             ;BR IF NOT DONE YET?
014752  012737  014764  001220                  MOV     #66$,LOCK       ;NEW SCOP1
014760  012700  000001                          MOV     #1,R0           ;START WITH BIT0
014764                                  66$:
014764  005100                                  COM     R0              ;CHANGE TO A FLOATING ZERO
014766  010005                                  MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
014770  010511                                  MOV     R5,(R1)         ;WRITE PORT6 REGISTER
014772  011104                                  MOV     (R1),R4         ;READ PORT6 REGISTER
014774  020504                                  CMP     R5,R4           ;COMPARE EXPECTED AND FOUND
014776  001401                                  BEQ     67$             ;BR IF OK
015000  104002                                  HLT     2               ;WRITE/READ ERROR
015002  104401                          67$:    SCOP1                   ;LOOP TO 66$ IF SW09=1
015004  005100                                  COM     R0              ;CHANGE BACK TO A FLOATING ONE
015006  000241                                  CLC                     ;CLEAR CARRY
015010  006100                                  ROL     R0              ;SHIFT TO NEXT BIT
015012  001364                                  BNE     66$             ;BR IF NOT DONE YET?
015014  104400                                  SCOPE                   ;SCOPE THIS TEST



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


                                                ;  TEST 26
                                                ;--------------
015016  012737  000026  001226          TST26:  MOV     #26,TSTNO
015024  012737  015146  001216                  MOV     #TST27,NEXT
015032  012737  015046  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015040  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
015042  012700  000001                          MOV     #1,R0           ;START PATTERN AT 1
015046  105011                          1$:     CLRB    (R1)            ;CLEAR REGISTER
```

```
015050 110005                          MOVB    R0,R5           ;PUT DATA IN 'EXPECTED''
015052 110011                          MOVB    R0,(R1)         ;WRITE DMC REGISTER WITH PATTERN
015054 111104                          MOVB    (R1),R4         ;READ DMC REGISTER INTO 'FOUND''
015056 020504                          CMP     R5,R4           ;IS DATA CORRECT
015060 001401                          BEQ     2$              ;BR IF YES
015062 104002                          HLT     2               ;DATA ERROR
015064 104401                  2$:     SCOP1                   ;SW09=1?
015066 105721                          TSTB    (R1)+           ;NEXT REGISTER
015070 005200                          INC     R0              ;INCREMENT DATA PATTERN
015072 022700 000011                   CMP     #11,R0          ;LAST REGISTER?
015076 001363                          BNE     1$              ;BR IF NO
015100 013701 001404                   MOV     DMCSR,R1        ;BASE DMC11  ADDRESS TO R1
015104 012700 000001                   MOV     #1,R0           ;RESTART PATTERN AT 1
015110 012737 015116 001220            MOV     #3$,LOCK        ;NEW SCOP1
015116 110005                  3$:     MOVB    R0,R5           ;PUT DATA IN 'EXPECTED''
015120 111104                          MOVB    (R1),R4         ;READ DMC REGISTER INTO 'FOUND''
015122 020504                          CMP     R5,R4           ;IS DATA CORRECT
015124 001401                          BEQ     4$              ;BR IF YES
015126 104002                          HLT     2               ;DUAL ADDRESSING ERROR
015130 104401                  4$:     SCOP1                   ;SW09=1?
015132 105721                          TSTB    (R1)+           ;NEXT REGISTER
015134 005200                          INC     R0              ;INCREMENT PATTERN
015136 022700 000011                   CMP     #11,R0          ;LAST REGISTER?
015142 001365                          BNE     3$              ;BR IF NO
015144 104400                          SCOPE                   ;SCOPE THIS TEST



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

                                       ;  TEST 27
                                       ;--------------
015146 012737 000027 001226    TST27:  MOV     #27,TSTNO
015154 012737 015306 001216            MOV     #TST30,NEXT
015162 012737 015200 001220            MOV     #1$,LOCK
                                                               ;R1 CONTAINS BASE DMC11 ADDRESS
015170 104412                          MSTCLR                  ;MASTER CLEAR DMC11
015172 012711 003000                   MOV     #BIT9.BIT10,(R1) ;SEL6 IS NOW THE IR
015176 005005                          CLR     R5              ;PUT 'EXPECTED'' IN R5
015200 010561 000006           1$:     MOV     R5,6(R1)        ;CLEAR THE IR
015204 016104 000006                   MOV     6(R1),R4        ;READ THE IR
015210 020504                          CMP     R5,R4           ;IS IT CLEARED?
015212 001401                          BEQ     2$              ;BR IF YES
015214 104023                          HLT     23              ;ERROR IR IS NOT CLEAR
015216 104401                  2$:     SCOP1                   ;LOOP TO 1$ IF SW09=1
015220 012737 015232 001220            MOV     #3$,LOCK        ;NEW SCOP1
015226 012705 177777                   MOV     #-1,R5          ;PUT 'EXPECTED'' IN R5
015232 010561 000006           3$:     MOV     R5,6(R1)        ;WRITE ALL ONES TO THE IR
015236 016104 000006                   MOV     6(R1),R4        ;READ THE IR
015242 020504                          CMP     R5,R4           ;IS IT ALL ONES?
015244 001401                          BEQ     4$              ;BR IF YES
015246 104023                          HLT     23              ;ERROR IR IS NOT ALL ONES
015250 104401                  4$:     SCOP1                   ;LOOP TO 3$ IF SW09=1
```

```
015252  012737  015262  001220           MOV     #5$,LOCK           ;NEW SCOP1
015260  005005                           CLR     R5                 ;PUT 'EXPECTED'' IN R5
015262  000005                   5$:     RESET                      ;BUS RESET
015264  012711  003000                   MOV     #BIT9!BIT10,(R1)   ;SEL6 IS IR
015270  016104  000006                   MOV     6(R1),R4           ;READ THE IR
015274  020504                           CMP     R5,R4              ;IS IT CLEARED?
015276  001401                           BEQ     6$                 ;BR IF YES
015300  104023                           HLT     23                 ;ERROR, IR IS NOT CLEARED
015302  104401                   6$:     SCOP1                      ;LOOP TO 5$ IF SW09-1
015304  104400                           SCOPE                      ;SCOPE THIS TEST


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

                                         ;  TEST 30
                                         ;-------------
015306  012737  000030  001226   TST30:  MOV     #30,TSTNO
015314  012737  015450  001216           MOV     #TST31,NEXT
015322  012737  015340  001220           MOV     #1$,LOCK
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
015330  104412                           MSTCLR                     ;MASTER CLEAR DMC11
015332  012711  003000                   MOV     #BIT9.BIT10,(R1)   ;SEL6 IS NOW THE IR
015336  005005                           CLR     R5                 ;PUT 'EXPECTED'' IN R5
015340  010561  000006           1$:     MOV     R5,6(R1)           ;CLEAR THE IR
015344  016104  000006                   MOV     6(R1),R4           ;READ THE IR
015350  020504                           CMP     R5,R4              ;IS IT CLEARED?
015352  001401                           BEQ     2$                 ;BR IF YES
015354  104023                           HLT     23                 ;ERROR IR IS NOT CLEAR
015356  104401                   2$:     SCOP1                      ;LOOP TO 1$ IF SW09-1
015360  012737  015372  001220           MOV     #3$,LOCK           ;NEW SCOP1
015366  012705  177777                   MOV     #-1,R5             ;PUT 'EXPECTED'' IN R5
015372  010561  000006           3$:     MOV     R5,6(R1)           ;WRITE ALL ONES TO THE IR
015376  016104  000006                   MOV     6(R1),R4           ;READ THE IR
015402  020504                           CMP     R5,R4              ;IS IT ALL ONES?
015404  001401                           BEQ     4$                 ;BR IF YES
015406  104023                           HLT     23                 ;ERROR IR IS NOT - ALL ONES
015410  104401                   4$:     SCOP1                      ;LOOP TO 3$ IF SW09=1
015412  012737  015422  001220           MOV     #5$,LOCK           ;NEW SCOP1
015420  005005                           CLR     R5                 ;PUT 'EXPECTED'' IN R5
015422  052711  040000           5$:     BIS     #BIT14,(R1)        ;MASTER CLEAR
015426  012711  003000                   MOV     #BIT9.BIT10,(R1)   ;SEL6 IS IR
015432  016104  000006                   MOV     6(R1),R4           ;READ THE IR
015436  020504                           CMP     R5,R4              ;IS IT CLEARED?
015440  001401                           BEQ     6$                 ;BR IF YES
015442  104023                           HLT     23                 ;ERROR, IR IS NOT CLEARED
015444  104401                   6$:     SCOP1                      ;LOOP TO 5$ IF SW09=1
015446  104400                           SCOPE                      ;SCOPE THIS TEST


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

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

                                              ;  TEST 31
                                              ;---------------
015450  012737  000031  001226  TST31:  MOV     #31,TSTNO
015456  012737  015534  001216          MOV     #TST32,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015464  104412                          MSTCLR                          ;MASTER CLEAR DMC11
015466  012761  000377  000004          MOV     #377,4(R1)              ;PORT4 HI-BYTE-0'S LO-BYTE=1'S
015474  012711  001000                  MOV     #BIT9,(R1)              ;SET ROMI
015500  012761  121105  000006          MOV     #121105,6(R1)           ;INSTRUCTION TO PORT6
015506  052711  001400                  BIS     #BIT8!BIT9,(R1)         ;CLK INTSRUCTION, MOVE IBUS*4 TO IBUS*5
015512  000240                          NOP                             ;
015514  012705  177777                  MOV     #-1,R5                  ;PUT ''EXPECTED'' IN R5
015520  016104  000004                  MOV     4(R1),R4                ;PUT ''FOUND'' INTO R4
015524  020504                          CMP     R5,R4                   ;IS DATA CORRECT
015526  001401                          BEQ     1$                      ;BR IF YES
015530  104003                          HLT     3                       ;ERROR
015532  104400                  1$:     SCOPE                           ;SCOPE THIS TEST


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

                                              ;  TEST 32
                                              ;---------------
015534  012737  000032  001226  TST32:  MOV     #32,TSTNO
015542  012737  015734  001216          MOV     #TST33,NEXT
015550  012737  015570  001220          MOV     #64$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
015556  104412                          MSTCLR                          ;MASTER CLEAR DMC11
015560  012702  000000                  MOV     #0,R2                   ;SAVE REGISTER ADDRESS FOR TYPEOUT
015564  012700  000001                  MOV     #1,R0                   ;START WITH BIT 0
015570                          64$:
015570  010061  000004                  MOV     R0,4(R1)                ;PUT PATTERN INTO PORT4
015574  042761  000030  000004          BIC     #30,4(R1)               ;CLEAR UNWANTED BITS
015602  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
015604  121100                          121100!0                        ;MOV DATA TO IBUS* REGISTER 0
015606  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
015610  121005                          121005!<0*20>                   ;READ FROM IBUS* REGISTER 0
015612  010005                          MOV     R0,R5                   ;PUT EXPECTED IN R5
015614  042705  000030                  BIC     #30,R5                  ;CLEAR UNWANTED BITS
015620  116104  000005                  MOVB    5(R1),R4                ;PUT ''FOUND'' INTO R4
015624  120504                          CMPB    R5,R4                   ;DATA CORRECT?
015626  001401                          BEQ     65$                     ;BR IF YES
015630  104004                          HLT     4                       ;ERROR
015632  104401                  65$:    SCOP1                           ;SW09-1?
015634  000241                          CLC                             ;CLEAR CARRY
015636  106100                          ROLB    R0                      ;SHIFT BIT IN R0
015640  001353                          BNE     64$                     ;IF R0=0 THEN DONE
```

```
015642  012737  015656  001220          MOV     #67$,LOCK           ;NEW SCOP1
015650  012700  000001                  MOV     #1,R0               ;START WITH BIT 0
015654  005100                  69$:    COM     R0                  ;CHANGE TO FLOATING ZERO
015656                          67$:
015656  010061  000004                  MOV     R0,4(R1)            ;PUT PATTERN INTO PORT4
015662  042761  000030  000004          BIC     #30,4(R1)           ;CLEAR UNWANTED BITS
015670  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
015672  121100                          121100!0                    ;MOV DATA TO IBUS* REGISTER 0
015674  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
015676  121005                          121005!<0*20>               ;READ FROM IBUS* REGISTER 0
015700  010005                          MOV     R0,R5               ;PUT EXPECTED IN R5
015702  042705  000030                  BIC     #30,R5              ;CLEAR UNWANTED BITS
015706  116104  000005                  MOVB    5(R1),R4            ;PUT ''FOUND'' INTO R4
015712  120504                          CMPB    R5,R4               ;DATA CORRECT?
015714  001401                          BEQ     68$                 ;BR IF YES
015716  104004                          HLT     4                   ;ERROR
015720  104401                  68$:    SCOP1                       ;SW09=1?
015722  005100                          COM     R0                  ;CHANGE TO FLOATING 1
015724  000241                          CLC                         ;CLEAR CARRY
015726  106100                          ROLB    R0                  ;SHIFT BIT IN R0
015730  001351                          BNE     69$                 ;IF R0=0 THEN DONE
015732  104400                          SCOPE                       ;SCOPE THIS TEST


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

                                        ;  TEST 33
                                        ;--------------
015734  012737  000033  001226  TST33:  MOV     #33,TSTNO
015742  012737  016134  001216          MOV     #TST34,NEXT
015750  012737  015770  001220          MOV     #64$,LOCK
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
015756  104412                          MSTCLR                      ;MASTER CLEAR DMC11
015760  012702  000002                  MOV     #2,R2               ;SAVE REGISTER ADDRESS FOR TYPEOUT
015764  012700  000001                  MOV     #1,R0               ;START WITH BIT 0
015770                          64$:
015770  010061  000004                  MOV     R0,4(R1)            ;PUT PATTERN INTO PORT4
015774  042761  000070  000004          BIC     #70,4(R1)           ;CLEAR UNWANTED BITS
016002  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016004  121102                          121100!2                    ;MOV DATA TO IBUS* REGISTER 2
016006  104414                          ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016010  121045                          121005.<2*20>               ;READ FROM IBUS* REGISTER 2
016012  010005                          MOV     R0,R5               ;PUT EXPECTED IN R5
016014  042705  000070                  BIC     #70,R5              ;CLEAR UNWANTED BITS
016020  116104  000005                  MOVB    5(R1),R4            ;PUT ''FOUND'' INTO R4
016024  120504                          CMPB    R5,R4               ;DATA CORRECT?
016026  001401                          BEQ     65$                 ;BR IF YES
016030  104004                          HLT     4                   ;ERROR
016032  104401                  65$:    SCOP1                       ;SW09-1?
016034  000241                          CLC                         ;CLEAR CARRY
016036  106100                          ROLB    R0                  ;SHIFT BIT IN R0
016040  001353                          BNE     64$                 ;IF R0 0 THEN DONE
```

```
016042  012737  016056  001220           MOV     #67$,LOCK           ;NEW SCOP1
016050  012700  000001                   MOV     #1,R0               ;START WITH BIT 0
016054  005100                   69$:    COM     R0                  ;CHANGE TO FLOATING ZERO
016056                           67$:
016056  010061  000004                   MOV     R0,4(R1)            ;PUT PATTERN INTO PORT4
016062  042761  000070  000004           BIC     #70,4(R1)           ;CLEAR UNWANTED BITS
016070  104414                           ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016072  121102                           121100.2                    ;MOV DATA TO IBUS* REGISTER 2
016074  104414                           ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016076  121045                           121005!<2*20>               ;READ FROM IBUS* REGISTER 2
016100  010005                           MOV     R0,R5               ;PUT EXPECTED IN R5
016102  042705  000070                   BIC     #70,R5              ;CLEAR UNWANTED BITS
016106  116104  000005                   MOVB    5(R1),R4            ;PUT ''FOUND'' INTO R4
016112  120504                           CMPB    R5,R4               ;DATA CORRECT?
016114  001401                           BEQ     68$                 ;BR IF YES
016116  104004                           HLT     4                   ;ERROR
016120  104401                   68$:    SCOP1                       ;SW09=1?
016122  005100                           COM     R0                  ;CHANGE TO FLOATING 1
016124  000241                           CLC                         ;CLEAR CARRY
016126  106100                           ROLB    R0                  ;SHIFT BIT IN R0
016130  001351                           BNE     69$                 ;IF R0=0 THEN DONE
016132  104400                           SCOPE                       ;SCOPE THIS TEST


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

                                         ;  TEST 34
                                         ;--------------
016134  012737  000034  001226   TST34:  MOV     #34,TSTNO
016142  012737  016310  001216           MOV     #TST35,NEXT
016150  012737  016170  001220           MOV     #64$,LOCK
                                                                     ;R1 CONTAINS BASE DMC11 ADDRESS
016156  104412                           MSTCLR                      ;MASTER CLEAR DMC11
016160  012702  000004                   MOV     #4,R2               ;SAVE REGISTER ADDRESS FOR TYPEOUT
016164  012700  000001                   MOV     #1,R0               ;START WITH BIT 0
016170                           64$:
016170  010061  000004                   MOV     R0,4(R1)            ;PUT PATTERN INTO PORT4
016174  104414                           ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016176  121104                           121100.4                    ;MOV DATA TO IBUS* REGISTER 4
016200  104414                           ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016202  121105                           121005.<4*20>               ;READ FROM IBUS* REGISTER 4
016204  010005                           MOV     R0,R5               ;PUT EXPECTED IN R5
016206  116104  000005                   MOVB    5(R1),R4            ;PUT ''FOUND'' INTO R4
016212  120504                           CMPB    R5,R4               ;DATA CORRECT?
016214  001401                           BEQ     65$                 ;BR IF YES
016216  104004                           HLT     4                   ;ERROR
016220  104401                   65$:    SCOP1                       ;SW09=1?
016222  000241                           CLC                         ;CLEAR CARRY
016224  106100                           ROLB    R0                  ;SHIFT BIT IN R0
016226  001360                           BNE     64$                 ;IF R0=0 THEN DONE
016230  012737  016244  001220           MOV     #67$,LOCK           ;NEW SCOP1
016236  012700  000001                   MOV     #1,R0               ;START WITH BIT 0
```

```
016242  005100                        69$:   COM     R0              ;CHANGE TO FLOATING ZERO
016244                                67$:
016244  010061  000004                       MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
016250  104414                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016252  121104                               121100.4                ;MOV DATA TO IBUS* REGISTER 4
016254  104414                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016256  121105                               121005!<4*20>           ;READ FROM IBUS* REGISTER 4
016260  010005                               MOV     R0,R5           ;PUT EXPECTED IN R5
016262  116104  000005                       MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
016266  120504                               CMPB    R5,R4           ;DATA CORRECT?
016270  001401                               BEQ     68$             ;BR IF YES
016272  104004                               HLT     4               ;ERROR
016274  104401                        68$:   SCOP1                   ;SW09=1?
016276  005100                               COM     R0              ;CHANGE TO FLOATING 1
016300  000241                               CLC                     ;CLEAR CARRY
016302  106100                               ROLB    R0              ;SHIFT BIT IN R0
016304  001356                               BNE     69$             ;IF R0-0 THEN DONE
016306  104400                               SCOPE                   ;SCOPE THIS TEST



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

                                             ;  TEST 35
                                             ;---------------
016310  012737  000035  001226        TST35: MOV     #35,TSTNO
016316  012737  016464  001216               MOV     #TST36,NEXT
016324  012737  016344  001220               MOV     #64$,LOCK
                                                                     ;R1 CONTAINS BASE DMC11 ADDRESS
016332  104412                               MSTCLR                  ;MASTER CLEAR DMC11
016334  012702  000005                       MOV     #5,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
016340  012700  000001                       MOV     #1,R0           ;START WITH BIT 0
016344                                64$:
016344  010061  000004                       MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
016350  104414                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016352  121105                               121100 5                ;MOV DATA TO IBUS* REGISTER 5
016354  104414                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016356  121125                               121005!<5*20>           ;READ FROM IBUS* REGISTER 5
016360  010005                               MOV     R0,R5           ;PUT EXPECTED IN R5
016362  116104  000005                       MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
016366  120504                               CMPB    R5,R4           ;DATA CORRECT?
016370  001401                               BEQ     65$             ;BR IF YES
016372  104004                               HLT     4               ;ERROR
016374  104401                        65$:   SCOP1                   ;SW09-1?
016376  000241                               CLC                     ;CLEAR CARRY
016400  106100                               ROLB    R0              ;SHIFT BIT IN R0
016402  001360                               BNE     64$             ;IF R0-0 THEN DONE
016404  012737  016420  001220               MOV     #67$,LOCK       ;NEW SCOP1
016412  012700  000001                       MOV     #1,R0           ;START WITH BIT 0
016416  005100                        69$:   COM     R0              ;CHANGE TO FLOATING ZERO
016420                                67$:
016420  010061  000004                       MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
016424  104414                               ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
016426 121105                             121100!5              ;MOV DATA TO IBUS* REGISTER 5
016430 104414                             ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016432 121125                             121005.<5*20>         ;READ FROM IBUS* REGISTER 5
016434 010005                             MOV     R0,R5         ;PUT EXPECTED IN R5
016436 116104 000005                      MOVB    5(R1),R4      ;PUT "FOUND" INTO R4
016442 120504                             CMPB    R5,R4         ;DATA CORRECT?
016444 001401                             BEQ     68$           ;BR IF YES
016446 104004                             HLT     4             ;ERROR
016450 104401                     68$:    SCOP1                 ;SW09=1?
016452 005100                             COM     R0            ;CHANGE TO FLOATING 1
016454 000241                             CLC                   ;CLEAR CARRY
016456 106100                             ROLB    R0            ;SHIFT BIT IN R0
016460 001356                             BNE     69$           ;IF R0=0 THEN DONE
016462 104400                             SCOPE                 ;SCOPE THIS TEST


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

                                          ;  TEST 36
                                          ;--------------
016464 012737 000036 001226       TST36:  MOV     #36,TSTNO
016472 012737 016664 001216               MOV     #TST37,NEXT
016500 012737 016520 001220               MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
016506 104412                             MSTCLR                ;MASTER CLEAR DMC11
016510 012702 000010                      MOV     #10,R2        ;SAVE REGISTER ADDRESS FOR TYPEOUT
016514 012700 000001                      MOV     #1,R0         ;START WITH BIT 0
016520                            64$:
016520 010061 000004                      MOV     R0,4(R1)      ;PUT PATTERN INTO PORT4
016524 042761 000141 000004               BIC     #141,4(R1)    ;CLEAR UNWANTED BITS
016532 104414                             ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016534 121110                             121100!10             ;MOV DATA TO IBUS* REGISTER 10
016536 104414                             ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016540 121205                             121005!<10*20>        ;READ FROM IBUS* REGISTER 10
016542 010005                             MOV     R0,R5         ;PUT EXPECTED IN R5
016544 042705 000141                      BIC     #141,R5       ;CLEAR UNWANTED BITS
016550 116104 000005                      MOVB    5(R1),R4      ;PUT "FOUND" INTO R4
016554 120504                             CMPB    R5,R4         ;DATA CORRECT?
016556 001401                             BEQ     65$           ;BR IF YES
016560 104004                             HLT     4             ;ERROR
016562 104401                     65$:    SCOP1                 ;SW09=1?
016564 000241                             CLC                   ;CLEAR CARRY
016566 106100                             ROLB    R0            ;SHIFT BIT IN R0
016570 001353                             BNE     64$           ;IF R0=0 THEN DONE
016572 012737 016606 001220               MOV     #67$,LOCK     ;NEW SCOP1
016600 012700 000001                      MOV     #1,R0         ;START WITH BIT 0
016604 005100                     69$:    COM     R0            ;CHANGE TO FLOATING ZERO
016606                            67$:
016606 010061 000004                      MOV     R0,4(R1)      ;PUT PATTERN INTO PORT4
016612 042761 000141 000004               BIC     #141,4(R1)    ;CLEAR UNWANTED BITS
016620 104414                             ROMCLK                ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
016622  121110                          121100.10               ;MOV DATA TO IBUS* REGISTER 10
016624  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
016626  121205                          121005!<10*20>          ;READ FROM IBUS* REGISTER 10
016630  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
016632  042705  000141                  BIC     #141,R5         ;CLEAR UNWANTED BITS
016636  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
016642  120504                          CMPB    R5,R4           ;DATA CORRECT?
016644  001401                          BEQ     68$             ;BR IF YES
016646  104004                          HLT     4               ;ERROR
016650  104401                  68$:    SCOP1                   ;SW09-1?
016652  005100                          COM     R0              ;CHANGE TO FLOATING 1
016654  000241                          CLC                     ;CLEAR CARRY
016656  106100                          ROLB    R0              ;SHIFT BIT IN R0
016660  001351                          BNE     69$             ;IF R0 0 THEN DONE
016662  104400                          SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 37
                                        ;--------------
016664  012737  000037  001226  TST37:  MOV     #37,TSTNO
016672  012737  017104  001216          MOV     #TST40,NEXT
016700  012737  016720  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
016706  104412                          MSTCLR                  ;MASTER CLEAR DMC11
016710  012702  000011                  MOV     #11,R2          ;SAVE REGISTER ADDRESS FOR TYPEOUT
016714  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
016720                          64$:
016720  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
016724  042761  000262  000004          BIC     #262,4(R1)      ;CLEAR UNWANTED BITS
016732  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016734  121111                          121100 11               ;MOV DATA TO IBUS* REGISTER 11
016736  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
016740  121225                          121005!<11*20>          ;READ FROM IBUS* REGISTER 11
016742  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
016744  042705  000262                  BIC     #262,R5         ;CLEAR UNWANTED BITS
016750  052705  000020                  BIS     #20,R5          ;ADD THESE BITS
016754  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' INTO R4
016760  052704  000020                  BIS     #20,R4          ;ADD THIS BIT
016764  120504                          CMPB    R5,R4           ;DATA CORRECT?
016766  001401                          BEQ     65$             ;BR IF YES
016770  104004                          HLT     4               ;ERROR
016772  104401                  65$:    SCOP1                   ;SW09=1?
016774  000241                          CLC                     ;CLEAR CARRY
016776  106100                          ROLB    R0              ;SHIFT BIT IN R0
017000  001347                          BNE     64$             ;IF R0=0 THEN DONE
017002  012737  017016  001220          MOV     #67$,LOCK       ;NEW SCOP1
017010  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
017014  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
```

```
017016                                 67$:
017016  010061  000004                         MOV     R0,4(R1)           ;PUT PATTERN INTO PORT4
017022  042761  000262  000004                 BIC     #262,4(R1)         ;CLEAR UNWANTED BITS
017030  104414                                 ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017032  121111                                 121100.11                  ;MOV DATA TO IBUS* REGISTER 11
017034  104414                                 ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017036  121225                                 121005!<11*20>             ;READ FROM IBUS* REGISTER 11
017040  010005                                 MOV     R0,R5              ;PUT EXPECTED IN R5
017042  042705  000262                         BIC     #262,R5            ;CLEAR UNWANTED BITS
017046  052705  000020                         BIS     #20,R5             ;ADD THESE BITS
017052  116104  000005                         MOVB    5(R1),R4           ;PUT 'FOUND'' INTO R4
017056  052704  000020                         BIS     #20,R4             ;ADD THIS BIT
017062  120504                                 CMPB    R5,R4              ;DATA CORRECT?
017064  001401                                 BEQ     68$                ;BR IF YES
017066  104004                                 HLT     4                  ;ERROR
017070  104401                         68$:    SCOP1                      ;SW09=1?
017072  005100                                 COM     R0                 ;CHANGE TO FLOATING 1
017074  000241                                 CLC                        ;CLEAR CARRY
017076  106100                                 ROLB    R0                 ;SHIFT BIT IN R0
017100  001345                                 BNE     69$                ;IF R0=0 THEN DONE
017102  104400                                 SCOPE                      ;SCOPE THIS TEST


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


                                               ;  TEST 40
                                               ;---------------
017104  012737  000040  001226         TST40:  MOV     #40,TSTNO
017112  012737  017260  001216                 MOV     #TST41,NEXT
017120  012737  017140  001220                 MOV     #64$,LOCK
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
017126  104412                                 MSTCLR                     ;MASTER CLEAR DMC11
017130  012702  000000                         MOV     #0,R2              ;SAVE REGISTER ADDRESS FOR TYPEOUT
017134  012700  000001                         MOV     #1,R0              ;START WITH BIT 0
017140                                 64$:
017140  010061  000004                         MOV     R0,4(R1)           ;PUT PATTERN INTO PORT4
017144  104414                                 ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017146  122100                                 122100!0                   ;MOV DATA TO IBUS REGISTER 0
017150  104414                                 ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017152  021005                                 21005!<0*20>               ;READ FROM IBUS REGISTER 0
017154  010005                                 MOV     R0,R5              ;PUT EXPECTED IN R5
017156  116104  000005                         MOVB    5(R1),R4           ;PUT 'FOUND'' INTO R4
017162  120504                                 CMPB    R5,R4              ;DATA CORRECT?
017164  001401                                 BEQ     65$                ;BR IF YES
017166  104005                                 HLT     5                  ;ERROR
017170  104401                         65$:    SCOP1                      ;SW09-1?
017172  000241                                 CLC                        ;CLEAR CARRY
017174  106100                                 ROLB    R0                 ;SHIFT BIT IN R0
017176  001360                                 BNE     64$                ;IF R0=0 THEN DONE
017200  012737  017214  001220                 MOV     #67$,LOCK          ;NEW SCOP1
017206  012700  000001                         MOV     #1,R0              ;START WITH BIT 0
017212  005100                         69$:    COM     R0                 ;CHANGE TO FLOATING ZERO
```

```
017214                           67$:
017214  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017220  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
017222  122100                          122100!0                ;MOV DATA TO IBUS REGISTER 0
017224  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017226  021005                          21005!<0*20>            ;READ FROM IBUS REGISTER 0
017230  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
017232  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
017236  120504                          CMPB    R5,R4           ;DATA CORRECT?
017240  001401                          BEQ     68$             ;BR IF YES
017242  104005                          HLT     5               ;ERROR
017244  104401                   68$:   SCOP1                   ;SW09=1?
017246  005100                          COM     R0              ;CHANGE TO FLOATING 1
017250  000241                          CLC                     ;CLEAR CARRY
017252  106100                          ROLB    R0              ;SHIFT BIT IN R0
017254  001356                          BNE     69$             ;IF R0-0 THEN DONE
017256  104400                          SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 41
                                        ;--------------
017260  012737  000041  001226   TST41: MOV     #41,TSTNO
017266  012737  017434  001216          MOV     #TST42,NEXT
017274  012737  017314  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017302  104412                          MSTCLR                  ;MASTER CLEAR DMC11
017304  012702  000001                  MOV     #1,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
017310  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
017314                           64$:
017314  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017320  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017322  122101                          122100.1                ;MOV DATA TO IBUS REGISTER 1
017324  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017326  021025                          21005.<1*20>            ;READ FROM IBUS REGISTER 1
017330  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
017332  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
017336  120504                          CMPB    R5,R4           ;DATA CORRECT?
017340  001401                          BEQ     65$             ;BR IF YES
017342  104005                          HLT     5               ;ERROR
017344  104401                   65$:   SCOP1                   ;SW09=1?
017346  000241                          CLC                     ;CLEAR CARRY
017350  106100                          ROLB    R0              ;SHIFT BIT IN R0
017352  001360                          BNE     64$             ;IF R0=0 THEN DONE
017354  012737  017370  001220          MOV     #67$,LOCK       ;NEW SCOP1
017362  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
017366  005100                   69$:   COM     R0              ;CHANGE TO FLOATING ZERO
017370                           67$:
017370  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017374  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017376  122101                          122100.1                ;MOV DATA TO IBUS REGISTER 1
```

```
017400  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017402  021025                          21005.<1*20>            ;READ FROM IBUS REGISTER 1
017404  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
017406  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
017412  120504                          CMPB    R5,R4           ;DATA CORRECT?
017414  001401                          BEQ     68$             ;BR IF YES
017416  104005                          HLT     5               ;ERROR
017420  104401                  68$:    SCOP1                   ;SW09=1?
017422  005100                          COM     R0              ;CHANGE TO FLOATING 1
017424  000241                          CLC                     ;CLEAR CARRY
017426  106100                          ROLB    R0              ;SHIFT BIT IN R0
017430  001356                          BNE     69$             ;IF R0=0 THEN DONE
017432  104400                          SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 42
                                        ;--------------
017434  012737  000042  001226  TST42:  MOV     #42,TSTNO
017442  012737  017610  001216          MOV     #TST43,NEXT
017450  012737  017470  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
017456  104412                          MSTCLR                  ;MASTER CLEAR DMC11
017460  012702  000002                  MOV     #2,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
017464  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
017470                          64$:
017470  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017474  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
017476  122102                          122100.2                ;MOV DATA TO IBUS REGISTER 2
017500  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017502  021045                          21005.<2*20>            ;READ FROM IBUS REGISTER 2
017504  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
017506  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
017512  120504                          CMPB    R5,R4           ;DATA CORRECT?
017514  001401                          BEQ     65$             ;BR IF YES
017516  104005                          HLT     5               ;ERROR
017520  104401                  65$:    SCOP1                   ;SW09=1?
017522  000241                          CLC                     ;CLEAR CARRY
017524  106100                          ROLB    R0              ;SHIFT BIT IN R0
017526  001360                          BNE     64$             ;IF R0-0 THEN DONE
017530  012737  017544  001220          MOV     #67$,LOCK       ;NEW SCOP1
017536  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
017542  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
017544                          67$:
017544  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017550  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017552  122102                          122100.2                ;MOV DATA TO IBUS REGISTER 2
017554  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
017556  021045                          21005.<2*20>            ;READ FROM IBUS REGISTER 2
017560  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
017562  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
```

```
017566 120504                                 CMPB    R5,R4           ;DATA CORRECT?
017570 001401                                 BEQ     68$             ;BR IF YES
017572 104005                                 HLT     5               ;ERROR
017574 104401                         68$:    SCOP1                   ;SW09=1?
017576 005100                                 COM     R0              ;CHANGE TO FLOATING 1
017600 000241                                 CLC                     ;CLEAR CARRY
017602 106100                                 ROLB    R0              ;SHIFT BIT IN R0
017604 001356                                 BNE     69$             ;IF R0=0 THEN DONE
017606 104400                                 SCOPE                   ;SCOPE THIS TEST


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

                                              ;  TEST 43
                                              ;--------------
017610 012737 000043 001226           TST43:  MOV     #43,TSTNO
017616 012737 017764 001216                   MOV     #TST44,NEXT
017624 012737 017644 001220                   MOV     #64$,LOCK
                                                                      ;R1 CONTAINS BASE DMC11 ADDRESS
017632 104412                                 MSTCLR                  ;MASTER CLEAR DMC11
017634 012702 000003                          MOV     #3,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
017640 012700 000001                          MOV     #1,R0           ;START WITH BIT 0
017644                                64$:
017644 010061 000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017650 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
017652 122103                                 122100 3                ;MOV DATA TO IBUS REGISTER 3
017654 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017656 021065                                 21005!<3*20>            ;READ FROM IBUS REGISTER 3
017660 010005                                 MOV     R0,R5           ;PUT EXPECTED IN R5
017662 116104 000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
017666 120504                                 CMPB    R5,R4           ;DATA CORRECT?
017670 001401                                 BEQ     65$             ;BR IF YES
017672 104005                                 HLT     5               ;ERROR
017674 104401                         65$:    SCOP1                   ;SW09=1?
017676 000241                                 CLC                     ;CLEAR CARRY
017700 106100                                 ROLB    R0              ;SHIFT BIT IN R0
017702 001360                                 BNE     64$             ;IF R0=0 THEN DONE
017704 012737 017720 001220                   MOV     #67$,LOCK       ;NEW SCOP1
017712 012700 000001                          MOV     #1,R0           ;START WITH BIT 0
017716 005100                         69$:    COM     R0              ;CHANGE TO FLOATING ZERO
017720                                67$:
017720 010061 000004                          MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
017724 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017726 122103                                 122100,3                ;MOV DATA TO IBUS REGISTER 3
017730 104414                                 ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017732 021065                                 21005!<3*20>            ;READ FROM IBUS REGISTER 3
017734 010005                                 MOV     R0,R5           ;PUT EXPECTED IN R5
017736 116104 000005                          MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
017742 120504                                 CMPB    R5,R4           ;DATA CORRECT?
017744 001401                                 BEQ     68$             ;BR IF YES
017746 104005                                 HLT     5               ;ERROR
017750 104401                         68$:    SCOP1                   ;SW09=1?
```

```
017752  005100                           COM     R0              ;CHANGE TO FLOATING 1
017754  000241                           CLC                     ;CLEAR CARRY
017756  106100                           ROLB    R0              ;SHIFT BIT IN R0
017760  001356                           BNE     69$             ;IF R0=0 THEN DONE
017762  104400                           SCOPE                   ;SCOPE THIS TEST


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

                                         ;  TEST 44
                                         ;--------------
017764  012737  000044  001226  TST44:   MOV     #44,TSTNO
017772  012737  020140  001216           MOV     #TST45,NEXT
020000  012737  020020  001220           MOV     #64$,LOCK
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
020006  104412                           MSTCLR                  ;MASTER CLEAR DMC11
020010  012702  000004                   MOV     #4,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
020014  012700  000001                   MOV     #1,R0           ;START WITH BIT 0
020020                          64$:
020020  010061  000004                   MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020024  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
020026  122104                           122100!4                ;MOV DATA TO IBUS REGISTER 4
020030  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
020032  021105                           21005.<4*20>            ;READ FROM IBUS REGISTER 4
020034  010005                           MOV     R0,R5           ;PUT EXPECTED IN R5
020036  116104  000005                   MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
020042  120504                           CMPB    R5,R4           ;DATA CORRECT?
020044  001401                           BEQ     65$             ;BR IF YES
020046  104005                           HLT     5               ;ERROR
020050  104401                  65$:     SCOP1                   ;SW09-1?
020052  000241                           CLC                     ;CLEAR CARRY
020054  106100                           ROLB    R0              ;SHIFT BIT IN R0
020056  001360                           BNE     64$             ;IF R0=0 THEN DONE
020060  012737  020074  001220           MOV     #67$,LOCK       ;NEW SCOP1
020066  012700  000001                   MOV     #1,R0           ;START WITH BIT 0
020072  005100                  69$:     COM     R0              ;CHANGE TO FLOATING ZERO
020074                          67$:
020074  010061  000004                   MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020100  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
020102  122104                           122100.4                ;MOV DATA TO IBUS REGISTER 4
020104  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020106  021105                           21005.<4*20>            ;READ FROM IBUS REGISTER 4
020110  010005                           MOV     R0,R5           ;PUT EXPECTED IN R5
020112  116104  000005                   MOVB    5(R1),R4        ;PUT 'FOUND'' INTO R4
020116  120504                           CMPB    R5,R4           ;DATA CORRECT?
020120  001401                           BEQ     68$             ;BR IF YES
020122  104005                           HLT     5               ;ERROR
020124  104401                  68$:     SCOP1                   ;SW09=1?
020126  005100                           COM     R0              ;CHANGE TO FLOATING 1
020130  000241                           CLC                     ;CLEAR CARRY
020132  106100                           ROLB    R0              ;SHIFT BIT IN R0
020134  001356                           BNE     67$             ;IF R0-0 THEN DONE
```

```
020136  104400                             SCOPE                           ;SCOPE THIS TEST


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

                                           ;  TEST 45
                                           ;---------------
020140  012737  000045  001226   TST45:    MOV     #45,TSTNO
020146  012737  020314  001216             MOV     #TST46,NEXT
020154  012737  020174  001220             MOV     #64$,LOCK
                                                                           ;R1 CONTAINS BASE DMC11 ADDRESS
020162  104412                             MSTCLR                          ;MASTER CLEAR DMC11
020164  012702  000005                     MOV     #5,R2                   ;SAVE REGISTER ADDRESS FOR TYPEOUT
020170  012700  000001                     MOV     #1,R0                   ;START WITH BIT 0
020174                           64$:
020174  010061  000004                     MOV     R0,4(R1)                ;PUT PATTERN INTO PORT4
020200  104414                             ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020202  122105                             122100.5                        ;MOV DATA TO IBUS REGISTER 5
020204  104414                             ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020206  021125                             21005.<5*20>                    ;READ FROM IBUS REGISTER 5
020210  010005                             MOV     R0,R5                   ;PUT EXPECTED IN R5
020212  116104  000005                     MOVB    5(R1),R4                ;PUT 'FOUND'' INTO R4
020216  120504                             CMPB    R5,R4                   ;DATA CORRECT?
020220  001401                             BEQ     65$                     ;BR IF YES
020222  104005                             HLT     5                       ;ERROR
020224  104401                   65$:      SCOP1                           ;SW09-1?
020226  000241                             CLC                             ;CLEAR CARRY
020230  106100                             ROLB    R0                      ;SHIFT BIT IN R0
020232  001360                             BNE     64$                     ;IF R0=0 THEN DONE
020234  012737  020250  001220             MOV     #67$,LOCK               ;NEW SCOP1
020242  012700  000001                     MOV     #1,R0                   ;START WITH BIT 0
020246  005100                   69$:      COM     R0                      ;CHANGE TO FLOATING ZERO
020250                           67$:
020250  010061  000004                     MOV     R0,4(R1)                ;PUT PATTERN INTO PORT4
020254  104414                             ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020256  122105                             122100.5                        ;MOV DATA TO IBUS REGISTER 5
020260  104414                             ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
020262  021125                             21005.<5*20>                    ;READ FROM IBUS REGISTER 5
020264  010005                             MOV     R0,R5                   ;PUT EXPECTED IN R5
020266  116104  000005                     MOVB    5(R1),R4                ;PUT 'FOUND'' INTO R4
020272  120504                             CMPB    R5,R4                   ;DATA CORRECT?
020274  001401                             BEQ     68$                     ;BR IF YES
020276  104005                             HLT     5                       ;ERROR
020300  104401                   68$:      SCOP1                           ;SW09-1?
020302  005100                             COM     R0                      ;CHANGE TO FLOATING 1
020304  000241                             CLC                             ;CLEAR CARRY
020306  106100                             ROLB    R0                      ;SHIFT BIT IN R0
020310  001356                             BNE     69$                     ;IF R0=0 THEN DONE
020312  104400                             SCOPE                           ;SCOPE THIS TEST


                                           ;************************** TEST 46 *************************
```

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

                                                        ;  TEST 46
                                                        ;--------------
020314  012737  000046  001226  TST46:  MOV     #46,TSTNO
020322  012737  020470  001216          MOV     #TST47,NEXT
020330  012737  020350  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
020336  104412                          MSTCLR                  ;MASTER CLEAR DMC11
020340  012702  000006                  MOV     #6,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
020344  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
020350                          64$:
020350  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020354  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020356  122106                          122100!6                ;MOV DATA TO IBUS REGISTER 6
020360  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
020362  021145                          21005!<6*20>            ;READ FROM IBUS REGISTER 6
020364  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
020366  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
020372  120504                          CMPB    R5,R4           ;DATA CORRECT?
020374  001401                          BEQ     65$             ;BR IF YES
020376  104005                          HLT     5               ;ERROR
020400  104401                  65$:    SCOP1                   ;SW09=1?
020402  000241                          CLC                     ;CLEAR CARRY
020404  106100                          ROLB    R0              ;SHIFT BIT IN R0
020406  001360                          BNE     64$             ;IF R0-0 THEN DONE
020410  012737  020424  001220          MOV     #67$,LOCK       ;NEW SCOP1
020416  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
020422  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
020424                          67$:
020424  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020430  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
020432  122106                          122100.6                ;MOV DATA TO IBUS REGISTER 6
020434  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020436  021145                          21005!<6*20>            ;READ FROM IBUS REGISTER 6
020440  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
020442  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND' INTO R4
020446  120504                          CMPB    R5,R4           ;DATA CORRECT?
020450  001401                          BEQ     68$             ;BR IF YES
020452  104005                          HLT     5               ;ERROR
020454  104401                  68$:    SCOP1                   ;SW09-1?
020456  005100                          COM     R0              ;CHANGE TO FLOATING 1
020460  000241                          CLC                     ;CLEAR CARRY
020462  106100                          ROLB    R0              ;SHIFT BIT IN R0
020464  001356                          BNE     69$             ;IF R0-0 THEN DONE
020466  104400                          SCOPE                   ;SCOPE THIS TEST



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

```
                                                ;   TEST 47
                                                ;--------------
020470  012737  000047  001226  TST47:  MOV     #47,TSTNO
020476  012737  020644  001216          MOV     #TST50,NEXT
020504  012737  020524  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
020512  104412                          MSTCLR                  ;MASTER CLEAR DMC11
020514  012702  000007                  MOV     #7,R2           ;SAVE REGISTER ADDRESS FOR TYPEOUT
020520  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
020524                          64$:
020524  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020530  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020532  122107                          122100.7                ;MOV DATA TO IBUS REGISTER 7
020534  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020536  021165                          21005.<7*20>            ;READ FROM IBUS REGISTER 7
020540  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
020542  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
020546  120504                          CMPB    R5,R4           ;DATA CORRECT?
020550  001401                          BEQ     65$             ;BR IF YES
020552  104005                          HLT     5               ;ERROR
020554  104401                  65$:    SCOP1                   ;SW09-1?
020556  000241                          CLC                     ;CLEAR CARRY
020560  106100                          ROLB    R0              ;SHIFT BIT IN R0
020562  001360                          BNE     64$             ;IF R0=0 THEN DONE
020564  012737  020600  001220          MOV     #67$,LOCK       ;NEW SCOP1
020572  012700  000001                  MOV     #1,R0           ;START WITH BIT 0
020576  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
020600                          67$:
020600  010061  000004                  MOV     R0,4(R1)        ;PUT PATTERN INTO PORT4
020604  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020606  122107                          122100.7                ;MOV DATA TO IBUS REGISTER 7
020610  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
020612  021165                          21005!<7*20>            ;READ FROM IBUS REGISTER 7
020614  010005                          MOV     R0,R5           ;PUT EXPECTED IN R5
020616  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" INTO R4
020622  120504                          CMPB    R5,R4           ;DATA CORRECT?
020624  001401                          BEQ     68$             ;BR IF YES
020626  104005                          HLT     5               ;ERROR
020630  104401                  68$:    SCOP1                   ;SW09=1?
020632  005100                          COM     R0              ;CHANGE TO FLOATING 1
020634  000241                          CLC                     ;CLEAR CARRY
020636  106100                          ROLB    R0              ;SHIFT BIT IN R0
020640  001356                          BNE     69$             ;IF R0=0 THEN DONE
020642  104400                          SCOPE                   ;SCOPE THIS TEST

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

                                                ;   TEST 50
                                                ;--------------
020644  012737  000050  001226  TST50:  MOV     #50,TSTNO
```

```
020652  012737  021072  001216          MOV     #TST51,NEXT
020660  012737  020676  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
020666  104412                          MSTCLR                  ;MASTER CLEAR DMC11
020670  012700  000001                  MOV     #1,R0           ;START WITH A ONE
020674  005002                          CLR     R2              ;R2 CONTAINS ADDRESS OF REGISTER
020676  010203                  1$:     MOV     R2,R3           ;R3=REGISTER ADDRESS
020700  010061  000004                  MOV     R0,4(R1)        ;WRITE DATA TO PORT4
020704  042737  000017  020720          BIC     #17,5$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
020712  050337  020720                  BIS     R3,5$           ;ADD ADDRESS TO INSTRUCTION
020716  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
020720  122100                  5$:     122100                  ;MOVE DATA TO IBUS REGISTER
020722  006303                          ASL     R3              ;SHIFT ADDRESS
020724  006303                          ASL     R3              ;4 TIMES TO GET
020726  006303                          ASL     R3              ;IT TO BITS 4-7
020730  006303                          ASL     R3              ;OF NEXT INSTRUCTION
020732  042737  000360  020746          BIC     #360,6$         ;CLEAR ADDRESS FIELD
020740  050337  020746                  BIS     R3,6$           ;ADD ADDRESS TO INSTRUCTION
020744  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
020746  021005                  6$:     21005                   ;READ FROM IBUS REGISTER
020750  010005                          MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
020752  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
020756  120504                          CMPB    R5,R4           ;IS DATA CORRECT?
020760  001401                          BEQ     2$              ;BR IF YES
020762  104005                          HLT     5               ;DATA ERROR
020764  104401                  2$:     SCOP1                   ;SW09-1?
020766  005200                          INC     R0              ;INCREMENT PATTERN
020770  005202                          INC     R2              ;INCREMENT REGISTER ADDRESS
020772  022702  000010                  CMP     #7+1,R2 ;LAST ADDRESS DONE?
020776  001337                          BNE     1$              ;BR IF NO
021000  012737  021016  001220          MOV     #3$,LOCK        ;NEW SCOP1
021006  012700  000001                  MOV     #1,R0           ;RESTART PATTERN TO 1
021012  005002                          CLR     R2              ;RESTART AT ADDRESS 0
021014  005003                          CLR     R3              ;RESTART AT ADDRESS 0
021016  042737  000360  021032  3$:     BIC     #360,7$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
021024  050337  021032                  BIS     R3,7$           ;ADD ADDRESS TO INSTRUCTION
021030  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
021032  021005                  7$:     21005                   ;READ FROM IBUS REGISTER
021034  010005                          MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
021036  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R5
021042  120504                          CMPB    R5,R4           ;DATA CORRECT?
021044  001401                          BEQ     4$              ;BR IF YES
021046  104005                          HLT     5               ;DUAL ADDRESSING ERROR
021050  104401                  4$:     SCOP1                   ;SW09=1?
021052  005200                          INC     R0              ;INCREMENT PATTERN
021054  005202                          INC     R2              ;NEXT ADDRESS
021056  062703  000020                  ADD     #20,R3          ;ADD 1 TO ADDRESS IN R3(SHIFTED 4 TIMES)
021062  022702  000010                  CMP     #7+1,R2 ;LAST ADDRESS DONE?
021066  001353                          BNE     3$              ;BR IF NO
021070  104400                          SCOPE                   ;SCOPE THIS TEST


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

```
                                        ;;*******************************************************************
                                        ;  TEST 51
                                        ;---------------
021072  012737  000051  001226  TST51:  MOV     #51,TSTNO
021100  012737  021242  001216          MOV     #TST52,NEXT
021106  012737  021122  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
021114  104412                          MSTCLR                  ;MASTER CLEAR DMC11
021116  012700  000001                  MOV     #1,R0           ;START PATTERN WITH BIT0
021122                          64$:
021122  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
021126  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021130  120500                          120500                  ;MOVE DATA TO THE BR REGISTER
021132  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
021134  061225                          061225                  ;MOVE BR TO PORT 5
021136  010005                          MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
021140  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
021144  120504                          CMPB    R5,R4           ;DATA CORRECT?
021146  001401                          BEQ     65$             ;BR IF YES
021150  104006                          HLT     6               ;DATA ERROR
021152  104401                  65$:    SCOP1
021154  000241                          CLC                     ;CLEAR CARRY
021156  106100                          ROLB    R0              ;SHIFT BIT IN R0
021160  001360                          BNE     64$             ;DONE IF R0=0
021162  012737  021176  001220          MOV     #67$,LOCK       ;NEW SCOP1
021170  012700  000001                  MOV     #1,R0           ;START PATTERN WITH BIT0
021174  005100                  69$:    COM     R0              ;CHANGE TO FLOATING ZERO
021176                          67$:
021176  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN IN PORT4
021202  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204  120500                          120500                  ;MOVE DATA TO THE BR REGISTER
021206  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021210  061225                          061225                  ;MOVE BR TO PORT 5
021212  010005                          MOV     R0,R5           ;PUT ''EXPECTED'' IN R5
021214  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
021220  120504                          CMPB    R5,R4           ;DATA CORRECT?
021222  001401                          BEQ     68$             ;BR IF YES
021224  104006                          HLT     6               ;DATA ERROR
021226  104401                  68$:    SCOP1
021230  005100                          COM     R0              ;CHANGE BACK TO A ONE
021232  000241                          CLC                     ;CLEAR CARRY
021234  106100                          ROLB    R0              ;SHIFT BIT IN R0
021236  001356                          BNE     69$             ;DONE IF R0=0
021240  104400                          SCOPE                   ;SCOPE THIS TEST
                                        ;*************************** TEST 52 ***************************
                                        ;*SCRATCH PAD TEST
                                        ;*FLOAT A 1 THROUGH EACH SCRATCH PAD LOCATION
                                        ;*FLOAT A 0 THROUGH EACH SCRATCH PAD LOCATION
                                        ;;*******************************************************************
                                        ;  TEST 52
                                        ;---------------
021242  012737  000052  001226  TST52:  MOV     #52,TSTNO
```

```
021250  012737  021510  001216          MOV     #TST53,NEXT
021256  012737  021274  001220          MOV     #64$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
021264  104412                          MSTCLR                  ;MASTER CLEAR DMC11
021266  005002                          CLR     R2              ;START AT ADDRESS ZERO
021270  012700  000001                  MOV     #1,R0           ;START WITH BIT0
021274  042737  000017  021314  64$:    BIC     #17,65$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
021302  050237  021314                  BIS     R2,65$          ;ADD ADDRESS TO INSTRUCTION
021306  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
021312  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
021314  123100                  65$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
021316  042737  000017  021332          BIC     #17,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
021324  050237  021332                  BIS     R2,66$          ;ADD ADDRESS TO INSTRUCTION
021330  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
021332  040600                  66$:    040600                  ;MOV SP TO BR
021334  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
021336  061225                          061225                  ;MOVE BR TO PORT5
021340  010005                          MOV     R0,R5           ;PUT 'EXPECTED'' IN R5
021342  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
021346  120504                          CMPB    R5,R4           ;DATA CORRECT
021350  001401                          BEQ     67$             ;BR IF YES
021352  104007                          HLT     7               ;DATA ERROR
021354  104401                  67$:    SCOP1                   ;SW09 1?
021356  000241                          CLC                     ;CLEAR CARRY
021360  106100                          ROLB    R0              ;SHIFT BIT IN R0
021362  001344                          BNE     64$             ;DONE IF R0=0
021364  012737  021400  001220          MOV     #69$,LOCK       ;NEW SCOP1
021372  012700  000001                  MOV     #1,R0           ;START WITH BIT0
021376  005100                  73$:    COM     R0              ;CHANGE TO FLOATING ZERO
021400  042737  000017  021420  69$:    BIC     #17,70$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
021406  050237  021420                  BIS     R2,70$          ;ADD ADDRESS TO INSTRUCTION
021412  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
021416  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
021420  123100                  70$:    123100                  ;WRITE SCRATCH PAD(ADDRESS IN R2)
021422  042737  000017  021436          BIC     #17,71$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
021430  050237  021436                  BIS     R2,71$          ;ADD ADDRESS TO INSTRUCTION
021434  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021436  040600                  71$:    040600                  ;MOV SP TO BR
021440  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021442  061225                          061225                  ;MOVE BR TO PORT5
021444  010005                          MOV     R0,R5           ;PUT 'EXPECTED'' IN R5
021446  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
021452  120504                          CMPB    R5,R4           ;DATA CORRECT
021454  001401                          BEQ     72$             ;BR IF YES
021456  104007                          HLT     7               ;DATA ERROR
021460  104401                  72$:    SCOP1                   ;SW09=1?
021462  005100                          COM     R0              ;CHANGE BACK TO A ONE
021464  000241                          CLC                     ;CLEAR CARRY
021466  106100                          ROLB    R0              ;SHIFT BIT IN R0
021470  001342                          BNE     73$             ;DONE IF R0=0
021472  012700  000001                  MOV     #1,R0           ;RESTART AT BIT 0
021476  005202                          INC     R2              ;NEXT SP ADDRESS
021500  022702  000020                  CMP     #20,R2  ;LAST ADDRESS?
021504  001273                          BNE     64$             ;BR IF NO
021506  104400                          SCOPE                   ;SCOPE THIS TEST
```

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

                                                ;  TEST 53
                                                ;---------------
021510  012737  000053  001226  TST53:  MOV     #53,TSTNO
021516  012737  021732  001216          MOV     #TST54,NEXT
021524  012737  021542  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
021532  104412                          MSTCLR                  ;MASTER CLEAR DMC11
021534  012700  000001                  MOV     #1,R0           ;START WITH A 1
021540  005003                          CLR     R3              ;ADDRESS 0
021542  010302                  1$:     MOV     R3,R2           ;MOVE ADDRESS TO R2
021544  042737  000017  021564          BIC     #17,2$          ;CLEAR ADDRESS FIELD
021552  050237  021564                  BIS     R2,2$           ;ADD ADDRESS TO INSTRUCTION
021556  010061  000004                  MOV     R0,4(R1)        ;WRITE PATTERN TO PORT4
021562  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021564  123100                  2$:     123100                  ;WRITE SP(ADDRESS IN R2)
021566  042737  000017  021602          BIC     #17,3$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
021574  050237  021602                  BIS     R2,3$           ;ADD ADDRESS TO INSTRUCTION
021600  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
021602  060600                  3$:     60600                   ;MOV SP TO BR
021604  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021606  061225                          61225                   ;MOV BR TO PORT5
021610  010005                          MOV     R0,R5           ;PUT 'EXPECTED'' IN R5
021612  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
021616  120504                          CMPB    R5,R4           ;DATA CORRECT?
021620  001401                          BEQ     4$              ;BR IF YES
021622  104007                          HLT     7               ;DATA ERROR
021624  104401                  4$:     SCOP1                   ;SW09-0
021626  005200                          INC     R0              ;INCREMENT PATTERN
021630  005203                          INC     R3              ;NEXT ADDRESS
021632  022703  000020                  CMP     #20,R3          ;LAST ADDRESS DONE?
021636  001341                          BNE     1$              ;BR IF NO
021640  012737  021654  001220          MOV     #5$,LOCK        ;NEW SCOP1
021646  012700  000001                  MOV     #1,R0           ;RESTART PATTERN AT 1
021652  005003                          CLR     R3              ;RESTART AT ADDRESS ZERO
021654  010302                  5$:     MOV     R3,R2           ;PUT ADDRESS IN R2
021656  042737  000017  021672          BIC     #17,6$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
021664  050237  021672                  BIS     R2,6$           ;ADD ADDRESS TO INSTRUCTION
021670  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
021672  060600                  6$:     60600                   ;MOV SP TO BR
021674  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021676  061225                          61225                   ;MOV BR TO PORT5
021700  010005                          MOV     R0,R5           ;PUT 'EXPECTED'' IN R5
021702  116104  000005                  MOVB    5(R1),R4        ;PUT 'FOUND'' IN R4
021706  120504                          CMPB    R5,R4           ;DATA CORRECT?
021710  001401                          BEQ     7$              ;BR IF YES
021712  104007                          HLT     7               ;SP ADDRESSING ERROR
021714  104401                  7$:     SCOP1                   ;SW09 1?
021716  005200                          INC     R0              ;INCREMENT PATTERN
021720  005203                          INC     R3              ;NEXT ADDRESS
```

```
021722  022703  000020                   CMP     #20,R3   ;LAST ADDRESS DONE?
021726  001352                           BNE     5$                 ;BR IF NO
021730  104400                           SCOPE                      ;SCOPE THIS TEST


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

                                         ;  TEST 54
                                         ;--------------
021732  012737  000054  001226  TST54:   MOV     #54,TSTNO
021740  012737  022026  001216           MOV     #TST55,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
021746  000005                           RESET                      ;BUS RESET
021750  005011                           CLR     (R1)               ;CLEAR RUN
021752  004537  034612                   JSR     R5,SETVEC          ;SET UP VECTORS
021756  022020                           3$                         ;XX0
021760  022016                           2$                         ;XX4
021762     340     340                   .BYTE   340,340            ;LEVEL 7
021764  012737  000340  177776  1$:      MOV     #340,PS            ;PS - LEVEL 7
021772  012761  000200  000004           MOV     #200,4(R1)         ;WRITE PORT4
022000  104414                           ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022002  121111                           121111                     ;SET BR RQ IN IBUS* REG 11
022004  005037  177776                   CLR     PS                 ;ALLOW INTERRUPT
022010  000240                           NOP
022012  104010                           HLT     10                 ;NO INTERRUPT
022014  000403                           BR      4$
022016  104011                  2$:      HLT     11                 ;WRONG VECTOR
022020  012706  001200          3$:      MOV     #STACK,SP          ;RESET STACK
022024  104400                  4$:      SCOPE                      ;SCOPE THIS TEST


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

                                         ;  TEST 55
                                         ;--------------
022026  012737  000055  001226  TST55:   MOV     #55,TSTNO
022034  012737  022120  001216           MOV     #TST56,NEXT
                                                                    ;R1 CONTAINS BASE DMC11 ADDRESS
022042  104412                           MSTCLR                     ;MASTER CLEAR DMC11
022044  004537  034612                   JSR     R5,SETVEC          ;SET UP VECTORS
022050  022110                           2$                         ;XX0
022052  022112                           3$                         ;XX4
022054     340     340                   .BYTE   340,340            ;LEVEL 7
022056  012737  000340  177776  1$:      MOV     #340,PS            ;PS = LEVEL 7
022064  012761  000300  000004           MOV     #300,4(R1)         ;WRITE PORT4
022072  104414                           ROMCLK                     ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
022074  121111                           121111                     ;SET BR RQ IN IBUS* REG 11
022076  005037  177776                   CLR     PS                 ;ALLOW INTERRUPT
022102  000240                           NOP
022104  104010                           HLT     10                 ;NO INTERRUPT
```

```
022106  000403                          BR      4$
022110  104011                  2$:     HLT     11                      ;WRONG VECTOR
022112  012706  001200          3$:     MOV     #STACK,SP               ;RESET STACK
022116  104400                  4$:     SCOPE                           ;SCOPE THIS TEST


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

                                        ;  TEST 56
                                        ;--------------
022120  012737  000056  001226  TST56:  MOV     #56,TSTNO
022126  012737  022240  001216          MOV     #TST57,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
022134  104412                          MSTCLR                          ;MASTER CLEAR DMC11
022136  012702  000340                  MOV     #340,R2                 ;PUT LEVEL 7 IN R2
022142  010237  177776                  MOV     R2,PS                   ;SET PRIORITY TO 7
022146  013700  001366                  MOV     STAT1,R0                ;GET BR LEVEL OF DMC11
022152  006200                          ASR     R0                      ;SHIFT R0 4 TIMES
022154  006200                          ASR     R0                      ;TO GET PROPER LEVEL
022156  006200                          ASR     R0
022160  006200                          ASR     R0
022162  042700  177437                  BIC     #177437,R0              ;CLEAR UNWANTED BITS
022166  004537  034612                  JSR     R5,SETVEC               ;SET UP VECTORS
022172  022234                          2$                              ;A VECTOR
022174  022234                          2$                              ;B VECTOR
022176     340     340                  .BYTE   340,340                 ;PRIORITY 7
022200  012761  000200  000004  4$:     MOV     #200,4(R1)              ;LOAD PORT4
022206  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
022210  121111                          121111                          ;SET BR REQUEST
022212  010237  177776          5$:     MOV     R2,PS                   ;PUT LEVEL IN R2 IN PS
022216  000240                          NOP
022220  020002                          CMP     R0,R2                   ;IS PRESENT PS LEVEL - TO DMC LEVEL
022222  001403                          BEQ     1$                      ;BR IF YES
022224  162702  000040                  SUB     #40,R2                  ;NO GET NEXT LOWER LEVEL IN R2
022230  000770                          BR      5$                      ;AND CONTINUE WITH TEST
022232  104400                  1$:     SCOPE                           ;SCOPE THIS TEST
022234  104020                  2$:     HLT     20                      ;ERROR UNEXPECTED INTERRUPT
022236  000002                          RTI


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

                                        ;  TEST 57
                                        ;--------------
022240  012737  000057  001226  TST57:  MOV     #57,TSTNO
022246  012737  022404  001216          MOV     #TST60,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
022254  104412                          MSTCLR                          ;MASTER CLEAR DMC11
```

```
022256  012702  000340                  MOV     #340,R2         ;PUT LEVEL 7 IN R2
022262  010237  177776                  MOV     R2,PS           ;SET PRIORITY TO 7
022266  013700  001366                  MOV     STAT1,R0        ;GET BR LEVEL OF DMC11
022272  006200                          ASR     R0              ;SHIFT R0 4 TIMES
022274  006200                          ASR     R0              ;TO GET PROPER LEVEL
022276  006200                          ASR     R0
022300  006200                          ASR     R0
022302  042700  177437                  BIC     #177437,R0      ;CLEAR UNWANTED BITS
022306  010002                          MOV     R0,R2           ;PUT DMC LEVEL IN R2
022310  162702  000040                  SUB     #40,R2          ;GET NEXT LOWER LEVEL IN R2
022314  004537  034612                  JSR     R5,SETVEC       ;SET UP VECTORS
022320  022366                          2$                      ;A VECTOR
022322  022374                          3$                      ;B VECTOR
022324     340     340                  .BYTE   340,340         ;PRIORITY 7
022326  012761  000200  000004  4$:     MOV     #200,4(R1)      ;LOAD PORT4
022334  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
022336  121111                          121111                  ;SET BR REQUEST
022340  010237  177776          5$:     MOV     R2,PS           ;PUT LEVEL IN R2 IN PS
022344  000240                          NOP
022346  104010                          HLT     10              ;ERROR, NO INTERRUPT
022350  022702  000140          6$:     CMP     #140,R2         ;IS IT DOWN TO LEVEL 3 YET?
022354  001403                          BEQ     1$              ;YES,DMC DID NOT INTERUPT, ERROR
022356  162702  000040                  SUB     #40,R2          ;PUT NEXT LOWER LEVEL IN R2
022362  000761                          BR      4$              ;CONTINUE TEST
022364  104400                  1$:     SCOPE                   ;SCOPE THIS TEST
022366  012716  022350          2$:     MOV     #6$,(SP)        ;SET UP FOR RTI
022372  000002                          RTI
022374  104011                  3$:     HLT     11              ;ERROR, WRONG VECTOR
022376  012716  022350                  MOV     #6$,(SP)        ;SET UP FOR RTI
022402  000002                          RTI


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

                                        ;  TEST 60
                                        ;--------------
022404  012737  000060  001226  TST60:  MOV     #60,TSTNO
022412  012737  022510  001216          MOV     #TST61,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
022420  000005                          RESET                   ;BUS RESET
022422  005011                          CLR     (R1)            ;CLEAR RUN
022424  005061  000004                  CLR     4(R1)           ;CLR PORT4
022430  004537  034634                  JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
022434  000000                          0                       ;IN DATA
022436  177777                          -1                      ;OUT DATA
022440  022506                          3$                      ;IN BA
022442  022504                          2$                      ;OUT BA
022444  005037  022504                  CLR     2$              ;CLEAR 2$
022450  012761  000021  000004          MOV     #21,4(R1)       ;WRITE PORT4
022456  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022460  121110                          121110                  ;SET NPR BITS IN IBUS* REG 11
022462  000240                          NOP
022464  012705  177777                  MOV     #-1,R5          ;PUT ''EXPECTED'' IN R5
```

```
022470  013704  022504                   MOV     2$,R4           ;PUT ''FOUND'' IN R4
022474  020504                           CMP     R5,R4           ;DATA CORRECT?
022476  001401                           BEQ     4$              ;BR IF YES
022500  104012                           HLT     12              ;ERROR NPR FAILED
022502  104400                   4$:     SCOPE                   ;SCOPE THIS TEST
022504  000000                   2$:     0                       ;OUT BA
022506  000000                   3$:     0                       ;IN BA


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

                                         ;  TEST 61
                                         ;--------------
022510  012737  000061  001226  TST61:  MOV     #61,TSTNO
022516  012737  022624  001216          MOV     #TST62,NEXT
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
022524  104412                           MSTCLR                  ;MASTER CLEAR DMC11
022526  005061  000004                   CLR     4(R1)           ;CLR PORT4
022532  004537  034634                   JSR     R5,NPRSET       ;SET UP IBUS REG 0-7
022536  000000                           0                       ;IN DATA
022540  177777                           -1                      ;OUT DATA
022542  022622                           3$                      ;IN BA
022544  022620                           2$                      ;OUT BA
022546  012737  177777  022622           MOV     #-1,3$          ;PUT DATA IN 3$
022554  012761  000001  000004           MOV     #1,4(R1)        ;WRITE PORT4
022562  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022564  121110                           121110                  ;SET NPR BITS IN IBUS* REG 11
022566  000240                           NOP
022570  012705  177777                   MOV     #-1,R5          ;PUT ''EXPECTED'' IN R5
022574  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022576  021004                           021004                  ;MOVE IN DATA LOW BYTE TO PORT4
022600  104414                           ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
022602  021025                           021025                  ;MOVE IN DATA HIGH BYTE TO PORT5
022604  016104  000004                   MOV     4(R1),R4        ;PUT ''FOUND'' IN R4
022610  020504                           CMP     R5,R4           ;DATA CORRECT?
022612  001401                           BEQ     4$              ;BR IF YES
022614  104012                           HLT     12              ;ERROR NPR FAILED
022616  104400                   4$:     SCOPE                   ;SCOPE THIS TEST
022620  000000                   2$:     0                       ;OUT BA
022622  000000                   3$:     0                       ;IN BA


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

                                         ;  TEST 62
                                         ;--------------
022624  012737  000062  001226  TST62:  MOV     #62,TSTNO
022632  012737  022726  001216          MOV     #TST63,NEXT
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
022640  104412                           MSTCLR                  ;MASTER CLEAR DMC11
```

```
022642  005061  000004                 CLR     4(R1)               ;CLR PORT4
022646  004537  034634                 JSR     R5,NPRSET           ;SET UP IBUS REG 0-7
022652  000000                         0                           ;IN DATA
022654  177777                         -1                          ;OUT DATA
022656  022724                         3$                          ;IN BA
022660  022723                         2$+1             ;OUT BA
022662  005037  022722                 CLR     2$                  ;CLEAR 2$
022666  012761  000221  000004         MOV     #221,4(R1)          ;WRITE PORT4
022674  104414                         ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
022676  121110                         121110                      ;SET NPR BITS IN IBUS* REG 11
022700  000240                         NOP
022702  012705  177400                 MOV     #177400,R5              ;PUT ''EXPECTED'' IN R5
022706  013704  022722                 MOV     2$,R4               ;PUT ''FOUND'' IN R4
022712  020504                         CMP     R5,R4               ;DATA CORRECT?
022714  001401                         BEQ     4$                  ;BR IF YES
022716  104012                         HLT     12                  ;ERROR NPR FAILED
022720  104400                 4$:     SCOPE                       ;SCOPE THIS TEST
022722  000000                 2$:     0                           ;OUT BA
022724  000000                 3$:     0                           ;IN BA


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

                                       ;  TEST 63
                                       ;--------------
022726  012737  000063  001226 TST63:  MOV     #63,TSTNO
022734  012737  023064  001216         MOV     #TST64,NEXT
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
022742  104412                         MSTCLR                      ;MASTER CLEAR DMC11
022744  013737  001412  022772         MOV     DMP04,1$            ;USE SEL4 FOR ADDRESS
022752  013737  001412  022770         MOV     DMP04,2$            ;USE SEL4 FOR ADDRESS
022760  004537  034634                 JSR     R5,NPRSET           ;LOAD BA AND DATA
022764  000000                         0                           ;IN DATA
022766  125252                         125252                      ;OUT DATA
022770  000000                 2$:     0                           ;IN BA
022772  000000                 1$:     0                           ;OUT BA
022774  012761  000014  000004         MOV     #14,4(R1)           ;LOAD SEL 4 WITH OUT BA16 AND 17
023002  104414                         ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023004  121111                         121111                      ;SET OUTBA 16 AND 17
023006  012761  000021  000004         MOV     #21,4(R1)           ;LOAD SEL4
023014  012761  121110  000006         MOV     #121110,6(R1)       ;PUT INSTRUCTION IN SEL6
023022  012711  003000                 MOV     #BIT9!BIT10,(R1)    ;SET CROMI AND CROMO..
023026  052711  000400                 BIS     #BIT8,(R1)          ;CLOCK IT!
023032  000240                         NOP                         ;WAIT FOR NPR
023034  012705  121110                 MOV     #121110,R5          ;PUT ''EXPECTED'' IN R5
023040  104414                         ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
023042  021044                         021044                      ;MOVE OUT DATA LB TO SEL4
023044  104414                         ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023046  021065                         021065                      ;MOVE OUT DATA HB TO SEL5
023050  016104  000004                 MOV     4(R1),R4            ;PUT ''FOUND'' IN R4
023054  020504                         CMP     R5,R4               ;CORRECT RESULTS ?
023056  001401                         BEQ     3$                  ;BR IF YES
```

```
023060  104012                          HLT     12              ;ERROR BA 16 AND 17 FAILED
023062  104400                  3$:     SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 64
                                        ;---------------
023064  012737  000064  001226  TST64:  MOV     #64,TSTNO
023072  012737  023210  001216          MOV     #TST65,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
023100  104412                          MSTCLR                  ;MASTER CLEAR DMC11
023102  013737  001412  023130          MOV     DMP04,1$        ;USE SEL4 FOR ADDRESS
023110  013737  001412  023126          MOV     DMP04,2$        ;USE SEL4 FOR ADDRESS
023116  004537  034634                  JSR     R5,NPRSET       ;LOAD BA AND DATA
023122  000000                          0                       ;IN DATA
023124  125252                          125252                  ;OUT DATA
023126  000000                  2$:     0                       ;IN BA
023130  000000                  1$:     0                       ;OUT BA
023132  012761  000015  000004          MOV     #15,4(R1)       ;LOAD SEL4
023140  012761  121110  000006          MOV     #121110,6(R1)   ;PUT INSTRUCTION IN SEL6
023146  012711  003000                  MOV     #BIT9!BIT10,(R1);SET CROMI AND CROMO..
023152  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK IT!
023156  000240                          NOP                     ;WAIT FOR NPR
023160  012705  121110                  MOV     #121110,R5      ;PUT ''EXPECTED'' IN R5
023164  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023166  021004                          021004                  ;MOVE IN DATA LB TO SEL4
023170  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023172  021025                          021025                  ;MOVE IN DATA HB TO SEL5
023174  016104  000004                  MOV     4(R1),R4        ;PUT ''FOUND'' IN R4
023200  020504                          CMP     R5,R4           ;CORRECT RESULTS ?
023202  001401                          BEQ     3$              ;BR IF YES
023204  104012                          HLT     12              ;ERROR BA 16 AND 17 FAILED
023206  104400                  3$:     SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 65
                                        ;---------------
023210  012737  000065  001226  TST65:  MOV     #65,TSTNO
023216  012737  023320  001216          MOV     #TST66,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
023224  104412                          MSTCLR                  ;MASTER CLEAR DMC11
023226  004537  034634                  JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
023232  000000                          0                       ;IN DATA
023234  000000                          0                       ;OUT DATA
023236  177320                          177320                  ;IN BA
```

```
023240  177320                          177320                  ;OUT BA
023242  012761  000014  000004          MOV     #14,4(R1)       ;SET OUT BA BITS 16+17 IN PORT4
023250  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023252  121111                          121111                  ;SET OUTBA 16 AND 17
023254  012761  000021  000004          MOV     #21,4(R1)       ;SET NPR REQUEST BITS IN PORT4
023262  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023264  121110                          121110                  ;MOV IBUS* 4 TO IBUS* 10
023266  000240                          NOP
023270  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023272  121225                          121225                  ;MOV IBUS*11 TO IBUS*5
023274  012705  000001                  MOV     #1,R5           ;PUT ''EXPECTED'' IN R5
023300  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
023304  042704  177776                  BIC     #177776,R4      ;CLEAR UNWANTED BITS
023310  020504                          CMP     R5,R4           ;DATA CORRECT?
023312  001401                          BEQ     1$              ;BR IF YES
023314  104012                          HLT     12              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
023316  104400                  1$:     SCOPE                   ;SCOPE THIS TEST


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

                                        ;  TEST 66
                                        ;--------------
023320  012737  000066  001226  TST66:  MOV     #66,TSTNO
023326  012737  023426  001216          MOV     #TST67,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
023334  104412                          MSTCLR                  ;MASTER CLEAR DMC11
023336  004537  034634                  JSR     R5,NPRSET       ;LOAD IBUS REGISTERS 0-7
023342  000000                          0                       ;IN DATA
023344  000000                          0                       ;OUT DATA
023346  177320                          177320                  ;IN BA
023350  177320                          177320                  ;OUT BA
023352  005061  000004                  CLR     4(R1)
023356  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023360  121111                          121111                  ;CLEAR NON-EXISTENT BIT
023362  012761  000015  000004          MOV     #15,4(R1)       ;SET NPR REQUEST BITS IN PORT4
023370  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023372  121110                          121110                  ;MOV IBUS* 4 TO IBUS* 10
023374  000240                          NOP
023376  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
023400  121225                          121225                  ;MOV IBUS*11 TO IBUS*5
023402  012705  000001                  MOV     #1,R5           ;PUT ''EXPECTED'' IN R5
023406  116104  000005                  MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
023412  042704  177776                  BIC     #177776,R4      ;CLEAR UNWANTED BITS
023416  020504                          CMP     R5,R4           ;DATA CORRECT?
023420  001401                          BEQ     1$              ;BR IF YES
023422  104012                          HLT     12              ;ERROR NON-EXISTENT MEM BIT FAILED TO SET
023424  104400                  1$:     SCOPE                   ;SCOPE THIS TEST


                                        ;*************************** TEST 67 ***************************
                                        ;*NPR TEST
```

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

                                                 ;  TEST 67
                                                 ;---------------
023426  012737  000067  001226  TST67:  MOV     #67,TSTNO
023434  012737  000003  001222          MOV     #3,ICOUNT
023442  012737  023624  001216          MOV     #TST70,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023450  104412                          MSTCLR                          ;MASTER CLEAR DMC11
023452  005037  023622                  CLR     5$                      ;START FLAG AT 0
023456  005000                          CLR     R0                      ;DATA
023460  012702  036534                  MOV     #CORMAX,R2              ;ADDRESS
023464                          1$:
023464  010037  023514                  MOV     R0,2$                   ;LOAD DATA
023470  010237  023520                  MOV     R2,4$                   ;LOAD BA
023474  032702  000001                  BIT     #BIT0,R2                ;IS BA ODD?
023500  001402                          BEQ     .+6                     ;BR IF NO
023502  000337  023514                  SWAB    2$                      ;IF ODD PUT DATA IN HI-BYTE
023506  004537  034634                  JSR     R5,NPRSET               ;LOAD NPR REGISTERS
023512  000000                          0                               ;IN DATA
023514  000000                  2$:     0                               ;OUT DATA
023516  000000                          0                               ;IN BA
023520  000000                  4$:     0                               ;OUT BA
023522  105012                          CLRB    (R2)                    ;CLEAR MEMORY LOCATION
023524  012761  000221  000004          MOV     #221,4(R1)              ;LOAD PORT4
023532  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023534  121110                          121110                          ;DO THE NPR
023536  000240                          NOP
023540  010005                          MOV     R0,R5                   ;PUT ''EXPECTED'' IN R5
023542  111204                          MOVB    (R2),R4                 ;PUT ''FOUND'' IN R4
023544  120504                          CMPB    R5,R4                   ;IS DATA CORRECT?
023546  001401                          BEQ     3$                      ;BR IF YES
023550  104021                          HLT     21                      ;ERROR, DATA INCORRECT
023552  104401                  3$:     SCOP1
023554  005200                          INC     R0                      ;NEXT CHARACTER
023556  042700  177400                  BIC     #177400,R0              ;USE ONLY LOW BYTE
023562  005737  023622                  TST     5$                      ;HAS MAX MEMORY BEEN REACHED YET?
023566  001402                          BEQ     6$                      ;BR IF NO
023570  005700                          TST     R0                      ;DONE PATTERN?
023572  001412                          BEQ     7$                      ;BR IF YES
023574  005202                  6$:     INC     R2                      ;INC BA
023576  023702  001304                  CMP     MEMLIM,R2               ;REACHED MEMORY LIMIT YET?
023602  001330                          BNE     1$                      ;BR IF NOT
023604  012702  036534                  MOV     #CORMAX,R2              ;RESTART BA AT FIRST ADDRESS
023610  012737  177777  023622          MOV     #-1,5$                  ;SET FLAG TO END TEST AT END OF DATA PATTERN
023616  000722                          BR 1$                           ;CONTINUE
023620  104400                  7$:     SCOPE                           ;SCOPE THIS TEST
023622  000000                  5$:     0                               ;THIS LOCATION IS A FLAG.  IT STARTS AT 0,
                                                                        ;AND IS SET TO -1 WHEN LAST MEMORY ADDRESS
                                                                        ;IS USED. TEST IS THEN ENDED WHEN PATTERN IS FINISHED


                                                 ;******** **************** TEST 70 *************************
                                                 ;*MAIN MEMORY TEST
```

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

                                                 ;   TEST 70
                                                 ;---------------
023624  012737  000070  001226  TST70:  MOV     #70,TSTNO
023632  012737  023752  001216          MOV     #TST71,NEXT
023640  012737  023656  001220          MOV     #65$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023646  104412                          MSTCLR                          ;MASTER CLEAR DMC11
023650  005002                          CLR     R2                      ;START WITH ADDRESS 0
023652  012700  000001          1$:     MOV     #1,R0                   ;START WITH BIT 0
023656  042737  000377  023672  65$:    BIC     #377,66$                        ;CLEAR ADDRESS FIELD OF INSTRUCTION
023664  050237  023672                  BIS     R2,66$                  ;ADD ADDRESS TO INSTRUCTION
023670  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
023672  010000                  66$:    010000                          ;LOAD MAR WITH ADDRESS IN R2
023674  010061  000004                  MOV     R0,4(R1)                ;WRITE PATTERN IN PORT4
023700  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
023702  122500                          122500                          ;MOVE PORT4 TO MEMORY
023704  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
023706  040620                          040620                          ;MOVE MEMORY TO BR
023710  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
023712  061225                          61225                           ;MOVE BR TO PORT5
023714  010005                          MOV     R0,R5                   ;PUT ''EXPECTED'' IN R5
023716  116104  000005                  MOVB    5(R1),R4                ;PUT ''FOUND'' IN R4
023722  120504                          CMPB    R5,R4                   ;DATA CORRECT?
023724  001401                          BEQ     67$                     ;BR IF YES
023726  104013                          HLT     13                      ;DATA ERROR
023730  104401                  67$:    SCOP1                           ;SW09=1?
023732  000241                          CLC                             ;CLEAR CARRY
023734  106100                          ROLB    R0                      ;SHIFT BIT IN R0
023736  001347                          BNE     65$                     ;DONE IF R0=0
023740  005202                          INC     R2                      ;NEXT ADDRESS
023742  022702  000400                  CMP     #400,R2                 ;LAST ADDRESS
023746  001341                          BNE     1$                      ;BR IF NO
023750  104400                          SCOPE                           ;SCOPE THIS TEST



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

                                                 ;   TEST 71
                                                 ;---------------
023752  012737  000071  001226  TST71:  MOV     #71,TSTNO
023760  012737  024104  001216          MOV     #TST72,NEXT
023766  012737  024006  001220          MOV     #65$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
023774  104412                          MSTCLR                          ;MASTER CLEAR DMC11
023776  005002                          CLR     R2                      ;START WITH ADDRESS 0
024000  012700  000001          1$:     MOV     #1,R0                   ;START WITH BIT 0
024004  005100                  64$:    COM     R0                      ;CHANGE TO FLOATING 0
024006  042737  000377  024022  65$:    BIC     #377,66$                        ;CLEAR ADDRESS FIELD OF INSTRUCTION
024014  050237  024022                  BIS     R2,66$                  ;ADD ADDRESS TO INSTRUCTION
024020  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
024022  010000                   66$:    010000                    ;LOAD MAR WITH ADDRESS IN R2
024024  010061  000004                   MOV     R0,4(R1)          ;WRITE PATTERN IN PORT4
024030  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
024032  122500                           122500                    ;MOVE PORT4 TO MEMORY
024034  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024036  040620                           040620                    ;MOVE MEMORY TO BR
024040  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
024042  061225                           61225                     ;MOVE BR TO PORT5
024044  010005                           MOV     R0,R5             ;PUT ''EXPECTED'' IN R5
024046  116104  000005                   MOVB    5(R1),R4          ;PUT ''FOUND'' IN R4
024052  120504                           CMPB    R5,R4             ;DATA CORRECT?
024054  001401                           BEQ     67$               ;BR IF YES
024056  104013                           HLT     13                ;DATA ERROR
024060  104401                   67$:    SCOP1                     ;SW09=1?
024062  005100                           COM     R0                ;CHANGE TO FLOATING 1
024064  000241                           CLC                       ;CLEAR CARRY
024066  106100                           ROLB    R0                ;SHIFT BIT IN R0
024070  001345                           BNE     64$               ;DONE IF R0=0
024072  005202                           INC     R2                ;NEXT ADDRESS
024074  022702  000400                   CMP     #400,R2           ;LAST ADDRESS
024100  001337                           BNE     1$                ;BR IF NO
024102  104400                           SCOPE                     ;SCOPE THIS TEST



                                         ;*************************** TEST 72 ***************************
                                         ;*MAIN MEMORY DUAL ADDRESSING TEST
                                         ;*LOAD EACH MEMORY LOCATION WITH IT'S OWN ADDRESS
                                         ;*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
                                         ;;*************************************************************

                                         ;  TEST 72
                                         ;--------------
024104  012737  000072  001226   TST72:  MOV     #72,TSTNO
024112  012737  024304  001216           MOV     #TST73,NEXT
024120  012737  024132  001220           MOV     #1$,LOCK
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
024126  104412                           MSTCLR                    ;MASTER CLEAR DMC11
024130  005002                           CLR     R2                ;START AT ADDRESS 0
024132  042737  000377  024146   1$:     BIC     #377,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
024140  050237  024146                   BIS     R2,2$             ;ADD ADDRESS TO INSTRUCTION
024144  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024146  010000                   2$:     010000                    ;LOAD MAR
024150  010261  000004                   MOV     R2,4(R1)
024154  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024156  122500                           122500                    ;MOVE PORT4 TO MEMORY
024160  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
024162  040620                           040620                    ;MOVE MEMORY TO THE BR
024164  104414                           ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
024166  061225                           61225                     ;MOV BR TO PORT5
024170  010205                           MOV     R2,R5             ;PUT ''EXPECTED'' IN R5
024172  116104  000005                   MOVB    5(R1),R4          ;PUT ''FOUND'' IN R4
024176  120504                           CMPB    R5,R4             ;DATA CORRECT?
024200  001401                           BEQ     3$                ;BR IF YES
024202  104013                           HLT     13                ;DATA ERROR
024204  104401                   3$:     SCOP1                     ;SW09=1?
024206  005202                           INC     R2                ;NEXT ADDRESS
```

```
        024210  022702  000400                          CMP     #400,R2         ;LAST ADDRESS
        024214  001346                                  BNE     1$              ;BR IF NO
        024216  012737  024226  001220                  MOV     #4$,LOCK        ;NEW SCOPE 1
        024224  005002                                  CLR     R2              ;RESTART AT ADDRESS 0
        024226  042737  000377  024242  4$:             BIC     #377,5$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
        024234  050237  024242                          BIS     R2,5$           ;ADD ADDRESS TO INSTRUCTION
        024240  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        024242  010000                          5$:     010000                  ;LOAD THE MAR
        024244  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
        024246  040620                                  040620                  ;MOVE MEMORY TO THE BR
        024250  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        024252  061225                                  61225                   ;MOV BR TO PORT5
        024254  010205                                  MOV     R2,R5           ;PUT ''EXPECTED'' IN R5
        024256  116104  000005                          MOVB    5(R1),R4        ;PUT ''FOUND'' IN R4
        024262  120504                                  CMPB    R5,R4           ;DATA CORRECT?
        024264  001401                                  BEQ     6$              ;BR IF YES
        024266  104013                                  HLT     13              ;ADDRESSING ERROR
        024270  104401                          6$:     SCOP1                   ;SW09=1?
        024272  005202                                  INC     R2              ;NEXT ADDRESS
        024274  022702  000400                          CMP     #400,R2         ;IS IT THE LAST
        024300  001352                                  BNE     4$              ;BR IF NO
        024302  104400                                  SCOPE                   ;SCOPE THIS TEST


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

                                                        ;  TEST 73
                                                        ;--------------
        024304  012737  000073  001226  TST73:          MOV     #73,TSTNO
        024312  012737  024440  001216                  MOV     #TST74,NEXT
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
        024320  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
        024322  005002                                  CLR     R2              ;START WITH A ZERO
        024324  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        024326  010000                                  010000                  ;LOAD MAR
        024330  032737  100000  001366                  BIT     #BIT15,STAT1    ;DMC?
        024336  001402                                  BEQ     .+6             ;BR IF YES
        024340  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
        024342  004000                                  4000                    ;MAR HI _ 0 (KMC ONLY)
        024344  010261  000004                  1$:     MOV     R2,4(R1)        ;WRITE DATA TO PORT4
        024350  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
        024352  136500                                  136500                  ;LOAD MEM AUTO-INC MAR
        024354  005202                                  INC     R2              ;INCREMENT DATA
        024356  022702  000400                          CMP     #400,R2         ;DONE YET?
        024362  001370                                  BNE     1$              ;BR IF NO
        024364  005002                                  CLR     R2              ;RESTART WITH A ZERO
        024366  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
        024370  010000                                  010000                  ;LOAD MAR
        024372  032737  100000  001366                  BIT     #BIT15,STAT1    ;DMC?
        024400  001402                                  BEQ     .+6             ;BR IF YES
        024402  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
        024404  004000                                  4000                    ;MAR HI _ 0 (KMC ONLY)
```

```
024406                          2$:
024406  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024410  055224                          055224                  ;MOVE MEM TO PORT4
024412  010205                          MOV     R2,R5           ;PUT 'EXPECTED'' IN R5
024414  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND'' IN R4
024420  120504                          CMPB    R5,R4           ;DATA CORRECT?
024422  001401                          BEQ     3$              ;BR IF YES
024424  104014                          HLT     14              ;MAR ERROR
024426  005202                  3$:     INC     R2              ;NEXT ADDRESS
024430  022702  000400                  CMP     #400,R2         ;DONE YET?
024434  001364                          BNE     2$              ;BR IF NO
024436  104400                          SCOPE                   ;SCOPE THIS TEST


                                        ;*************************** TEST 74 ***************************
                                        ;*ALU C BIT TEST
                                        ;*TEST THAT AN ADD OF 377 AND 377 WILL SET THE C BIT
                                        ;:*************************************************************

                                        ;  TEST 74
                                        ;--------------
024440  012737  000074  001226  TST74:  MOV     #74,TSTNO
024446  012737  024552  001216          MOV     #TST75,NEXT
024454  012737  024500  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
024462  104412                          MSTCLR                  ;MASTER CLEAR DMC11
024464  004737  034676                  JSR     PC,MEMLD        ;LOAD MAINMEM DATA
024470  024542                          TDATA                   ;POINTER TO DATA
024472  004737  034732                  JSR     PC,SPLD         ;LOAD SP DATA
024476  024542                          TDATA                   ;POINTER TO DATA
024500                          1$:
024500  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
024502  010000                          010000                  ;MAR 0
024504  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
024506  054400                          054400.<0*20>           ;ADD 377 AND 377, TO SET C BIT
024510  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
024512  040421                          040401.<1*20>           ;ADD 0 AND 0 AND THE C BIT
024514  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
024516  061224                          61224                   ;PUT RESULTS IN PORT4
024520  012705  000001                  MOV     #1,R5           ;PUT 'EXPECTED'' IN R5
024524  016104  000004                  MOV     4(R1),R4        ;PUT 'FOUND'' IN R4
024530  120504                          CMPB    R5,R4           ;DATA CORRECT?
024532  001401                          BEQ     2$              ;BR IF YES
024534  104015                          HLT     15              ;ERROR C BIT NOT SET
024536  104401                  2$:     SCOP1                   ;SW09=1?
024540  104400                          SCOPE                   ;SCOPE THIS TEST
024542     377     000     000  TDATA:  .BYTE   -1,0,0,0,0,0,0,0,0
024545     000     000     000
024550     000     000
                                .EVEN


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

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

                                                   ;  TEST 75
                                                   ;--------------
 024552  012737  000075  001226    TST75:   MOV     #75,TSTNO
 024560  012737  024726  001216             MOV     #TST76,NEXT
 024566  012737  024620  001220             MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
 024574  104412                             MSTCLR                      ;MASTER CLEAR DMC11
 024576  005000                             CLR     R0                  ;MEM + SP ADDRESS
 024600  012702  024716                     MOV     #5$,R2              ;POINTER TO CORRECT DATA
 024604  004737  034676                     JSR     PC,MEMLD            ;LOAD 8 WORDS OF MAIN MEMORY
 024610  035022                             MEMDAT                      ;POINTER TO DATA
 024612  004737  034732                     JSR     PC,SPLD             ;LOAD 8 WORDS OF SP
 024616  035032                             SPDAT                       ;POINTER TO DATA
 024620  004737  034776            1$:      JSR     PC,CLRC             ;CLEAR C BIT!
 024624  042737  000017  024640             BIC     #17,2$              ;CLEAR ADDRESS FIELD OF INSTRUCTION
 024632  050037  024640                     BIS     R0,2$               ;ADD ADDRESS TO INSTRUCTION
 024636  104414                             ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 024640  010000                    2$:      010000                      ;LOAD MAR
 024642  042737  000017  024656             BIC     #17,3$              ;CLEAR ADDRESS OF INSTRUCTION
 024650  050037  024656                     BIS     R0,3$               ;ADD ADDRESS TO INSTRUCTION
 024654  104414                             ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 024656  040620                    3$:      040400!<11*20>              ;BR    SEL B
 024660  104414                             ROMCLK                      ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 024662  061224                             61224                       ;MOVE BR TO PORT4
 024664  111205                             MOVB    (R2),R5             ;PUT ''EXPECTED'' IN R5
 024666  116104  000004                     MOVB    4(R1),R4            ;PUT ''FOUND'' IN R4
 024672  120504                             CMPB    R5,R4               ;DATA CORRECT?
 024674  001401                             BEQ     4$                  ;BR IF YES
 024676  104015                             HLT     15                  ;ALU ERROR
 024700  104401                    4$:      SCOP1                       ;SW09=1?
 024702  005202                             INC     R2                  ;NEXT DATA
 024704  005200                             INC     R0                  ;NEXT ADDRESS
 024706  022700  000010                     CMP     #10,R0              ;DONE YET?
 024712  001342                             BNE     1$                  ;BR IF NO
 024714  104400                             SCOPE                       ;SCOPE THIS TEST
 024716     000     377     000    5$:      .BYTE   0,-1,0,-1,125,252,125,252
 024721     377     125     252
 024724     125     252
                                   .EVEN



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

                                                   ;  TEST 76
                                                   ;--------------
 024726  012737  000076  001226    TST76:   MOV     #76,TSTNO
```

```
024734 012737 025102 001216          MOV     #TST77,NEXT
024742 012737 024774 001220          MOV     #1$,LOCK
                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
024750 104412                        MSTCLR                   ;MASTER CLEAR DMC11
024752 005000                        CLR     R0               ;MEM + SP ADDRESS
024754 012702 025072                 MOV     #5$,R2           ;POINTER TO CORRECT DATA
024760 004737 034676                 JSR     PC,MEMLD         ;LOAD 8 WORDS OF MAIN MEMORY
024764 035022                        MEMDAT                   ;POINTER TO DATA
024766 004737 034732                 JSR     PC,SPLD          ;LOAD 8 WORDS OF SP
024772 035032                        SPDAT                    ;POINTER TO DATA
024774 004737 034776          1$:    JSR     PC,CLRC          ;CLEAR C BIT.
025000 042737 000017 025014          BIC     #17,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
025006 050037 025014                 BIS     R0,2$            ;ADD ADDRESS TO INSTRUCTION
025012 104414                        ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025014 010000                 2$:    010000                   ;LOAD MAR
025016 042737 000017 025032          BIC     #17,3$           ;CLEAR ADDRESS OF INSTRUCTION
025024 050037 025032                 BIS     R0,3$            ;ADD ADDRESS TO INSTRUCTION
025030 104414                        ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025032 040600                 3$:    040400!<10*20>           ;BR    SEL A
025034 104414                        ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
025036 061224                        61224                    ;MOVE BR TO PORT4
025040 111205                        MOVB    (R2),R5          ;PUT ''EXPECTED'' IN R5
025042 116104 000004                 MOVB    4(R1),R4         ;PUT ''FOUND'' IN R4
025046 120504                        CMPB    R5,R4            ;DATA CORRECT?
025050 001401                        BEQ     4$               ;BR IF YES
025052 104015                        HLT     15               ;ALU ERROR
025054 104401                 4$:    SCOP1                    ;SW09=1?
025056 005202                        INC     R2               ;NEXT DATA
025060 005200                        INC     R0               ;NEXT ADDRESS
025062 022700 000010                 CMP     #10,R0           ;DONE YET?
025066 001342                        BNE     1$               ;BR IF NO
025070 104400                        SCOPE                    ;SCOPE THIS TEST
025072    000    000    377   5$:    .BYTE   0,0,-1,-1,125,125,252,252
025075    377    125    125
025100    252    252
                              .EVEN


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

                                     ;  TEST 77
                                     ;--------------
025102 012737 000077 001226   TST77: MOV     #77,TSTNO
025110 012737 025256 001216          MOV     #TST100,NEXT
025116 012737 025150 001220          MOV     #1$,LOCK
                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
025124 104412                        MSTCLR                   ;MASTER CLEAR DMC11
025126 005000                        CLR     R0               ;MEM + SP ADDRESS
025130 012702 025246                 MOV     #5$,R2           ;POINTER TO CORRECT DATA
025134 004737 034676                 JSR     PC,MEMLD         ;LOAD 8 WORDS OF MAIN MEMORY
```

```
025140 035022                           MEMDAT                   ;POINTER TO DATA
025142 004737 034732                    JSR     PC,SPLD          ;LOAD 8 WORDS OF SP
025146 035032                           SPDAT                    ;POINTER TO DATA
025150 004737 034776            1$:     JSR     PC,CLRC          ;CLEAR C BIT.
025154 042737 000017 025170             BIC     #17,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
025162 050037 025170                    BIS     R0,2$            ;ADD ADDRESS TO INSTRUCTION
025166 104414                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025170 010000                   2$:     010000                   ;LOAD MAR
025172 042737 000017 025206             BIC     #17,3$           ;CLEAR ADDRESS OF INSTRUCTION
025200 050037 025206                    BIS     R0,3$            ;ADD ADDRESS TO INSTRUCTION
025204 104414                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025206 040640                   3$:     040400.<12*20>           ;BR   A OR NOTB
025210 104414                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
025212 061224                           61224                    ;MOVE BR TO PORT4
025214 111205                           MOVB    (R2),R5          ;PUT ''EXPECTED'' IN R5
025216 116104 000004                    MOVB    4(R1),R4         ;PUT ''FOUND'' IN R4
025222 120504                           CMPB    R5,R4            ;DATA CORRECT?
025224 001401                           BEQ     4$               ;BR IF YES
025226 104015                           HLT     15               ;ALU ERROR
025230 104401                   4$:     SCOP1                    ;SW09-1?
025232 005202                           INC     R2               ;NEXT DATA
025234 005200                           INC     R0               ;NEXT ADDRESS
025236 022700 000010                    CMP     #10,R0           ;DONE YET?
025242 001342                           BNE     1$               ;BR IF NO
025244 104400                           SCOPE                    ;SCOPE THIS TEST
025246    377    000    377     5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
025251    377    377    125
025254    252    377
                                .EVEN



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

                                        ;  TEST 100
                                        ;-------------
025256 012737 000100 001226     TST100: MOV     #100,TSTNO
025264 012737 025432 001216             MOV     #TST101,NEXT
025272 012737 025324 001220             MOV     #1$,LOCK
                                                                 ;R1 CONTAINS BASE DMC11 ADDRESS
025300 104412                           MSTCLR                   ;MASTER CLEAR DMC11
025302 005000                           CLR     R0               ;MEM + SP ADDRESS
025304 012702 025422                    MOV     #5$,R2           ;POINTER TO CORRECT DATA
025310 004737 034676                    JSR     PC,MEMLD         ;LOAD 8 WORDS OF MAIN MEMORY
025314 035022                           MEMDAT                   ;POINTER TO DATA
025316 004737 034732                    JSR     PC,SPLD          ;LOAD 8 WORDS OF SP
025322 035032                           SPDAT                    ;POINTER TO DATA
025324 004737 034776            1$:     JSR     PC,CLRC          ;CLEAR C BIT!
025330 042737 000017 025344             BIC     #17,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
025336 050037 025344                    BIS     R0,2$            ;ADD ADDRESS TO INSTRUCTION
025342 104414                           ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
025344  010000                          2$:     010000                  ;LOAD MAR
025346  042737  000017  025362                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
025354  050037  025362                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
025360  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025362  040660                          3$:     040400.<13*20>          ;BR   A AND B
025364  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
025366  061224                                  61224                   ;MOVE BR TO PORT4
025370  111205                                  MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
025372  116104  000004                          MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
025376  120504                                  CMPB    R5,R4           ;DATA CORRECT?
025400  001401                                  BEQ     4$              ;BR IF YES
025402  104015                                  HLT     15              ;ALU ERROR
025404  104401                          4$:     SCOP1                   ;SW09=1?
025406  005202                                  INC     R2              ;NEXT DATA
025410  005200                                  INC     R0              ;NEXT ADDRESS
025412  022700  000010                          CMP     #10,R0          ;DONE YET?
025416  001342                                  BNE     1$              ;BR IF NO
025420  104400                                  SCOPE                   ;SCOPE THIS TEST
025422     000     000     000          5$:     .BYTE   0,0,0,-1,125,0,0,252
025425     377     125     000
025430     000     252
                                        .EVEN



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

                                                ;  TEST 101
                                                ;--------------
025432  012737  000101  001226          TST101: MOV     #101,TSTNO
025440  012737  025606  001216                  MOV     #TST102,NEXT
025446  012737  025500  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
025454  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
025456  005000                                  CLR     R0              ;MEM + SP ADDRESS
025460  012702  025576                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
025464  004737  034676                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
025470  035022                                  MEMDAT                  ;POINTER TO DATA
025472  004737  034732                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
025476  035032                                  SPDAT                   ;POINTER TO DATA
025500  004737  034776                  1$:     JSR     PC,CLRC         ;CLEAR C BIT.
025504  042737  000017  025520                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
025512  050037  025520                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
025516  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
025520  010000                          2$:     010000                  ;LOAD MAR
025522  042737  000017  025536                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
025530  050037  025536                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
025534  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
025536  040700                          3$:     040400.<14*20>          ;BR   A OR B
025540  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
025542  061224                                  61224                   ;MOVE BR TO PORT4
```

```
025544  111205                            MOVB    (R2),R5          ;PUT ''EXPECTED'' IN R5
025546  116104  000004                    MOVB    4(R1),R4         ;PUT ''FOUND'' IN R4
025552  120504                            CMPB    R5,R4            ;DATA CORRECT?
025554  001401                            BEQ     4$               ;BR IF YES
025556  104015                            HLT     15               ;ALU ERROR
025560  104401                    4$:     SCOP1                    ;SW09=1?
025562  005202                            INC     R2               ;NEXT DATA
025564  005200                            INC     R0               ;NEXT ADDRESS
025566  022700  000010                    CMP     #10,R0           ;DONE YET?
025572  001342                            BNE     1$               ;BR IF NO
025574  104400                            SCOPE                    ;SCOPE THIS TEST
025576     000     377     377    5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
025601     377     125     377
025604     377     252
                                  .EVEN



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

                                          ;  TEST 102
                                          ;--------------
025606  012737  000102  001226  TST102:   MOV     #102,TSTNO
025614  012737  025762  001216            MOV     #TST103,NEXT
025622  012737  025654  001220            MOV     #1$,LOCK
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
025630  104412                            MSTCLR                   ;MASTER CLEAR DMC11
025632  005000                            CLR     R0               ;MEM + SP ADDRESS
025634  012702  025752                    MOV     #5$,R2           ;POINTER TO CORRECT DATA
025640  004737  034676                    JSR     PC,MEMLD         ;LOAD 8 WORDS OF MAIN MEMORY
025644  035022                            MEMDAT                   ;POINTER TO DATA
025646  004737  034732                    JSR     PC,SPLD          ;LOAD 8 WORDS OF SP
025652  035032                            SPDAT                    ;POINTER TO DATA
025654  004737  034776            1$:     JSR     PC,CLRC          ;CLEAR C BIT.
025660  042737  000017  025674            BIC     #17,2$           ;CLEAR ADDRESS FIELD OF INSTRUCTION
025666  050037  025674                    BIS     R0,2$            ;ADD ADDRESS TO INSTRUCTION
025672  104414                            ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
025674  010000                    2$:     010000                   ;LOAD MAR
025676  042737  000017  025712            BIC     #17,3$           ;CLEAR ADDRESS OF INSTRUCTION
025704  050037  025712                    BIS     R0,3$            ;ADD ADDRESS TO INSTRUCTION
025710  104414                            ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
025712  040720                    3$:     040400!<15*20>           ;BR   A XOR B
025714  104414                            ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
025716  061224                            61224                    ;MOVE BR TO PORT4
025720  111205                            MOVB    (R2),R5          ;PUT ''EXPECTED'' IN R5
025722  116104  000004                    MOVB    4(R1),R4         ;PUT ''FOUND'' IN R4
025726  120504                            CMPB    R5,R4            ;DATA CORRECT?
025730  001401                            BEQ     4$               ;BR IF YES
025732  104015                            HLT     15               ;ALU ERROR
025734  104401                    4$:     SCOP1                    ;SW09-1?
025736  005202                            IN      R2               ;NEXT DATA
```

```
025740  005200                                  INC     R0              ;NEXT ADDRESS
025742  022700  000010                          CMP     #10,R0          ;DONE YET?
025746  001342                                  BNE     1$              ;BR IF NO
025750  104400                                  SCOPE                   ;SCOPE THIS TEST
025752     000     377     377  5$:             .BYTE   0,-1,-1,0,0,-1,-1,0
025755     000     000     377
025760     377     000
                                        .EVEN


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

                                                ;  TEST 103
                                                ;--------------
025762  012737  000103  001226  TST103: MOV     #103,TSTNO
025770  012737  026136  001216          MOV     #TST104,NEXT
025776  012737  026030  001220          MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
026004  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
026006  005000                                  CLR     R0              ;MEM + SP ADDRESS
026010  012702  026126                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
026014  004737  034676                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
026020  035022                                  MEMDAT                  ;POINTER TO DATA
026022  004737  034732                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
026026  035032                                  SPDAT                   ;POINTER TO DATA
026030  004737  034776          1$:             JSR     PC,CLRC         ;CLEAR C BIT!
026034  042737  000017  026050                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
026042  050037  026050                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
026046  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
026050  010000                  2$:             010000                  ;LOAD MAR
026052  042737  000017  026066                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
026060  050037  026066                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
026064  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
026066  040400                  3$:             040400!<00*20>          ;BR    ADD
026070  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
026072  061224                                  61224                   ;MOVE BR TO PORT4
026074  111205                                  MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
026076  116104  000004                          MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
026102  120504                                  CMPB    R5,R4           ;DATA CORRECT?
026104  001401                                  BEQ     4$              ;BR IF YES
026106  104015                                  HLT     15              ;ALU ERROR
026110  104401                  4$:             SCOP1                   ;SW09=1?
026112  005202                                  INC     R2              ;NEXT DATA
026114  005200                                  INC     R0              ;NEXT ADDRESS
026116  022700  000010                          CMP     #10,R0          ;DONE YET?
026122  001342                                  BNE     1$              ;BR IF NO
026124  104400                                  SCOPE                   ;SCOPE THIS TEST
026126     000     377     377  5$:             .BYTE   0,-1,-1,376,252,-1,-1,124
026131     376     252     377
026134     377     124
```

```
                                        .EVEN
                                                ;************************** TEST 104 **************************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED
                                                ;*ALU FUNCTION (A PLUS A PLUS C)     CODE=6
                                                ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                ;;**************************************************************
                                                ;  TEST 104
                                                ;---------------
026136  012737  000104  001226  TST104: MOV     #104,TSTNO
026144  012737  026312  001216          MOV     #TST105,NEXT
026152  012737  026204  001220          MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
026160  104412                          MSTCLR                          ;MASTER CLEAR DMC11
026162  005000                          CLR     R0                      ;MEM + SP ADDRESS
026164  012702  026302                  MOV     #5$,R2                  ;POINTER TO CORRECT DATA
026170  004737  034676                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
026174  035022                          MEMDAT                          ;POINTER TO DATA
026176  004737  034732                  JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
026202  035032                          SPDAT                           ;POINTER TO DATA
026204  004737  034776          1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
026210  042737  000017  026224          BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
026216  050037  026224                  BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
026222  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
026224  010000                  2$:     010000                          ;LOAD MAR
026226  042737  000017  026242          BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
026234  050037  026242                  BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
026240  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
026242  040540                  3$:     040400!<6*20>                   ;BR    2A W/C
026244  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
026246  061224                          61224                           ;MOVE BR TO PORT4
026250  111205                          MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
026252  116104  000004                  MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
026256  120504                          CMPB    R5,R4                   ;DATA CORRECT?
026260  001401                          BEQ     4$                      ;BR IF YES
026262  104015                          HLT     15                      ;ALU ERROR
026264  104401                  4$:     SCOP1                           ;SW09=1?
026266  005202                          INC     R2                      ;NEXT DATA
026270  005200                          INC     R0                      ;NEXT ADDRESS
026272  022700  000010                  CMP     #10,R0                  ;DONE YET?
026276  001342                          BNE     1$                      ;BR IF NO
026300  104400                          SCOPE                           ;SCOPE THIS TEST
026302     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
026305     376     252     252
026310     124     124
                                        .EVEN
                                                ;************************** TEST 105 **************************
                                                ;*ALU TEST
                                                ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED
                                                ;*ALU FUNCTION  (A-B)     CODE=16
```

```
                                                  ;*LOAD MAIN MEM AND SP WITH 8 WORDS OF DATA
                                                  ;*PERFORM THE FUNCTION, VERIFY THE RESULTS
                                                  ;:***********************************************************

                                                  ;  TEST 105
                                                  ;---------------
026312  012737  000105  001226  TST105: MOV     #105,TSTNO
026320  012737  026466  001216          MOV     #TST106,NEXT
026326  012737  026360  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
026334  104412                          MSTCLR                  ;MASTER CLEAR DMC11
026336  005000                          CLR     R0              ;MEM + SP ADDRESS
026340  012702  026456                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
026344  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
026350  035022                          MEMDAT                  ;POINTER TO DATA
026352  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
026356  035032                          SPDAT                   ;POINTER TO DATA
026360  004737  034776          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
026364  042737  000017  026400          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
026372  050037  026400                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
026376  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
026400  010000                  2$:     010000                  ;LOAD MAR
026402  042737  000017  026416          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
026410  050037  026416                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
026414  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
026416  040740                  3$:     040400!<16*20>          ;BR    SUB
026420  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026422  061224                          61224                   ;MOVE BR TO PORT4
026424  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
026426  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
026432  120504                          CMPB    R5,R4           ;DATA CORRECT?
026434  001401                          BEQ     4$              ;BR IF YES
026436  104015                          HLT     15              ;ALU ERROR
026440  104401                  4$:     SCOP1                   ;SW09=1?
026442  005202                          INC     R2              ;NEXT DATA
026444  005200                          INC     R0              ;NEXT ADDRESS
026446  022700  000010                  CMP     #10,R0          ;DONE YET?
026452  001342                          BNE     1$              ;BR IF NO
026454  104400                          SCOPE                   ;SCOPE THIS TEST
026456     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
026461     000     000     253
026464     125     000
                                .EVEN



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

                                                  ;  TEST 106
                                                  ;---------------
026466  012737  000106  001226  TST106: MOV     #106,TSTNO
```

```
026474  012737  026642  001216          MOV     #TST107,NEXT
026502  012737  026534  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
026510  104412                          MSTCLR                  ;MASTER CLEAR DMC11
026512  005000                          CLR     R0              ;MEM + SP ADDRESS
026514  012702  026632                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
026520  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
026524  035022                          MEMDAT                  ;POINTER TO DATA
026526  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
026532  035032                          SPDAT                   ;POINTER TO DATA
026534  004737  034776          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
026540  042737  000017  026554          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
026546  050037  026554                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
026552  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026554  010000                  2$:     010000                  ;LOAD MAR
026556  042737  000017  026572          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
026564  050037  026572                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
026570  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026572  040420                  3$:     040400!<01*20>          ;BR    ADD W/C
026574  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026576  061224                          61224                   ;MOVE BR TO PORT4
026600  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
026602  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
026606  120504                          CMPB    R5,R4           ;DATA CORRECT?
026610  001401                          BEQ     4$              ;BR IF YES
026612  104015                          HLT     15              ;ALU ERROR
026614  104401                  4$:     SCOP1                   ;SW09=1?
026616  005202                          INC     R2              ;NEXT DATA
026620  005200                          INC     R0              ;NEXT ADDRESS
026622  022700  000010                  CMP     #10,R0          ;DONE YET?
026626  001342                          BNE     1$              ;BR IF NO
026630  104400                          SCOPE                   ;SCOPE THIS TEST
026632     000     377     377  5$:     .BYTE   0,-1,-1,376,252,-1,-1,124
026635     376     252     377
026640     377     124
                                .EVEN


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

                                        ;  TEST 107
                                        ;--------------
026642  012737  000107  001226  TST107: MOV     #107,TSTNO
026650  012737  027016  001216          MOV     #TST110,NEXT
026656  012737  026710  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
026664  104412                          MSTCLR                  ;MASTER CLEAR DMC11
026666  005000                          CLR     R0              ;MEM + SP ADDRESS
026670  012702  027006                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
026674  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
026700 035022                         MEMDAT                  ;POINTER TO DATA
026702 004737 034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
026706 035032                         SPDAT                   ;POINTER TO DATA
026710 004737 034776          1$:     JSR     PC,CLRC         ;CLEAR C BIT.
026714 042737 000017 026730           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
026722 050037 026730                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
026726 104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026730 010000                 2$:     010000                  ;LOAD MAR
026732 042737 000017 026746           BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
026740 050037 026746                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
026744 104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026746 040440                 3$:     040400!<2*20>           ;BR   SUB W/C
026750 104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
026752 061224                         61224                   ;MOVE BR TO PORT4
026754 111205                         MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
026756 116104 000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
026762 120504                         CMPB    R5,R4           ;DATA CORRECT?
026764 001401                         BEQ     4$              ;BR IF YES
026766 104015                         HLT     15              ;ALU ERROR
026770 104401                 4$:     SCOP1                   ;SW09=1?
026772 005202                         INC     R2              ;NEXT DATA
026774 005200                         INC     R0              ;NEXT ADDRESS
026776 022700 000010                  CMP     #10,R0          ;DONE YET?
027002 001342                         BNE     1$              ;BR IF NO
027004 104400                         SCOPE                   ;SCOPE THIS TEST
027006    377    000    376   5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
027011    377    377    252
027014    124    377
                              .EVEN

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

                                      ;  TEST 110
                                      ;--------------
027016 012737 000110 001226   TST110: MOV     #110,TSTNO
027024 012737 027172 001216           MOV     #TST111,NEXT
027032 012737 027064 001220           MOV     #1$,LOCK
                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
027040 104412                         MSTCLR                  ;MASTER CLEAR DMC11
027042 005000                         CLR     R0              ;MEM + SP ADDRESS
027044 012702 027162                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
027050 004737 034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
027054 035022                         MEMDAT                  ;POINTER TO DATA
027056 004737 034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
027062 035032                         SPDAT                   ;POINTER TO DATA
027064 004737 034776          1$:     JSR     PC,CLRC         ;CLEAR C BIT.
027070 042737 000017 027104           BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
027076 050037 027104                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
027102 104414                         ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
027104  010000                          2$:     010000                  ;LOAD MAR
027106  042737  000017  027122                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
027114  050037  027122                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
027120  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
027122  040460                          3$:     040400.<3*20>           ;BR    INC A
027124  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027126  061224                                  61224                   ;MOVE BR TO PORT4
027130  111205                                  MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
027132  116104  000004                          MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
027136  120504                                  CMPB    R5,R4           ;DATA CORRECT?
027140  001401                                  BEQ     4$              ;BR IF YES
027142  104015                                  HLT     15              ;ALU ERROR
027144  104401                          4$:     SCOP1                   ;SW09=1?
027146  005202                                  INC     R2              ;NEXT DATA
027150  005200                                  INC     R0              ;NEXT ADDRESS
027152  022700  000010                          CMP     #10,R0          ;DONE YET?
027156  001342                                  BNE     1$              ;BR IF NO
027160  104400                                  SCOPE                   ;SCOPE THIS TEST
027162     001     001     000          5$:     .BYTE   1,1,0,0,126,126,253,253
027165     000     126     126
027170     253     253
                                        .EVEN

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

                                                ;  TEST 111
                                                ;---------------
027172  012737  000111  001226          TST111: MOV     #111,TSTNO
027200  012737  027346  001216                  MOV     #TST112,NEXT
027206  012737  027240  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
027214  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
027216  005000                                  CLR     R0              ;MEM + SP ADDRESS
027220  012702  027336                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
027224  004737  034676                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
027230  035022                                  MEMDAT                  ;POINTER TO DATA
027232  004737  034732                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
027236  035032                                  SPDAT                   ;POINTER TO DATA
027240  004737  034776                  1$:     JSR     PC,CLRC         ;CLEAR C BIT.
027244  042737  000017  027260                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
027252  050037  027260                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
027256  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027260  010000                          2$:     010000                  ;LOAD MAR
027262  042737  000017  027276                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
027270  050037  027276                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
027274  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027276  040520                          3$:     040400.<5*20>           ;BR    2A
027300  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027302  061224                                  61224                   ;MOVE BR TO PORT4
```

```
027304  111205                          MOVB    (R2),R5         ;PUT 'EXPECTED'' IN R5
027306  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
027312  120504                          CMPB    R5,R4           ;DATA CORRECT?
027314  001401                          BEQ     4$              ;BR IF YES
027316  104015                          HLT     15              ;ALU ERROR
027320  104401                  4$:     SCOP1                   ;SW09=1?
027322  005202                          INC     R2              ;NEXT DATA
027324  005200                          INC     R0              ;NEXT ADDRESS
027326  022700  000010                  CMP     #10,R0          ;DONE YET?
027332  001342                          BNE     1$              ;BR IF NO
027334  104400                          SCOPE                   ;SCOPE THIS TEST
027336     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
027341     376     252     252
027344     124     124
                                .EVEN



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

                                        ;  TEST 112
                                        ;---------------
027346  012737  000112  001226  TST112: MOV     #112,TSTNO
027354  012737  027522  001216          MOV     #TST113,NEXT
027362  012737  027414  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
027370  104412                          MSTCLR                  ;MASTER CLEAR DMC11
027372  005000                          CLR     R0              ;MEM + SP ADDRESS
027374  012702  027512                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
027400  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
027404  035022                          MEMDAT                  ;POINTER TO DATA
027406  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
027412  035032                          SPDAT                   ;POINTER TO DATA
027414  004737  034776          1$:     JSR     PC,CLRC         ;CLEAR C BIT!
027420  042737  000017  027434          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
027426  050037  027434                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
027432  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
027434  010000                  2$:     010000                  ;LOAD MAR
027436  042737  000017  027452          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
027444  050037  027452                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
027450  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027452  040500                  3$:     040400.<4*20>           ;BR    A PLUS C
027454  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
027456  061224                          61224                   ;MOVE BR TO PORT4
027460  111205                          MOVB    (R2),R5         ;PUT 'EXPECTED'' IN R5
027462  116104  000004                  MOVB    4(R1),R4        ;PUT 'FOUND'' IN R4
027466  120504                          CMPB    R5,R4           ;DATA CORRECT?
027470  001401                          BEQ     4$              ;BR IF YES
027472  104015                          HLT     15              ;ALU ERROR
027474  104401                  4$:     SCOP1                   ;SW09 1?
027476  005202                          INC     R2              ;NEXT DATA
```

```
027500  005200                             INC     R0                   ;NEXT ADDRESS
027502  022700  000010                     CMP     #10,R0               ;DONE YET?
027506  001342                             BNE     1$                   ;BR IF NO
027510  104400                             SCOPE                        ;SCOPE THIS TEST
027512     000     000     377     5$:     .BYTE   0,0,-1,-1,125,125,252,252
027515     377     125     125
027520     252     252
                                   .EVEN



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


                                           ;  TEST 113
                                           ;--------------
027522  012737  000113  001226  TST113:    MOV     #113,TSTNO
027530  012737  027676  001216             MOV     #TST114,NEXT
027536  012737  027570  001220             MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
027544  104412                             MSTCLR                       ;MASTER CLEAR DMC11
027546  005000                             CLR     R0                   ;MEM + SP ADDRESS
027550  012702  027666                     MOV     #5$,R2               ;POINTER TO CORRECT DATA
027554  004737  034676                     JSR     PC,MEMLD             ;LOAD 8 WORDS OF MAIN MEMORY
027560  035022                             MEMDAT                       ;POINTER TO DATA
027562  004737  034732                     JSR     PC,SPLD              ;LOAD 8 WORDS OF SP
027566  035032                             SPDAT                        ;POINTER TO DATA
027570  004737  034776          1$:        JSR     PC,CLRC              ;CLEAR C BIT.
027574  042737  000017  027610             BIC     #17,2$               ;CLEAR ADDRESS FIELD OF INSTRUCTION
027602  050037  027610                     BIS     R0,2$                ;ADD ADDRESS TO INSTRUCTION
027606  104414                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
027610  010000                  2$:        010000                       ;LOAD MAR
027612  042737  000017  027626             BIC     #17,3$               ;CLEAR ADDRESS OF INSTRUCTION
027620  050037  027626                     BIS     R0,3$                ;ADD ADDRESS TO INSTRUCTION
027624  104414                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
027626  040760                  3$:        040400.<17*20>               ;BR    2'S COMP SUB
027630  104414                             ROMCLK                       ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
027632  061224                             61224                        ;MOVE BR TO PORT4
027634  111205                             MOVB    (R2),R5              ;PUT "EXPECTED" IN R5
027636  116104  000004                     MOVB    4(R1),R4             ;PUT "FOUND" IN R4
027642  120504                             CMPB    R5,R4                ;DATA CORRECT?
027644  001401                             BEQ     4$                   ;BR IF YES
027646  104015                             HLT     15                   ;ALU ERROR
027650  104401                  4$:        SCOP1                        ;SW09-1?
027652  005202                             INC     R2                   ;NEXT DATA
027654  005200                             INC     R0                   ;NEXT ADDRESS
027656  022700  000010                     CMP     #10,R0               ;DONE YET?
027662  001342                             BNE     1$                   ;BR IF NO
027664  104400                             SCOPE                        ;SCOPE THIS TEST
027666     377     000     376     5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
027671     377     377     252
027674     124     377
```

```
                                          .EVEN


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

                                                  ;  TEST 114
                                                  ;-------------
 027676  012737  000114  001226   TST114: MOV     #114,TSTNO
 027704  012737  030052  001216           MOV     #TST115,NEXT
 027712  012737  027744  001220           MOV     #1$,LOCK
                                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
 027720  104412                           MSTCLR                          ;MASTER CLEAR DMC11
 027722  005000                           CLR     RC                      ;MEM + SP ADDRESS
 027724  012702  030042                   MOV     #5$,R2                  ;POINTER TO CORRECT DATA
 027730  004737  034676                   JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
 027734  035022                           MEMDAT                          ;POINTER TO DATA
 027736  004737  034732                   JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
 027742  035032                           SPDAT                           ;POINTER TO DATA
 027744  004737  034776           1$:     JSR     PC,CLRC                 ;CLEAR C BIT!
 027750  042737  000017  027764           BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
 027756  050037  027764                   BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
 027762  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 027764  010000                   2$:     010000                          ;LOAD MAR
 027766  042737  000017  030002           BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
 027774  050037  030002                   BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
 030000  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
 030002  040560                   3$:     040400!<7*20>                   ;BR    DEC A
 030004  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
 030006  061224                           61224                           ;MOVE BR TO PORT4
 030010  111205                           MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
 030012  116104  000004                   MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
 030016  120504                           CMPB    R5,R4                   ;DATA CORRECT?
 030020  001401                           BEQ     4$                      ;BR IF YES
 030022  104015                           HLT     15                      ;ALU ERROR
 030024  104401                   4$:     SCOP1                           ;SW09-1?
 030026  005202                           INC     R2                      ;NEXT DATA
 030030  005200                           INC     R0                      ;NEXT ADDRESS
 030032  022700  000010                   CMP     #10,R0                  ;DONE YET?
 030036  001342                           BNE     1$                      ;BR IF NO
 030040  104400                           SCOPE                           ;SCOPE THIS TEST
 030042     377     377     376   5$:     .BYTE   -1,-1,376,376,124,124,251,251
 030045     376     124     124
 030050     251     251
                                          .EVEN


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

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

                                                ;  TEST 115
                                                ;---------------
030052  012737  000115  001226  TST115: MOV     #115,TSTNO
030060  012737  030226  001216          MOV     #TST116,NEXT
030066  012737  030120  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
030074  104412                          MSTCLR                  ;MASTER CLEAR DMC11
030076  005000                          CLR     R0              ;MEM + SP ADDRESS
030100  012702  030216                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
030104  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
030110  035022                          MEMDAT                  ;POINTER TO DATA
030112  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
030116  035032                          SPDAT                   ;POINTER TO DATA
030120  004737  035010          1$:     JSR     PC,SETC         ;SET C BIT.
030124  042737  000017  030140          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
030132  050037  030140                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
030136  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030140  010000                  2$.     010000                  ;LOAD MAR
030142  042737  000017  030156          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
030150  050037  030156                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
030154  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
030156  040620                  3$:     040400!<11*20>          ;BR   SEL B
030160  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030162  061224                          61224                   ;MOVE BR TO PORT4
030164  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
030166  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
030172  120504                          CMPB    R5,R4           ;DATA CORRECT?
030174  001401                          BEQ     4$              ;BR IF YES
030176  104015                          HLT     15              ;ALU ERROR
030200  104401                  4$:     SCOP1                   ;SW09-1?
030202  005202                          INC     R2              ;NEXT DATA
030204  005200                          INC     R0              ;NEXT ADDRESS
030206  022700  000010                  CMP     #10,R0          ;DONE YET?
030212  001342                          BNE     1$              ;BR IF NO
030214  104400                          SCOPE                   ;SCOPE THIS TEST
030216     000     377     000  5$:     .BYTE   0,-1,0,-1,125,252,125,252
030221     377     125     252
030224     125     252
                                .EVEN



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

                                                ;  TEST 116
                                                ;---------------
030226  012737  000116  001226  TST116: MOV     #116,TSTNO
```

```
030234  012737  030402  001216            MOV     #TST117,NEXT
030242  012737  030274  001220            MOV     #1$,LOCK
                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
030250  104412                            MSTCLR                  ;MASTER CLEAR DMC11
030252  005000                            CLR     R0              ;MEM + SP ADDRESS
030254  012702  030372                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
030260  004737  034676                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
030264  035022                            MEMDAT                  ;POINTER TO DATA
030266  004737  034732                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
030272  035032                            SPDAT                   ;POINTER TO DATA
030274  004737  035010          1$:       JSR     PC,SETC         ;SET C BIT!
030300  042737  000017  030314            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
030306  050037  030314                    BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
030312  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030314  010000                  2$:       010000                  ;LOAD MAR
030316  042737  000017  030332            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
030324  050037  030332                    BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
030330  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
030332  040600                  3$:       040400,<10*20>          ;BR    SEL A
030334  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030336  061224                            61224                   ;MOVE BR TO PORT4
030340  111205                            MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
030342  116104  000004                    MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
030346  120504                            CMPB    R5,R4           ;DATA CORRECT?
030350  001401                            BEQ     4$              ;BR IF YES
030352  104015                            HLT     15              ;ALU ERROR
030354  104401                  4$:       SCOP1                   ;SW09=1?
030356  005202                            INC     R2              ;NEXT DATA
030360  005200                            INC     R0              ;NEXT ADDRESS
030362  022700  000010                    CMP     #10,R0          ;DONE YET?
030366  001342                            BNE     1$              ;BR IF NO
030370  104400                            SCOPE                   ;SCOPE THIS TEST
030372     000     000     377  5$:       .BYTE   0,0,-1,-1,125,125,252,252
030375     377     125     125
030400     252     252
                                .EVEN



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

                                          ;   TEST 117
                                          ;--------------
030402  012737  000117  001226  TST117:   MOV     #117,TSTNO
030410  012737  030556  001216            MOV     #TST120,NEXT
030416  012737  030450  001220            MOV     #1$,LOCK
                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
030424  104412                            MSTCLR                  ;MASTER CLEAR DMC11
030426  005000                            CLR     R0              ;MEM + SP ADDRESS
030430  012702  030546                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
030434  004737  034676                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
030440  035022                                  MEMDAT                          ;POINTER TO DATA
030442  004737  034732                          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
030446  035032                                  SPDAT                           ;POINTER TO DATA
030450  004737  035010                  1$:     JSR     PC,SETC                 ;SET C BIT!
030454  042737  000017  030470                  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
030462  050037  030470                          BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
030466  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030470  010000                          2$:     010000                          ;LOAD MAR
030472  042737  000017  030506                  BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
030500  050037  030506                          BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
030504  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
030506  040640                          3$:     040400!<12*20>                  ;BR    A OR NOTB
030510  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
030512  061224                                  61224                           ;MOVE BR TO PORT4
030514  111205                                  MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
030516  116104  000004                          MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
030522  120504                                  CMPB    R5,R4                   ;DATA CORRECT?
030524  001401                                  BEQ     4$                      ;BR IF YES
030526  104015                                  HLT     15                      ;ALU ERROR
030530  104401                          4$:     SCOP1                           ;SW09=1?
030532  005202                                  INC     R2                      ;NEXT DATA
030534  005200                                  INC     R0                      ;NEXT ADDRESS
030536  022700  000010                          CMP     #10,R0                  ;DONE YET?
030542  001342                                  BNE     1$                      ;BR IF NO
030544  104400                                  SCOPE                           ;SCOPE THIS TEST
030546     377     000     377          5$:     .BYTE   -1,0,-1,-1,-1,125,252,-1
030551     377     377     125
030554     252     377
                                        .EVEN



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

                                                ;   TEST 120
                                                ;--------------
030556  012737  000120  001226  TST120: MOV     #120,TSTNO
030564  012737  030732  001216          MOV     #TST121,NEXT
030572  012737  030624  001220          MOV     #1$,LOCK
                                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
030600  104412                                  MSTCLR                          ;MASTER CLEAR DMC11
030602  005000                                  CLR     R0                      ;MEM + SP ADDRESS
030604  012702  030722                          MOV     #5$,R2                  ;POINTER TO CORRECT DATA
030610  004737  034676                          JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
030614  035022                                  MEMDAT                          ;POINTER TO DATA
030616  004737  034732                          JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
030622  035032                                  SPDAT                           ;POINTER TO DATA
030624  004737  035010                  1$:     JSR     PC,SETC                 ;SET C BIT!
030630  042737  000017  030644                  BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
030636  050037  030644                          BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
030642  104414                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
```

```
030644  010000                          2$:     010000                  ;LOAD MAR
030646  042737  000017  030662                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
030654  050037  030662                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
030660  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030662  040660                          3$:     040400.<13*20>          ;BR     A AND B
030664  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
030666  061224                                  61224                   ;MOVE BR TO PORT4
030670  111205                                  MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
030672  116104  000004                          MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
030676  120504                                  CMPB    R5,R4           ;DATA CORRECT?
030700  001401                                  BEQ     4$              ;BR IF YES
030702  104015                                  HLT     15              ;ALU ERROR
030704  104401                          4$:     SCOP1                   ;SW09=1?
030706  005202                                  INC     R2              ;NEXT DATA
030710  005200                                  INC     R0              ;NEXT ADDRESS
030712  022700  000010                          CMP     #10,R0          ;DONE YET?
030716  001342                                  BNE     1$              ;BR IF NO
030720  104400                                  SCOPE                   ;SCOPE THIS TEST
030722     000     000     000          5$:     .BYTE   0,0,0,-1,125,0,0,252
030725     377     125     000
030730     000     252
                                        .EVEN


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

                                                ;  TEST 121
                                                ;--------------
030732  012737  000121  001226          TST121: MOV     #121,TSTNO
030740  012737  031106  001216                  MOV     #TST122,NEXT
030746  012737  031000  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
030754  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
030756  005000                                  CLR     R0              ;MEM + SP ADDRESS
030760  012702  031076                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
030764  004737  034676                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
030770  035022                                  MEMDAT                  ;POINTER TO DATA
030772  004737  034732                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
030776  035032                                  SPDAT                   ;POINTER TO DATA
031000  004737  035010                  1$:     JSR     PC,SETC         ;SET C BIT!
031004  042737  000017  031020                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031012  050037  031020                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031016  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
031020  010000                          2$:     010000                  ;LOAD MAR
031022  042737  000017  031036                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031030  050037  031036                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031034  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031036  040700                          3$:     040400.<14*20>          ;BR     A OR B
031040  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031042  061224                                  61224                   ;MOVE BR TO PORT4
```

```
031044  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
031046  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
031052  120504                          CMPB    R5,R4           ;DATA CORRECT?
031054  001401                          BEQ     4$              ;BR IF YES
031056  104015                          HLT     15              ;ALU ERROR
031060  104401                  4$:     SCOP1                   ;SW09=1?
031062  005202                          INC     R2              ;NEXT DATA
031064  005200                          INC     R0              ;NEXT ADDRESS
031066  022700  000010                  CMP     #10,R0          ;DONE YET?
031072  001342                          BNE     1$              ;BR IF NO
031074  104400                          SCOPE                   ;SCOPE THIS TEST
031076     000     377     377  5$:     .BYTE   0,-1,-1,-1,125,-1,-1,252
031101     377     125     377
031104     377     252
                                .EVEN


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

                                        ;  TEST 122
                                        ;--------------
031106  012737  000122  001226  TST122: MOV     #122,TSTNO
031114  012737  031262  001216          MOV     #TST123,NEXT
031122  012737  031154  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
031130  104412                          MSTCLR                  ;MASTER CLEAR DMC11
031132  005000                          CLR     R0              ;MEM + SP ADDRESS
031134  012702  031252                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
031140  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031144  035022                          MEMDAT                  ;POINTER TO DATA
031146  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031152  035032                          SPDAT                   ;POINTER TO DATA
031154  004737  035010          1$:     JSR     PC,SETC         ;SET C BIT.
031160  042737  000017  031174          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031166  050037  031174                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031172  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031174  010000                  2$:     010000                  ;LOAD MAR
031176  042737  000017  031212          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031204  050037  031212                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031210  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031212  040720                  3$:     040400.<15*20>          ;BR    A XOR B
031214  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031216  061224                          61224                   ;MOVE BR TO PORT4
031220  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
031222  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
031226  120504                          CMPB    R5,R4           ;DATA CORRECT?
031230  001401                          BEQ     4$              ;BR IF YES
031232  104015                          HLT     15              ;ALU ERROR
031234  104401                  4$:     SCOP1                   ;SW09=1?
031236  005202                          INC     R2              ;NEXT DATA
```

```
031240  005200                                   INC     R0              ;NEXT ADDRESS
031242  022700  000010                           CMP     #10,R0          ;DONE YET?
031246  001342                                   BNE     1$              ;BR IF NO
031250  104400                                   SCOPE                   ;SCOPE THIS TEST
031252     000     377     377  5$:              .BYTE   0,-1,-1,0,0,-1,-1,0
031255     000     000     377
031260     377     000
                                .EVEN


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

                                                 ;  TEST 123
                                                 ;--------------
031262  012737  000123  001226  TST123: MOV      #123,TSTNO
031270  012737  031436  001216          MOV      #TST124,NEXT
031276  012737  031330  001220          MOV      #1$,LOCK
                                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
031304  104412                                   MSTCLR                  ;MASTER CLEAR DMC11
031306  005000                                   CLR     R0              ;MEM + SP ADDRESS
031310  012702  031426                           MOV     #5$,R2          ;POINTER TO CORRECT DATA
031314  004737  034676                           JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031320  035022                                   MEMDAT                  ;POINTER TO DATA
031322  004737  034732                           JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031326  035032                                   SPDAT                   ;POINTER TO DATA
031330  004737  035010          1$:              JSR     PC,SETC         ;SET C BIT.
031334  042737  000017  031350                   BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031342  050037  031350                           BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031346  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031350  010000                  2$:              010000                  ;LOAD MAR
031352  042737  000017  031366                   BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031360  050037  031366                           BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031364  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031366  040400                  3$:              040400!<00*20>          ;BR    ADD
031370  104414                                   ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031372  061224                                   61224                   ;MOVE BR TO PORT4
031374  111205                                   MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
031376  116104  000004                           MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
031402  120504                                   CMPB    R5,R4           ;DATA CORRECT?
031404  001401                                   BEQ     4$              ;BR IF YES
031406  104015                                   HLT     15              ;ALU ERROR
031410  104401                  4$:              SCOP1                   ;SW09-1?
031412  005202                                   INC     R2              ;NEXT DATA
031414  005200                                   INC     R0              ;NEXT ADDRESS
031416  022700  000010                           CMP     #10,R0          ;DONE YET?
031422  001342                                   BNE     1$              ;BR IF NO
031424  104400                                   SCOPE                   ;SCOPE THIS TEST
031426     000     377     377  5$:              .BYTE   0,-1,-1,376,252,-1,-1,124
031431     376     252     377
031434     377     124
```

```
                                           .EVEN


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

                                                  ;   TEST 124
                                                  ;--------------
031436  012737  000124  001226   TST124:  MOV     #124,TSTNO
031444  012737  031612  001216            MOV     #TST125,NEXT
031452  012737  031504  001220            MOV     #1$,LOCK
                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
031460  104412                            MSTCLR                  ;MASTER CLEAR DMC11
031462  005000                            CLR     R0              ;MEM + SP ADDRESS
031464  012702  031602                    MOV     #5$,R2          ;POINTER TO CORRECT DATA
031470  004737  034676                    JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031474  035022                            MEMDAT                  ;POINTER TO DATA
031476  004737  034732                    JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031502  035032                            SPDAT                   ;POINTER TO DATA
031504  004737  035010           1$:      JSR     PC,SETC         ;SET C BIT!
031510  042737  000017  031524            BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031516  050037  031524                    BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031522  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
031524  010000                   2$:      010000                  ;LOAD MAR
031526  042737  000017  031542            BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031534  050037  031542                    BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031540  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031542  040540                   3$:      040400!<6*20>           ;BR    2A W/C
031544  104414                            ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031546  061224                            61224                   ;MOVE BR TO PORT4
031550  111205                            MOVB    (R2),R5         ;PUT "EXPECTED" IN R5
031552  116104  000004                    MOVB    4(R1),R4        ;PUT "FOUND" IN R4
031556  120504                            CMPB    R5,R4           ;DATA CORRECT?
031560  001401                            BEQ     4$              ;BR IF YES
031562  104015                            HLT     15              ;ALU ERROR
031564  104401                   4$:      SCOP1                   ;SW09=1?
031566  005202                            INC     R2              ;NEXT DATA
031570  005200                            INC     R0              ;NEXT ADDRESS
031572  022700  000010                    CMP     #10,R0          ;DONE YET?
031576  001342                            BNE     1$              ;BR IF NO
031600  104400                            SCOPE                   ;SCOPE THIS TEST
031602     001     001     377   5$:      .BYTE   1,1,-1,-1,253,253,125,125
031605     377     253     253
031610     125     125
                                          .EVEN


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

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

                                                ;  TEST 125
                                                ;--------------
031612  012737  000125  001226  TST125: MOV     #125,TSTNO
031620  012737  031766  001216          MOV     #TST126,NEXT
031626  012737  031660  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
031634  104412                          MSTCLR                  ;MASTER CLEAR DMC11
031636  005000                          CLR     R0              ;MEM + SP ADDRESS
031640  012702  031756                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
031644  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
031650  035022                          MEMDAT                  ;POINTER TO DATA
031652  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
031656  035032                          SPDAT                   ;POINTER TO DATA
031660  004737  035010          1$:     JSR     PC,SETC         ;SET C BIT!
031664  042737  000017  031700          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
031672  050037  031700                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
031676  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
031700  010000                  2$:     010000                  ;LOAD MAR
031702  042737  000017  031716          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
031710  050037  031716                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
031714  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
031716  040740                  3$:     040400.<16*20>          ;BR    SUB
031720  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
031722  061224                          61224                   ;MOVE BR TO PORT4
031724  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
031726  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
031732  120504                          CMPB    R5,R4           ;DATA CORRECT?
031734  001401                          BEQ     4$              ;BR IF YES
031736  104015                          HLT     15              ;ALU ERROR
031740  104401                  4$:     SCOP1                   ;SW09-1?
031742  005202                          INC     R2              ;NEXT DATA
031744  005200                          INC     R0              ;NEXT ADDRESS
031746  022700  000010                  CMP     #10,R0          ;DONE YET?
031752  001342                          BNE     1$              ;BR IF NO
031754  104400                          SCOPE                   ;SCOPE THIS TEST
031756     000     001     377  5$:     .BYTE   0,1,-1,0,0,253,125,0
031761     000     000     253
031764     125     000
                                .EVEN



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

                                                ;  TEST 126
                                                ;--------------
031766  012737  000126  001226  TST126: MOV     #126,TSTNO
```

```
031774  012737  032142  001216          MOV     #TST127,NEXT
032002  012737  032034  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
032010  104412                          MSTCLR                  ;MASTER CLEAR DMC11
032012  005000                          CLR     R0              ;MEM + SP ADDRESS
032014  012702  032132                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
032020  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
032024  035022                          MEMDAT                  ;POINTER TO DATA
032026  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
032032  035032                          SPDAT                   ;POINTER TO DATA
032034  004737  035010          1$:     JSR     PC,SETC         ;SET C BIT!
032040  042737  000017  032054          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
032046  050037  032054                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
032052  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032054  010000                  2$:     010000                  ;LOAD MAR
032056  042737  000017  032072          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
032064  050037  032072                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
032070  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032072  040420                  3$:     040400.<01*20>          ;BR   ADD W/C
032074  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
032076  061224                          61224                   ;MOVE BR TO PORT4
032100  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
032102  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
032106  120504                          CMPB    R5,R4           ;DATA CORRECT?
032110  001401                          BEQ     4$              ;BR IF YES
032112  104015                          HLT     15              ;ALU ERROR
032114  104401                  4$:     SCOP1                   ;SW09=1?
032116  005202                          INC     R2              ;NEXT DATA
032120  005200                          INC     R0              ;NEXT ADDRESS
032122  022700  000010                  CMP     #10,R0          ;DONE YET?
032126  001342                          BNE     1$              ;BR IF NO
032130  104400                          SCOPE                   ;SCOPE THIS TEST
032132     001     000     000  5$:     .BYTE   1,0,0,-1,255,0,0,125
032135     377     253     000
032140     000     125
                                .EVEN



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

                                        ;  TEST 127
                                        ;--------------
032142  012737  000127  001226  TST127: MOV     #127,TSTNO
032150  012737  032316  001216          MOV     #TST130,NEXT
032156  012737  032210  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
032164  104412                          MSTCLR                  ;MASTER CLEAR DMC11
032166  005000                          CLR     R0              ;MEM + SP ADDRESS
032170  012702  032306                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
032174  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
```

```
032200  035022                           MEMDAT                          ;POINTER TO DATA
032202  004737  034732                   JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
032206  035032                           SPDAT                           ;POINTER TO DATA
032210  004737  035010          1$:      JSR     PC,SETC                 ;SET C BIT!
032214  042737  000017  032230           BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
032222  050037  032230                   BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
032226  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032230  010000                  2$:      010000                          ;LOAD MAR
032232  042737  000017  032246           BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
032240  050037  032246                   BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
032244  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032246  040440                  3$:      040400!<2*20>                   ;BR   SUB W/C
032250  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032252  061224                           61224                           ;MOVE BR TO PORT4
032254  111205                           MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
032256  116104  000004                   MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
032262  120504                           CMPB    R5,R4                   ;DATA CORRECT?
032264  001401                           BEQ     4$                      ;BR IF YES
032266  104015                           HLT     15                      ;ALU ERROR
032270  104401                  4$:      SCOP1                           ;SW09-1?
032272  005202                           INC     R2                      ;NEXT DATA
032274  005200                           INC     R0                      ;NEXT ADDRESS
032276  022700  000010                   CMP     #10,R0                  ;DONE YET?
032302  001342                           BNE     1$                      ;BR IF NO
032304  104400                           SCOPE                           ;SCOPE THIS TEST
032306     000     001     377  5$:      .BYTE   0,1,-1,0,0,253,125,0
032311     000     000     253
032314     125     000
                                .EVEN


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

                                         ;  TEST 130
                                         ;--------------
032316  012737  000130  001226  TST130:  MOV     #130,TSTNO
032324  012737  032472  001216           MOV     #TST131,NEXT
032332  012737  032364  001220           MOV     #1$,LOCK
                                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
032340  104412                           MSTCLR                          ;MASTER CLEAR DMC11
032342  005000                           CLR     R0                      ;MEM + SP ADDRESS
032344  012702  032462                   MOV     #5$,R2                  ;POINTER TO CORRECT DATA
032350  004737  034676                   JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
032354  035022                           MEMDAT                          ;POINTER TO DATA
032356  004737  034732                   JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
032362  035032                           SPDAT                           ;POINTER TO DATA
032364  004737  035010          1$:      JSR     PC,SETC                 ;SET C BIT!
032370  042737  000017  032404           BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
032376  050037  032404                   BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
032402  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
```

```
032404  010000                          2$:     010000                  ;LOAD MAR
032406  042737  000017  032422                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
032414  050037  032422                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
032420  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
032422  040460                          3$:     040400!<3*20>           ;BR   INC A
032424  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032426  061224                                  61224                   ;MOVE BR TO PORT4
032430  111205                                  MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
032432  116104  000004                          MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
032436  120504                                  CMPB    R5,R4           ;DATA CORRECT?
032440  001401                                  BEQ     4$              ;BR IF YES
032442  104015                                  HLT     15              ;ALU ERROR
032444  104401                          4$:     SCOP1                   ;SW09=1?
032446  005202                                  INC     R2              ;NEXT DATA
032450  005200                                  INC     R0              ;NEXT ADDRESS
032452  022700  000010                          CMP     #10,R0          ;DONE YET?
032456  001342                                  BNE     1$              ;BR IF NO
032460  104400                                  SCOPE                   ;SCOPE THIS TEST
032462     001     001     000          5$:     .BYTE   1,1,0,0,126,126,253,253
032465     000     126     126
032470     253     253
                                        .EVEN


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

                                                ;  TEST 131
                                                ;--------------
032472  012737  000131  001226          TST131: MOV     #131,TSTNO
032500  012737  032646  001216                  MOV     #TST132,NEXT
032506  012737  032540  001220                  MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
032514  104412                                  MSTCLR                  ;MASTER CLEAR DMC11
032516  005000                                  CLR     R0              ;MEM + SP ADDRESS
032520  012702  032636                          MOV     #5$,R2          ;POINTER TO CORRECT DATA
032524  004737  034676                          JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
032530  035022                                  MEMDAT                  ;POINTER TO DATA
032532  004737  034732                          JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
032536  035032                                  SPDAT                   ;POINTER TO DATA
032540  004737  035010                  1$:     JSR     PC,SETC         ;SET C BIT!
032544  042737  000017  032560                  BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
032552  050037  032560                          BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
032556  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
032560  010000                          2$:     010000                  ;LOAD MAR
032562  042737  000017  032576                  BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
032570  050037  032576                          BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
032574  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
032576  040520                          3$:     040400!<5*20>           ;BR   2A
032600  104414                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
032602  061224                                  61224                   ;MOVE BR TO PORT4
```

```
032604  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
032606  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
032612  120504                          CMPB    R5,R4           ;DATA CORRECT?
032614  001401                          BEQ     4$              ;BR IF YES
032616  104015                          HLT     15              ;ALU ERROR
032620  104401                  4$:     SCOP1                   ;SW09=1?
032622  005202                          INC     R2              ;NEXT DATA
032624  005200                          INC     R0              ;NEXT ADDRESS
032626  022700  000010                  CMP     #10,R0          ;DONE YET?
032632  001342                          BNE     1$              ;BR IF NO
032634  104400                          SCOPE                   ;SCOPE THIS TEST
032636     000     000     376  5$:     .BYTE   0,0,376,376,252,252,124,124
032641     376     252     252
032644     124     124
                                .EVEN



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

                                        ;  TEST 132
                                        ;--------------
032646  012737  000132  001226  TST132: MOV     #132,TSTNO
032654  012737  033022  001216          MOV     #TST133,NEXT
032662  012737  032714  001220          MOV     #1$,LOCK
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
032670  104412                          MSTCLR                  ;MASTER CLEAR DMC11
032672  005000                          CLR     R0              ;MEM + SP ADDRESS
032674  012702  033012                  MOV     #5$,R2          ;POINTER TO CORRECT DATA
032700  004737  034676                  JSR     PC,MEMLD        ;LOAD 8 WORDS OF MAIN MEMORY
032704  035022                          MEMDAT                  ;POINTER TO DATA
032706  004737  034732                  JSR     PC,SPLD         ;LOAD 8 WORDS OF SP
032712  035032                          SPDAT                   ;POINTER TO DATA
032714  004737  035010          1$:     JSR     PC,SETC         ;SET C BIT.
032720  042737  000017  032734          BIC     #17,2$          ;CLEAR ADDRESS FIELD OF INSTRUCTION
032726  050037  032734                  BIS     R0,2$           ;ADD ADDRESS TO INSTRUCTION
032732  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
032734  010000                  2$:     010000                  ;LOAD MAR
032736  042737  000017  032752          BIC     #17,3$          ;CLEAR ADDRESS OF INSTRUCTION
032744  050037  032752                  BIS     R0,3$           ;ADD ADDRESS TO INSTRUCTION
032750  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
032752  040500                  3$:     040400!<4*20>           ;BR    A PLUS C
032754  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
032756  061224                          61224                   ;MOVE BR TO PORT4
032760  111205                          MOVB    (R2),R5         ;PUT ''EXPECTED'' IN R5
032762  116104  000004                  MOVB    4(R1),R4        ;PUT ''FOUND'' IN R4
032766  120504                          CMPB    R5,R4           ;DATA CORRECT?
032770  001401                          BEQ     4$              ;BR IF YES
032772  104015                          HLT     15              ;ALU ERROR
032774  104401                  4$:     SCOP1                   ;SW09-1?
032776  005202                          INC     R2              ;NEXT DATA
```

```
033000  005200                           INC     R0                      ;NEXT ADDRESS
033002  022700  000010                   CMP     #10,R0                  ;DONE YET?
033006  001342                           BNE     1$                      ;BR IF NO
033010  104400                           SCOPE                           ;SCOPE THIS TEST
033012     001     001     000   5$:     .BYTE   1,1,0,0,126,126,253,253
033015     000     126     126
033020     253     253
                                 .EVEN



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

                                         ;  TEST 133
                                         ;--------------
033022  012737  000133  001226   TST133: MOV     #133,TSTNO
033030  012737  033176  001216           MOV     #TST134,NEXT
033036  012737  033070  001220           MOV     #1$,LOCK
                                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
033044  104412                           MSTCLR                          ;MASTER CLEAR DMC11
033046  005000                           CLR     R0                      ;MEM + SP ADDRESS
033050  012702  033166                   MOV     #5$,R2                  ;POINTER TO CORRECT DATA
033054  004737  034676                   JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
033060  035022                           MEMDAT                          ;POINTER TO DATA
033062  004737  034732                   JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
033066  035032                           SPDAT                           ;POINTER TO DATA
033070  004737  035010           1$:     JSR     PC,SETC                 ;SET C BIT!
033074  042737  000017  033110           BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
033102  050037  033110                   BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
033106  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033110  010000                   2$:     010000                          ;LOAD MAR
033112  042737  000017  033126           BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
033120  050037  033126                   BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
033124  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
033126  040760                   3$:     040400!<17*20>                  ;BR   2'S COMP SUB
033130  104414                           ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033132  061224                           61224                           ;MOVE BR TO PORT4
033134  111205                           MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
033136  116104  000004                   MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
033142  120504                           CMPB    R5,R4                   ;DATA CORRECT?
033144  001401                           BEQ     4$                      ;BR IF YES
033146  104015                           HLT     15                      ;ALU ERROR
033150  104401                   4$:     SCOP1                           ;SW09=1?
033152  005202                           INC     R2                      ;NEXT DATA
033154  005200                           INC     R0                      ;NEXT ADDRESS
033156  022700  000010                   CMP     #10,R0                  ;DONE YET?
033162  001342                           BNE     1$                      ;BR IF NO
033164  104400                           SCOPE                           ;SCOPE THIS TEST
033166     377     000     376   5$:     .BYTE   -1,0,376,-1,-1,252,124,-1
033171     377     377     252
033174     124     377
```

```
                                                .EVEN


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

                                                        ;  TEST 134
                                                        ;--------------
033176  012737  000134  001226  TST134: MOV     #134,TSTNO
033204  012737  033352  001216          MOV     #TST135,NEXT
033212  012737  033244  001220          MOV     #1$,LOCK
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
033220  104412                          MSTCLR                          ;MASTER CLEAR DMC11
033222  005000                          CLR     R0                      ;MEM + SP ADDRESS
033224  012702  033342                  MOV     #5$,R2                  ;POINTER TO CORRECT DATA
033230  004737  034676                  JSR     PC,MEMLD                ;LOAD 8 WORDS OF MAIN MEMORY
033234  035022                          MEMDAT                          ;POINTER TO DATA
033236  004737  034732                  JSR     PC,SPLD                 ;LOAD 8 WORDS OF SP
033242  035032                          SPDAT                           ;POINTER TO DATA
033244  004737  035010          1$:     JSR     PC,SETC                 ;SET C BIT!
033250  042737  000017  033264          BIC     #17,2$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
033256  050037  033264                  BIS     R0,2$                   ;ADD ADDRESS TO INSTRUCTION
033262  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
033264  010000                  2$:     010000                          ;LOAD MAR
033266  042737  000017  033302          BIC     #17,3$                  ;CLEAR ADDRESS OF INSTRUCTION
033274  050037  033302                  BIS     R0,3$                   ;ADD ADDRESS TO INSTRUCTION
033300  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033302  040560                  3$:     040400!<7*20>                   ;BR    DEC A
033304  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
033306  061224                          61224                           ;MOVE BR TO PORT4
033310  111205                          MOVB    (R2),R5                 ;PUT ''EXPECTED'' IN R5
033312  116104  000004                  MOVB    4(R1),R4                ;PUT ''FOUND'' IN R4
033316  120504                          CMPB    R5,R4                   ;DATA CORRECT?
033320  001401                          BEQ     4$                      ;BR IF YES
033322  104015                          HLT     15                      ;ALU ERROR
033324  104401                  4$:     SCOP1                           ;SW09=1?
033326  005202                          INC     R2                      ;NEXT DATA
033330  005200                          INC     R0                      ;NEXT ADDRESS
033332  022700  000010                  CMP     #10,R0                  ;DONE YET?
033336  001342                          BNE     1$                      ;BR IF NO
033340  104400                          SCOPE                           ;SCOPE THIS TEST
033342     377     377     376  5$:     .BYTE   -1,-1,376,376,124,124,251,251
033345     376     124     124
033350     251     251
                                .EVEN


                                                        ;*************************** TEST 135 ***************************
                                                        ;*TEST OF PROGRAM CLOCK BIT
                                                        ;*DO A MASTER CLEAR, WRITE THE PROGRAM CLOCK BIT TO A ONE,
                                                        ;*VERIFY THAT IT CLEARS, AND THEN SETS SOME TIME LATER
```

```
                                          ;;*****************************************************************
                                          ;  TEST 135
                                          ;---------------
033352  012737  000135  001226  TST135:   MOV     #135,TSTNO
033360  012737  033544  001216            MOV     #TST136,NEXT
                                                                   ;R1 CONTAINS BASE DMC11 ADDRESS
033366  104412                            MSTCLR                   ;MASTER CLEAR DMC11
033370  005037  001416                    CLR     TEMP             ;PREPARE FOR
033374  005037  001246                    CLR     TEMP1            ;DELAY
033400  012702  000011                    MOV     #11,R2           ;SAVE FOR TYPEOUT
033404  012761  000020  000004  1$:       MOV     #20,4(R1)        ;LOAD PORT 4
033412  152761  000002  000001            BISB    #BIT1,1(R1)      ;SET ROMI
033420  012761  121111  000006            MOV     #121111,6(R1)    ;SEL6   INSTRUCTION
033426  152761  000003  000001            BISB    #BIT1!BIT0,1(R1);SET CLOCK BIT
033434  012761  121224  000006            MOV     #121224,6(R1)    ;LOAD NEXT INSTRUCTION
033442  152761  000003  000001            BISB    #BIT1.BIT0,1(R1);READ CLOCK BIT
033450  142761  000003  000001            BICB    #BIT1!BIT0,1(R1);CLEAR MAINT BITS
033456  016104  000004                    MOV     4(R1),R4         ;PUT ''FOUND'' IN R4
033462  005005                            CLR     R5               ;PUT ''EXPECTED'' IN R5
033464  120504                            CMPB    R5,R4            ;IS PGM CLOCK CLEAR?
033466  001401                            BEQ     2$
033470  104016                            HLT     16               ;ERROR, PGM CLOCK IS NOT CLEAR
033472                          2$:
033472  104414                            ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
033474  121224                            121224                   ;PORT4 LU11
033476  122761  000020  000004            CMPB    #20,4(R1)        ;IS PGM CLOCK SET?
033504  001416                            BEQ     3$               ;BR IF YES
033506  062737  000001  001416            ADD     #1,TEMP          ;INCREMENT DELAY
033514  005537  001246                    ADC     TEMP1            ;INCREMENT DELAY
033520  022737  000006  001246            CMP     #6,TEMP1         ;IS DELAY DONE
033526  001361                            BNE     2$               ;BR IF NO
033530  012705  000020                    MOV     #20,R5           ;PUT ''EXPECTED'' IN R5
033534  016104  000004                    MOV     4(R1),R4         ;PUT ''FOUND'' IN R4
033540  104016                            HLT     16               ;ERROR PGM CLOCK NOT SET
033542  104400                  3$:       SCOPE                    ;SCOPE THIS TEST
                                          ;*************************** TEST 136 ***************************
                                          ;*FORCE POWER FAIL TEST
                                          ;*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
                                          ;*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
                                          ;*BLOCKED FROM GETTING TO THE DMC DURING THE POWER FAIL
                                          ;*THIS TEST MAY HANG ON SOME PROCESSORS IF AN M9301 IS PRESENT.
                                          ;*TO AVOID HANGING SW02 (POWER ON REBOOT ENABLE) ON THE M9301
                                          ;*MUST BE IN THE OFF POSITION. THIS TEST WILL ALSO FAIL IF THE
                                          ;*CPU POWER FAIL VECTOR IS SET TO ANY LOCATION OTHER THAN 24.
                                          ;*IF THIS TEST HANGS OR FAILS DUE TO EITHER REASON ABOVE THE
                                          ;*FOLLOWING PATCH MAY BE INSTALLED TO SKIP THIS TEST:
                                          ;*
                                          ;*      LOC 33362      WAS 33544      SB 33736
                                          ;;*****************************************************************
                                          ;  TEST 136
                                          ;---------------
033544  012737  000136  001226  TST136:   MOV     #136,TSTNO
```

```
033552 012737 033736 001216          MOV     #TST137,NEXT
                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
033560 104412                        MSTCLR              ;MASTER CLEAR DMC11
033562 005037 001416                 CLR     TEMP        ;PREPARE FOR DELAY
033566 013746 000024                 MOV     @#24,-(SP)  ;STORE POWER FAIL ADDRESS
033572 012737 033636 000024          MOV     #1$,@#24    ;SET UP FOR FORCE POWER FAIL
033600 012761 000002 000004          MOV     #2,4(R1)    ;LOAD PORT4
033606 012711 001000                 MOV     #BIT9,(R1)  ;SET ROMI
033612 012761 121111 000006          MOV     #121111,6(R1)  ;LOAD INSTRUCTION
033620 012711 001400                 MOV     #BIT9.BIT8,(R1) ;CLOCK INSTRUCTION
033624 005237 001416         5$:     INC     TEMP        ;WAIT FOR POWER FAIL
033630 001375                        BNE     5$          ;BR IF DELAY NOT DONE
033632 104017                        HLT     17          ;ERROR, NO POWER FAIL
033634 000426                        BR      4$
033636 012737 033654 000024  1$:     MOV     #3$,@#24    ;POWER UP ADDRESS
033644 010637 033652                 MOV     SP,2$       ;STORE STACK
033650 000000                        HALT                ;WAIT FOR POWER UP SEQUENCE
033652 000000                2$:     0
033654 013706 033652         3$:     MOV     2$,SP       ;RESTORE STACK
033660 022626                        POP2SP              ;POP STACK TWICE
033662 012637 000024                 MOV     (SP)+,@#24  ;RESTORE TRUE POWER FAIL ADDRESS
033666 022737 005336 000024          CMP     #.PFAIL,@#24 ;IS IT CORRECT?
033674 001406                        BEQ     4$          ;BR IF YES
033676 104017                        HLT     17          ;ERROR, STACK IS INCORRECT
033700 012737 005336 000024          MOV     #.PFAIL,@#24 ;RESTORE TRUE POWER FAIL ADDRESS
033706 012706 001200                 MOV     #STACK,SP   ;RESTORE STACK
033712 012711 003000         4$:     MOV     #BIT9.BIT10,(R1);SEL6 = MAINT IR
033716 012705 121111                 MOV     #121111,R5  ;R5 = EXPECTED
033722 016104 000004                 MOV     4(R1),R4    ;R4 = FOUND
033726 020504                        CMP     R5,R4       ;MAINT IR SHOULD   12111
033730 001401                        BEQ     .+4         ;BR IF OK
033732 104025                        HLT     25          ;IF - 0 THEN BUS INIT WAS
                                                         ;NOT BLOCKED FROM CLEARING
                                                         ;THE DMC-11
033734 104400                        SCOPE               ;SCOPE THIS TEST


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

                                     ;  TEST 137
                                     ;--------------
033736 012737 000137 001226  TST137: MOV     #137,TSTNO
033744 012737 003364 001216          MOV     #.EOP,NEXT
                                                         ;R1 CONTAINS BASE DMC11 ADDRESS
033752 104412                        MSTCLR              ;MASTER CLEAR DMC11
033754 005002                        CLR     R2          ;R2 IS INDEX REGISTER
033756 042737 000017 034002  1$:     BIC     #17,2$      ;CLEAR ADDRESS FIELD
033764 156237 034576 034002          BISB    30$(R2),2$  ;ADD IBUS* REG ADDRESS TO INSTRUCTION
033772 116261 034604 000004          MOVB    31$(R2),4(R1) ;LOAD PORT4
```

```
034000 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
034002 121100            2$:    121100             ;WRITE IBUS* REGISTER
034004 005202                   INC    R2          ;INC INDEX REGISTER
034006 022702 000005            CMP    #5,R2       ;DONE YET?
034012 001361                   BNE    1$          ;BR IF NO
034014 005002                   CLR    R2          ;R2 IS IBUS REGISTER ADDRESS
034016 042737 000017 034062 3$: BIC    #17,4$      ;CLEAR ADDRESS FIELD OF INSTRUCTIONS
034024 042737 000017 034074     BIC    #17,5$
034032 042737 000017 034104     BIC    #17,6$
034040 050237 034062            BIS    R2,4$       ;ADD IBUS REG ADDRESS TO INSTRUCTION
034044 050237 034074            BIS    R2,5$
034050 050237 034104            BIS    R2,6$
034054 105061 000004            CLRB   4(R1)       ;CLEAR PORT4
034060 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034062 122100            4$:    122100             ;WRITE 0 TO IBUS REG
034064 112761 000377 000004     MOVB   #377,4(R1)  ;LOAD PORT4
034072 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034074 122100            5$:    122100             ;WRITE ALL ONES TO IBUS REG
034076 110261 000004            MOVB   R2,4(R1)    ;LOAD PORT4
034102 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034104 122100            6$:    122100             ;WRITE ITS OWN ADDRESS TO IBUS REG
034106 005202                   INC    R2          ;NEXT ADDRESS
034110 022702 000010            CMP    #10,R2      ;DONE YET?
034114 001340                   BNE    3$          ;BR IF NO
034116 005002                   CLR    R2          ;START AT SP ADDRESS 0
034120 042737 000017 034164 7$: BIC    #17,8$      ;CLEAR ADDRESS FIELD
034126 042737 000017 034176     BIC    #17,9$
034134 042737 000017 034206     BIC    #17,10$
034142 050237 034164            BIS    R2,8$       ;ADD ADDRESS TO INSTRUCTION
034146 050237 034176            BIS    R2,9$
034152 050237 034206            BIS    R2,10$
034156 105061 000004            CLRB   4(R1)       ;CLEAR PORT4
034162 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034164 123100            8$:    123100             ;WRITE ZERO TO SP
034166 112761 000377 000004     MOVB   #377,4(R1)  ;LOAD PORT4
034174 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034176 123100            9$:    123100             ;WRITE ALL ONES TO SP
034200 110261 000004            MOVB   R2,4(R1)    ;LOAD PORT4
034204 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034206 123100            10$:   123100             ;WRITE SP ADDRESS TO ITSELF
034210 005202                   INC    R2          ;NEXT SP ADDRESS
034212 022702 000020            CMP    #20,R2      ;DONE YET?
034216 001340                   BNE    7$          ;BR IF NO
034220 005002                   CLR    R2          ;R2 = MAIN MEM ADDRESS
034222 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034224 010000                   010000             ;MAR   0
034226 105061 000004     11$:   CLRB   4(R1)       ;CLEAR PORT4
034232 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034234 122500                   122500             ;WRITE ZEROS TO MEM
034236 112761 000377 000004     MOVB   #377,4(R1)  ;LOAD PORT4
034244 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034246 122500                   122500             ;WRITE ONES TO MEM
034250 110261 000004            MOVB   R2,4(R1)    ;LOAD PORT4
034254 104414                   ROMCLK             ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034256 136500                   136500             ;WRITE TO MEM IT OWN ADDRESS
034260 005202                   INC    R2          ;NEXT MEM ADDRESS
```

```
034262  022702  000400                  CMP     #400,R2         ;DONE YET?
034266  001357                          BNE     11$             ;BR IF NO

                                        ;NOW GO BACK AND READ EVERYTHING

034270  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034272  010000                          010000                  ;MAR 0
034274  032737  100000  001366          BIT     #BIT15,STAT1    ;DMC?
034302  001402                          BEQ     .+6             ;BR IF YES
034304  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034306  004000                          4000                    ;MAR HI   0 (KMC ONLY)
034310  005000                          CLR     R0              ;R0 IS INDEX REGISTER
034312  042737  000360  034350  12$:    BIC     #360,13$        ;CLEAR ADDRESS FIELD
034320  116002  034576                  MOVB    30$(R0),R2      ;R2 = IBUS* ADDRESS
034324  010203                          MOV     R2,R3           ;PUT IBUS* ADDRESS IN R3
034326  006303                          ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
034330  006303                          ASL     R3
034332  006303                          ASL     R3
034334  006303                          ASL     R3
034336  050337  034350                  BIS     R3,13$          ;ADD ADDRESS TO INSTRUCTION
034342  116005  034604                  MOVB    31$(R0),R5      ;R5 - 'EXPECTED''
034346  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
034350  121004                  13$:    121004                  ;PORT4   IBUS* REGISTER
034352  016104  000004                  MOV     4(R1),R4        ;R4 = 'FOUND''
034356  120504                          CMPB    R5,R4           ;IBUS* CONTENTS OK?
034360  001401                          BEQ     .+4             ;BR IF YES
034362  104004                          HLT     4               ;IBUS* DATA ERROR
034364  005200                          INC     R0              ;INC COUNTER
034366  022700  000005                  CMP     #5,R0           ;DONE YET?
034372  001347                          BNE     12$             ;BR IF NO
034374  005002                          CLR     R2              ;R2 - IBUS REG ADDRESS
034376  042737  000360  034426  14$:    BIC     #360,15$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
034404  010203                          MOV     R2,R3           ;R3 = IBUS ADDRESS
034406  006303                          ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
034410  006303                          ASL     R3
034412  006303                          ASL     R3
034414  006303                          ASL     R3
034416  050337  034426                  BIS     R3,15$          ;ADD ADDRESS TO INSTUCTION
034422  010205                          MOV     R2,R5           ;R5 - 'EXPECTED''
034424  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034426  021004                  15$:    021004                  ;PORT4   IBUS REG
034430  016104  000004                  MOV     4(R1),R4        ;R4 = 'FOUND''
034434  120504                          CMPB    R5,R4           ;IBUS CONTENTS OK?
034436  001401                          BEQ     .+4             ;BR IF YES
034440  104005                          HLT     5               ;IBUS DATA ERROR
034442  005202                          INC     R2              ;NEXT IBUS REGISTER
034444  022702  000010                  CMP     #10,R2          ;DONE YET?
034450  001352                          BNE     14$             ;BR IF NO
034452  005002                          CLR     R2              ;R2 = SP ADDRESS
034454  042737  000017  034470  16$:    BIC     #17,17$ ;CLEAR ADRESS FIELD OF INSTRUCTION
034462  050237  034470                  BIS     R2,17$          ;ADD ADDRESS TO INSTRUCTION
034466  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034470  040600                  17$:    040600                  ;BR   SP
034472  010205                          MOV     R2,R5           ;R5 = 'EXPECTED''
034474  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034476  061224                          061224                  ;PORT4 _ BR
```

```
034500 016104 000004                 MOV     4(R1),R4          ;R4 - 'FOUND'
034504 120504                        CMPB    R5,R4             ;SP CONTENTS OK?
034506 001401                        BEQ     .+4               ;BR IF YES
034510 104007                        HLT     7                 ;SP DATA ERROR
034512 005202                        INC     R2                ;NEXT SP LOCATION
034514 022702 000020                 CMP     #20,R2            ;DONE YET?
034520 001355                        BNE     16$               ;BR IF NO
034522 005002                        CLR     R2                ;R2 - MEMORY ADDRESS
034524 104414                        ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034526 010000                        010000                    ;MAR _ 0
034530 032737 100000 001366          BIT     #BIT15,STAT1      ;DMC?
034536 001402                        BEQ     .+6               ;BR IF YES
034540 104414                        ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034542 004000                        4000                      ;MAR HI _ 0 (KMC ONLY)
034544 010205                18$:    MOV     R2,R5             ;R5 - 'EXPECTED'
034546 104414                        ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034550 055224                        055224                    ;PORT4 _ MAIN MEM
034552 016104 000004                 MOV     4(R1),R4          ;R4 = 'FOUND'
034556 120504                        CMPB    R5,R4             ;MAIN MEM CONTENTS OK?
034560 001401                        BEQ     .+4               ;BR IF YES
034562 104013                        HLT     13                ;MAIN MEM DATA ERROR
034564 005202                        INC     R2                ;NEXT MEM ADDRESS
034566 022702 000400                 CMP     #400,R2           ;DONE YET?
034572 001364                        BNE     18$               ;BR IF NO
034574 104400                        SCOPE                     ;SCOPE THIS TEST
034576    000    002    003  30$:    .BYTE   0,2,3,5,10
034601    005    010
       034604                .EVEN
034604    001    003    004  31$:    .BYTE   1,3,4,6,10
034607    006    010
       034612                .EVEN


                             ;SUBROUTINES
                             ;------------

034612                       SETVEC:
                                     ;THIS SUBROUTINE LOADS THE VECTORS AND VECTOR LEVELS

034612 012577 144556                 MOV     (R5)+,@DMRVEC     ;LOAD BASE VECTOR
034616 012577 144556                 MOV     (R5)+,@DMTVEC     ;LOAD VECTOR + 2
034622 112577 144550                 MOVB    (R5)+,@DMRLVL     ;LOAD VECTOR + 4
034626 112577 144550                 MOVB    (R5)+,@DMTLVL     ;LOAD VECTOR + 6
034632 000205                        RTS     R5                ;RETURN


034634                       NPRSET:
                                     ;THIS SUBROUTINE LOADS IBUS REGISTERS 0-7
                                     ;WITH NPR INFORMATION (INBA, OUTBA, OUT DATA)

034634 010246                        MOV     R2,-(SP)          ;SAVE R2
034636 005002                        CLR     R2                ;START AT IBUS REG 0
034640 112561 000004          1$:    MOVB    (R5)+,4(R1)       ;LOAD PORT4
034644 042737 000017 034[illegible]          BIC     #17,2$            ;CLEAR ADDRESS FIELD OF INSTRUCTION
034652 050237 034660                 BIS     R2,2$             ;ADD ADDRESS TO INSTRUCTION
034656 104414                        ROMCLK                    ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
```

```
034660  122100            2$:     122100                 ;MOVE PORT4 TO IBUS REG
034662  005202                    INC     R2             ;NEXT ADDRESS
034664  022702  000010            CMP     #10,R2         ;ALL DONE?
034670  001363                    BNE     1$             ;BR IF NO
034672  012602                    MOV     (SP)+,R2       ;RESTORE R2
034674  000205                    RTS     R5             ;RETURN


034676                    MEMLD:
                                  ;THIS SUBROUTINE LOADS THE FIRST 8 LOCATIONS OF MAIN
                                  ;MEMORY WITH THIS DATA: 0,-1,,0,-1,125,252,125,252

034676  013605                    MOV     @(SP)+,R5      ;PUT POINTER TO DATA IN R5
034700  062746  000002            ADD     #2,-(SP)       ;ADJUST STACK
034704  012704  000010            MOV     #10,R4         ;DO 8 LOADS
034710  104414                    ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034712  010000                    010000                 ;MAR < 0
034714  112577  144444    1$:     MOVB    (R5)+,@DMP04   ;LOAD PORT4
034720  104414                    ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
034722  136500                    136500                 ;MOV DATA TO MEM, AUTO INC MAR
034724  005304                    DEC     R4             ;DECREMENT COUNT
034726  001372                    BNE     1$             ;BR IF NOT DONE
034730  000207                    RTS     PC             ;RETURN


034732                    SPLD:
                                  ;THIS SUBROUTINE LOADS THE FIRST 8 SCRATCH PAD
                                  ;LOCATIONS WITH: 0,0,-1,-1,125,125,252,252

034732  013605                    MOV     @(SP)+,R5      ;PUT POINTER TO DATA IN R5
034734  062746  000002            ADD     #2,-(SP)       ;ADJUST STACK
034740  005004                    CLR     R4             ;START AT SP ADDRESS 0
034742  112577  144444    1$:     MOVB    (R5)+,@DMP04   ;LOAD PORT4 WITH DATA
034746  042737  000017  034762    BIC     #17,2$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
034754  050437  034762            BIS     R4,2$          ;ADD ADDRESS TO INSTRUCTION
034760  104414                    ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
034762  123100            2$:     123100                 ;MOVE DATA TO SP
034764  005204                    INC     R4             ;INCREMENT COUNT
034766  022704  000010            CMP     #10,R4         ;DONE YET?
034772  001363                    BNE     1$             ;BR IF NO
034774  000207                    RTS     PC             ;RETURN


034776                    [illegible]:
                                  ;THIS SUBROUTINE CLEARS THE MICRO PROCESSOR C BIT

034776  104414                    ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
035000  010000                    010000                 ;MAR 0
035002  104414                    ROMCLK                 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
035004  040400                    040400.<0*20>          ;CLEAR C BIT
035006  000207                    RTS     PC             ;RETURN


[illegible]               [illegible]:
                                  ;THIS SUBROUTINE [illegible] THE MICRO PROCESSOR [illegible] BIT
```

```
035010  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC 5304
035012  010003                          010003                          ;MAR 3
035014  104414                          ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
035016  040403                          040403 <0*20>                   ;SET C BIT
035020  000207                          RTS     PC                      ;RETURN


035022     000     377     000  MEMDAT: .BYTE   0,-1,0,-1,125,252,125,252
035025     377     125     252
035030     125     252
035032     000     000     377  SPDAT:  .BYTE   0,0,-1,-1,125,125,252,252
035035     377     125     125
035040     252     252
                                .EVEN
035042  052777  044516  052502  EM1:    .ASCIZ  <377>/UNIBUS REGISTER ADDRESSING TIME-OUT/
035107     377  047125  041111  EM2:    .ASCIZ  <377>^UNIBUS REGISTER WRITE/READ TEST^
035150  046777  041511  047522  EM3:    .ASCIZ  <377>/MICRO PROCESSOR TEST/
035176  046777  041511  047522  EM4:    .ASCIZ  <377>^MICRO PROCESSOR WRITE/READ TEST^
035237     377  051102  051040  EM5:    .ASCIZ  <377>/BR REGISTER TEST/
035261     377  041523  040522  EM6:    .ASCIZ  <377>/SCRATCH PAD TEST/
035303     377  042504  044526  EM7:    .ASCIZ  <377>/DEVICE FAILED TO INTERRUPT/
035337     377  042504  044526  EM10:   .ASCIZ  <377>/DEVICE INTERRUPTED TO WRONG VECTOR/
035403     377  050116  020122  EM11:   .ASCIZ  <377>/NPR TEST/
035415     377  040515  047111  EM12:   .ASCIZ  <377>/MAIN MEMORY TEST/
035437     377  040515  020122  EM13:   .ASCIZ  <377>/MAR TEST/
035451     377  046101  020125  EM14:   .ASCIZ  <377>/ALU TEST/
035463     377  051120  043517  EM15:   .ASCIZ  <377>/PROGRAM CLOCK TEST/
035507     377  047506  041522  EM16:   .ASCIZ  <377>/FORCE POWER FAIL ERROR/
035537     377  047125  054105  EM17:   .ASCIZ  <377>/UNEXPECTED INTERRUPT/
035565     377  046504  030503  EM20:   .ASCIZ  <377>/DMC11 CONFIGURATION ERROR/
035620  046777  044501  052116  EM21:   .ASCIZ  <377>/MAINTENANCE INSTRUCTION REGISTER TEST/
035667     377  047520  042527  EM22:   .ASCIZ  <377>/POWER FAIL INITIALIZE FAILURE/

035726  051377  043505  051511  DH1:    .ASCIZ  <377>/REGISTER             TRAPPED FROM/
035767     377  054105  042520  DH2:    .ASCIZ  <377>/EXPECTED  FOUND       REGISTER/
036025     377  054105  042520  DH3:    .ASCIZ  <377>/EXPECTED  FOUND/
036046  042777  050130  041505  DH4:    .ASCIZ  <377>/EXPECTED  FOUND    IBUS* REGISTER/
036110  042777  050130  041505  DH5:    .ASCIZ  <377>/EXPECTED  FOUND    IBUS REGISTER/
036151     377  054105  042520  DH6:    .ASCIZ  <377>/EXPECTED  FOUND       ADDRESS/
                                .EVEN

036206  000002                  DT1:    2
036210     006     015                  .BYTE   6,15
036212  001262                          SAVR1
036214     006     002                  .BYTE   6,2
036216  001264                          SAVR2
036220  000003                  DT2:    3
036222     006     004                  .BYTE   6,4
036224  001272                          SAVR5
036226     006     004                  .BYTE   6,4
036230  001270                          SAVR4
036232     006     002                  .BYTE   6,2
036234  001262                          SAVR1
036236  000002                  DT3:    2
036240     006     004                  .BYTE   6,4
036242  001272                          SAVR5
```

```
036244     006     002           .BYTE   6,2
036246  001270                   SAVR4
036250  000003           DT4:    3
036252     003     007           .BYTE   3,7
036254  001272                   SAVR5
036256     003     011           .BYTE   3,11
036260  001270                   SAVR4
036262     002     002           .BYTE   2,2
036264  001264                   SAVR2
036266  000003           DT5:    3
036270     003     007           .BYTE   3,7
036272  001272                   SAVR5
036274     003     007           .BYTE   3,7
036276  001270                   SAVR4
036300     006     002           .BYTE   6,2
036302  001264                   SAVR2
036304  000002           DT6:    2
036306     003     007           .BYTE   3,7
036310  001272                   SAVR5
036312     003     002           .BYTE   3,2
036314  001270                   SAVR4
036316  000002           DT7:    2
036320     006     004           .BYTE   6,4
036322  001262                   SAVR1
036324     006     002           .BYTE   6,2
036326  001404                   DMCSR
036330                   .ERRTAB:
036330  000000                   0
036332  000000                   0
036334  000000                   0
036336  035042                   EM1
036340  035726                   DH1     ;HLT    1
036342  036206                   DT1
036344  035107                   EM2
036346  035767                   DH2     ;HLT    2
036350  036220                   DT2
036352  035150                   EM3
036354  036025                   DH3     ;HLT    3
036356  036236                   DT3
036360  035176                   EM4
036362  036046                   DH4     ;HLT    4
036364  036250                   DT4
036366  035176                   EM4
036370  036110                   DH5     ;HLT    5
036372  036250                   DT4
036374  035237                   EM5
036376  036025                   DH3     ;HLT    6
036400  036304                   DT6
036402  035261                   EM6
036404  036151                   DH6     ;HLT    7
036406  036266                   DT5
036410  035303                   EM7
036412  000000                   0       ;HLT    10
036414  000000                   0
036416  035337                   EM10
036420  000000                   0       ;HLT    [illegible]
```

```
036422  000000          0
036424  035403          EM11
036426  036025          DH3     ;HLT    12
036430  036236          DT3
036432  035415          EM12
036434  036151          DH6     ;HLT    13
036436  036266          DT5
036440  035437          EM13
036442  036151          DH6     ;HLT    14
036444  036266          DT5
036446  035451          EM14
036450  036025          DH3     ;HLT    15
036452  036304          DT6
036454  035463          EM15
036456  036046          DH4     ;HLT    16
036460  036250          DT4
036462  035507          EM16
036464  000000          0       ;HLT    17
036466  000000          0
036470  035537          EM17
036472  000000          0       ;HLT    20
036474  000000          0
036476  035403          EM11
036500  036151          DH6     ;HLT    21
036502  036266          DT5
036504  035565          EM20
036506  036025          DH3     ;HLT    22
036510  036316          DT7
036512  035620          EM21
036514  036025          DH3     ;HLT    23
036516  036236          DT3
036520  035565          EM20
036522  000000          0       ;HLT    24
036524  000000          0
036526  035667          EM22
036530  036025          DH3     ;HLT    25
036532  036236          DT3

036534                  ORMAX:
        000001          .END
```

```
ADRCNT= 004373     1466*   1502*   1511#
AUDONE  003024     1156    1195    1200    1246#
AUSTRT  002446     1155#   1250
AUTO.S  010512     1113    1955#
BINWRD  004714     1552*   1555*   1556    1593#
BIT0    000001      682#   1742    1743    2354    2362    2624    2632    4451    6296    6298    6299
BIT1  - 000002      681#   1118    1736    1742    1743    2036    2119    2132    2384    2392    2654    2662    6294
                   6296    6298    6299
BIT10   002000      672#   2100    2111    2962    2981    3003    3022    4311    4347    6364
BIT11 = 004000      671#   2564    2572
BIT12 = 010000      670#   2051    2130    2594    2602    2774    2782
BIT13 - 020000      669#   2054    2103    2134    2804    2812
BIT14   040000      668#   1368    2065    2067    2134    2145    3021
BIT15 - 100000      667#   1072    1159    1162    1183    1186    2038    2106    4641    4654    6457    6516
BIT2  = 000004      680#   1118    1299    1743    2414    2422    2684    2692
BIT3  - 000010      679#   2136    2143
BIT4    000020      678#   1726    1749    1751
BIT5    000040      677#   2248    2444    2452
BIT6    000100      676#   1731    1732    2138    2474    2482    2714    2722
BIT7  - 000200      675#   1732    1852    2248    2504    2512    2744    2752
BIT8    000400      674#   2110    2125    2127    2140    2142    2148    2151    2209    3048    4312    4348    6347
BIT9    001000      673#   2108    2110    2122    2148    2151    2207    2209    2534    2542    2962    2981    3003
                   3022    3046    3048    4311    4347    6345    6347    6364
BM      007054     1774#   2079
BRLVL   012252     2204    2214    2222    2234#
BRW     003730     1305    1391#
BRX     003732     1306    1392#
CHRCNT  004712     1550*   1553    1557    1573*   1591#   1592
CKSWR   007606     1099    1366    1398    1613    1799#
CKSWR1  007666     1812#   1824
CKSWR2  007700     1815#
CKSWR3  007704     1817#
CKSWR4  007710     1818#   1826    1833
CKSWR5  010014     1800    1807    1842#
CLKX    001242      758#
CLRC    034776     4731    4780    4829    4878    4927    4976    5025    5074    5123    5172    5221    5270    5319
                   5368    5417    5466    6605#
CNERR   007277     1213    1774#
CNT.MA  001702      783     951#   1071    1073    1075    1880
CNVRT   104411      820#   1210    1325    1327    1329    1331    1647    1649    1709    1815
CONERR  007223     1208    1774#
CONN    007114     1774#   2056
CONTAB  002776     1217    1230#
CONVRT  104410      818#   1130    1216    1663    1978
CORMAX  036534     4447    4479    6632#
CRAM    006606     1774#   2017
CREAM   001320      782#   1070*   1876*   1877    1879*   1883
CSR     006510     1774#   1982
CSRMAP  010514     1957#
CYCLE   010060     1307    1348    1349    1867#
DATABP  005216     1636*   1639    1661    1664#
DATACL  104415      828#
DATAHD  005204     1635*   1657    1660#
DELAY   104413      824#
DEVADR  004370     1464*   1499    1509#
DEVTAB  003010     1170    1235#
```

```
DH1      035726      6632#
DH2      035767      6632#
DH3      036025      6632#
DH4      036046      6632#
DH5      036110      6632#
DH6      036151      6632#
DISPLA   001200       729#   1085*   1091*   1322*
DISPRE   000174       715#   1091
DMACTV   001306       776#   1108    1263*   1264    1867    1881    1963*   2164*   2170*   2171*   2175    2200
DMCM     007320      1221    1774#
DMCR00   001500       866#
DMCR01   001510       871#
DMCR02   001520       876#
DMCR03   001530       881#
DMCR04   001540       886#
DMCR05   001550       891#
DMCR06   001560       896#
DMCR07   001570       901#
DMCR10   001600       906#
DMCR11   001610       911#
DMCR12   001620       916#
DMCR13   001630       921#
DMCR14   001640       926#
DMCR15   001650       931#
DMCR16   001660       936#
DMCR17   001670       941#
DMCSR    001404       849#   1155*   1189    1194*   1199*   1234    1352    1389    1681    1704    1750    1885*   1894
                     1943    2264    2325    2353    2383    2413    2443    2473    2503    2533    2563    2593    2933
                     6632
DMCSRH   001406       850#   1731*   1732*   1736*   1742*   1743*   1749*   1751*   1894*   1895*   1896
DMCTL    001410       851#   1896*   1897*   1898    2623    2653    2683    2713    2743    2773    2803
DMNUM    001310       777#   1066    1338    1961*   1976*   2155*   2156    2165    2167
DMP04    001412       852#   1720*   1726    1763    1768    1898*   1899*   1900    2833    4299    4300    4338    4339
                     6579*   6594*
DMP06    001414       853#   1737*   1900*   1901*   2876
DMRLVI   001376       846#   1903*   1904*   1905    6547*
DMRVEC   001374       845#   1355    1886*   1887*   1903    6545*
DMS100   001502       867#
DMS101   001512       872#
DMS102   001522       877#
DMS103   001532       882#
DMS104   001542       887#
DMS105   001552       892#
DMS106   001562       897#
DMS107   001572       902#
DMS110   001602       907#
DMS111   001612       912#
DMS112   001622       917#
DMS113   001632       922#
DMS114   001642       927#
DMS115   001652       932#
DMS116   001662       937#
DMS117   001672       942#
DMS200   001504       868#
DMS201   001514       873#
DMS202   001524       878#
```

```
DMS203  001534        883#
DMS204  001544        888#
DMS205  001554        893#
DMS206  001564        898#
DMS207  001574        903#
DMS210  001604        908#
DMS211  001614        913#
DMS212  001624        918#
DMS213  001634        923#
DMS214  001644        928#
DMS215  001654        933#
DMS216  001664        938#
DMS217  001674        943#
DMS300  001506        869#
DMS301  001516        874#
DMS302  001526        879#
DMS303  001536        884#
DMS304  001546        889#
DMS305  001556        894#
DMS306  001566        899#
DMS307  001576        904#
DMS310  001606        909#
DMS311  001616        914#
DMS312  001626        919#
DMS313  001636        924#
DMS314  001646        929#
DMS315  001656        934#
DMS316  001666        939#
DMS317  001676        944#
DMTLVL  001402        848#    1907*   1908*   6548*
DMTVEC  001400        847#    1905*   1906*   1907    6546*
DM.END  001700        946#    1959
DM.MAP  001500        782      865#   1070    1123    1133    1249    1877    1879    1957    1962    2192
DONE    003734       1371     1373*   1393#   1803*
DT1     036206       6632#
DT2     036220       6632#
DT3     036236       6632#
DT4     036250       6632#
DT5     036266       6632#
DT6     036304       6632#
DT7     036316       6632#
EM1     035042       6632#
EM10    035337       6632#
EM11    035403       6632#
EM12    035415       6632#
EM13    035437       6632#
EM14    035451       6632#
EM15    035463       6632#
EM16    035507       6632#
EM17    035537       6632#
EM2     035107       6632#
EM20    035565       6632#
EM21    035620       6632#
EM22    035667       6632#
EM3     035150       6632#
EM4     035176       6632#
```

```
EM5      035237        6632#
EM6      035261        6632#
EM7      035303        6632#
ERCT00   001704         953#
ERCT01   001710         956#
ERCT02   001714         959#
ERCT03   001720         962#
ERCT04   001724         965#
ERCT05   001730         968#
ERCT06   001734         971#
ERCT07   001740         974#
ERCT10   001744         977#
ERCT11   001750         980#
ERCT12   001754         983#
ERCT13   001760         986#
ERCT14   001764         989#
ERCT15   001770         992#
ERCT16   001774         995#
ERCT17   002000         998#
ERR      002700        1179     1205#    1209
ERRCNT   001232         750#    1334     1361     1674*    1892*
ERRFLG   001325         789#    1068*    1320*    1382*    1624*    1637     1651*    1710*
ERRMSG   005172        1634*    1652     1655#
ERRPC    002770        1211     1227#
ERTABO   005322        1649     1684#
EXIT   = 000205         683#
EXITER   005252        1669     1674#
FLOAT    002536        1172#    1178
FY       002566        1181#    1192     1196     1202
HALTS    005222        1620     1666#
HILIM    004366        1463*    1490     1508#
ICOUNT   001222         746#    1380     1385*    4441*
INBUF    007502        1433     1469     1789#
INCHAR   010020        1818     1846#
INIFLG   001324         788#    1094     1114     1121*
INSTER=  104404         810#    1484
INSTR    104403         808#    1916     1968     1981     1990     2069     2078
INSTR2   004166        1440     1452#
INTTY    012266        2001     2018     2028     2041     2057     2242#
KMCM     007330        1224     1774#
LIMITS   004314        1479     1490#
LINE     007016        1774#    2070
LOBITS   004372        1465*    1494     1510#    1511
LOCK     001220         745#    1384*    1401     1403     1643     2262*    2309*    2327*    2351*    2361*    2381*    2391*    2411*
                       2421*    2441*    2451*    2471*    2481*    2501*    2511*    2531*    2541*    2561*    2571*    2591*    2601*
                       2621*    2631*    2651*    2661*    2681*    2691*    2711*    2721*    2741*    2751*    2771*    2781*    2801*
                       2811*    2831*    2846*    2874*    2889*    2917*    2935*    2959*    2970*    2978*    3000*    3011*    3019*
                       3068*    3090*    3124*    3146*    3180*    3200*    3232*    3252*    3285*    3307*    3343*    3367*    3403*
                       3423*    3455*    3475*    3507*    3527*    3559*    3579*    3611*    3631*    3663*    3683*    3715*    3735*
                       3767*    3787*    3819*    3848*    3880*    3899*    3931*    3956*    3997*    4024*    4497*    4537*    4580*
                       4604*    4681*    4722*    4771*    4820*    4869*    4918*    4967*    5016*    5065*    5114*    5163*    5212*
                       5261*    5310*    5359*    5408*    5457*    5506*    5555*    5604*    5653*    5702*    5751*    5800*    5849*
                       5898*    5947*    5996*    6045*    6094*    6143*    6192*    6241*
LOKFLG   001326         790#
LOLIM    004364        1462*    1492     1507#
LPCNT    001224         747#    1379*    1380     1383*
```

```
LSTERR  001234       751#    1077*   1319*   1621    1623*   1711*
MASKX   001244       759#
MASTEK  006142      1645     1774#
MCRLF   005672      1418     1541    1641    1642    1650    1774#   1915    1977
MCSRX   006072      1324     1774#
MDATA   007544      1571     1581    1791#
MEMDAT  035022      4728     4777    4826    4875    4924    4973    5022    5071    5120    5169    5218    5267    5316
                    5365     5414    5463    5512    5561    5610    5659    5708    5757    5806    5855    5904    5953
                    6002     6051    6100    6149    6198    6247    6625#
MEMLD   034676      4684     4727    4776    4825    4874    4923    4972    5021    5070    5119    5168    5217    5266
                    5315     5364    5413    5462    5511    5560    5609    5658    5707    5756    5805    5854    5903
                    5952     6001    6050    6099    6148    6197    6246    6570#
MEMLIM  001304       775#    1287*   4477
MEPASS  005733      1323     1774#
MERRPC  006217      1648     1774#
MERRX   006117      1330     1774#
MERR2   005760      1774#    2177
MERR3   006005      1260     1774#
MILK    001322       783#    1071*   1332    1875*   1580*   1884
MLOCK   006043      1301     1774#
MNEW    006144      1255     1774#
MODU    006704      1774#    2040
MPASSX  006106      1328     1774#
MPFAIL  005675      1708     1774#
MQM     005666      1448     1774#   1939    2014    2024    2034    2049    2063
MR      005755      1310     1774#   1933
MRESET  004000       683#
MSTCLR  104412       822#    1713    2311    2352    2382    2412    2442    2472    2502    2532    2562    2592    2622
                    2652     2682    2712    2742    2772    2802    2832    2875    2919    2961    3002    3044    3070
                    3126     3182    3234    3287    3345    3405    3457    3509    3561    3613    3665    3717    3769
                    3821     3882    3933    3999    4086    4115    4153    4229    4265    4298    4337    4373    4408
                    4444     4499    4539    4582    4637    4683    4724    4773    4822    4871    4920    4969    5018
                    5067     5116    5165    5214    5263    5312    5361    5410    5459    5508    5557    5606    5655
                    5704     5753    5802    5851    5900    5949    5998    6047    6096    6145    6194    6243    6289
                    6340     6389
MTITLF  001000       723#    1098
MTSTN   006130      1646     1774#   1917
MTSTPC  006031      1774#
MVECX   006100      1326     1774#
NEXT    001216       744#    1386    1679    2261*   2289*   2308*   2350*   2380*   2410*   2440*   2470*   2500*   2530*
                    2560*    2590*   2620*   2650*   2680*   2710*   2740*   2770*   2800*   2830*   2873*   2916*   2958*
                    2999*    3042*   3067*   3123*   3179*   3231*   3284*   3342*   3402*   3454*   3506*   3558*   3610*
                    3662*    3714*   3766*   3818*   3879*   3930*   3996*   4055*   4084*   4113*   4151*   4194*   4227*
                    4263*    4296*   4335*   4371*   4406*   4442*   4496*   4536*   4579*   4635*   4680*   4721*   4770*
                    4819*    4868*   4917*   4966*   5015*   5064*   5113*   5162*   5211*   5260*   5309*   5358*   5407*
                    5456*    5505*   5554*   5603*   5652*   5701*   5750*   5799*   5848*   5897*   5946*   5995*   6044*
                    6093*    6142*   6191*   6240*   6287*   6338*   6387*
NOACT   007154      1110     1774#   1869
NODEV   002674      1164     1203#
NPRSET  034634      4199     4231    4267    4301    4340    4374    4409    4454    6552#
NUM     006450      1774#    1969
OK      002646      1190     1197#   1226
ONE     001302       774#
PACT00  001702       952#
PACT01  001706       955#
PACT02  001712       958#
```

```
PACT03   001716       961#
PACT04   001722       964#
PACT05   001726       967#
PACT06   001732       970#
PACT07   001736       973#
PACT10   001742       976#
PACT11   001746       979#
PACT12   001752       982#
PACT13   001756       985#
PACT14   001762       988#
PACT15   001766       991#
PACT16   001772       994#
PACT17   001776       997#
PARAM    104405       812#   1918   1970   1983   1992   2071   2080
PARAM1   004234      1468#   1485
PARBIT=  040000       683#
PARERR   004310      1471    1473   1475   1484#  1491   1493   1495
PASCNT   001230       749#   1321*  1322   1333   1358   1891*
PERFOR=  004537       683#
PFTAB    005430      1709    1715#
POPRO =  012600       659#   1673
POP1SP=  005726       657#
POP2SP=  022626       661#   1388   6357
PRIO     006547       774#   2000
PS    =  177776       650#   1063*  1298*  2204*  2214*  4063*  4067*  4091*  4095*  4117*  4131*  4155*  4171*
PUSHRO   010046       658#   1670
PUSH1S=  005746       656#
PUSH2S   024646       660#
QV.FLG   001327       791#   1069*  1337*  1377
RESREG   005220      1662    1665#
RESTAR   005350      1695    1701#
RESTRT   003534      1336    1340   1348#
RES05 =  104407       816#   1665
RETURN   001214       743#   1079*  1307*  1311   1348*  1386*  1390   1679*  1632   1714   1932*  1942*  1944
ROMCLK   104414       826#   1721   1724   1761   1766   3076   3078   3096   3098   3132   3134   3152   3154
                     3187    3189   3205   3207   3239   3241   3257   3259   3293   3295   3313   3315   3351
                     3353    3373   3375   3410   3412   3428   3430   3462   3464   3480   3482   3514   3516
                     3532    3534   3566   3568   3584   3586   3618   3620   3636   3638   3670   3672   3688
                     3690    3722   3724   3740   3742   3774   3776   3792   3794   3828   3836   3854   3886
                     3888    3904   3906   3939   3943   3945   3962   3966   3968   4006   4010   4012   4030
                     4032    4065   4093   4129   4169   4206   4238   4242   4244   4274   4307   4315   4317
                     4351    4353   4380   4383   4386   4415   4418   4421   4461   4504   4507   4509   4511
                     4545    4548   4550   4552   4586   4589   4591   4593   4608   4610   4612   4639   4643
                     4646    4652   4656   4659   4689   4691   4693   4695   4734   4738   4740   4783   4787
                     4789    4832   4836   4838   4881   4885   4887   4930   4934   4936   4979   4983   4985
                     5028    5032   5034   5077   5081   5083   5126   5130   5132   5175   5179   5181   5224
                     5228    5230   5273   5277   5279   5322   5326   5328   5371   5375   5377   5420   5424
                     5426    5469   5473   5475   5518   5522   5524   5567   5571   5573   5616   5620   5622
                     5665    5669   5671   5714   5718   5720   5763   5767   5769   5812   5816   5818   5861
                     5865    5867   5910   5914   5916   5959   5963   5965   6008   6012   6014   6057   6061
                     6063    6106   6110   6112   6155   6159   6161   6204   6208   6210   6253   6257   6259
                     6306    6394   6407   6410   6413   6426   6429   6432   6438   6441   6444   6447   6455
                     6459    6471   6489   6501   6504   6514   6518   6521   6561   6577   6580   6597   6608
                     6610    6618   6620
RUN      001316       780#   1072*  1873*  1874*  1981
SAVACT   001312       778#   1258   2175*
```

```
SAVNUM  001314           779#    1066*   1335*   1338*   2168*
SAVPC   001276           772#    1209*   1229    1516*   1686
SAVR0   001260           765#    1525*   1530
SAVR1   001262           766#    1215*   1524*   1531    6632
SAVR2   001264           767#    1523*   1532    6632
SAVR3   001266           768#    1522*   1533
SAVR4   001270           769#    1521*   1534    6632
SAVR5   001272           770#    1520*   1535    6632
SAVSP   001274           771#
SAV05 = 104406           814#    1625
SCOPE = 104400           802#    2276    2296    2338    2368    2398    2428    2458    2488    2518    2548    2578    2608
                        2638    2668    2698    2728    2758    2788    2818    2861    2904    2946    2987    3028    3055
                        3111    3167    3219    3271    3328    3390    3442    3494    3546    3598    3650    3702    3754
                        3806    3867    3918    3984    4044    4073    4101    4137    4178    4214    4250    4282    4323
                        4359    4394    4429    4482    4525    4567    4623    4669    4703    4752    4801    4850    4899
                        4948    4997    5046    5095    5144    5193    5242    5291    5340    5389    5438    5487    5536
                        5585    5634    5683    5732    5781    5830    5879    5928    5977    6026    6075    6124    6173
                        6222    6271    6317    6372    6530
SCOP1 = 104401           804#    2270    2320    2333    2360    2367    2390    2397    2420    2427    2450    2457    2480
                        2487    2510    2517    2540    2547    2570    2577    2600    2607    2630    2637    2660    2667
                        2690    2697    2720    2727    2750    2757    2780    2787    2810    2817    2842    2856    2885
                        2899    2928    2941    2969    2977    2986    3010    3018    3027    3086    3106    3142    3162
                        3196    3214    3248    3266    3303    3323    3363    3385    3419    3437    3471    3489    3523
                        3541    3575    3593    3627    3645    3679    3697    3731    3749    3783    3801    3843    3861
                        3895    3913    3952    3975    4019    4039    4469    4518    4559    4600    4619    4702    4747
                        4796    4845    4894    4943    4992    5041    5090    5139    5188    5237    5286    5335    5384
                        5433    5482    5531    5580    5629    5678    5727    5776    5825    5874    5923    5972    6021
                        6070    6119    6168    6217    6266
SETC    035010          5515    5564    5613    5662    5711    5760    5809    5858    5907    5956    6005    6054    6103
                        6152    6201    6250    6615#
SETVEC  034612          4059    4087    4124    4164    6542#
SKIP    002632          1160    1163    1184    1187    1193#
SOFTSW  010052          1816    1855#
SPACNT  004713          1551*   1575    1578*   1592#
SPDAT   035032          4730    4779    4828    4877    4926    4975    5024    5073    5122    5171    5220    5269    5318
                        5367    5416    5465    5514    5563    5612    5661    5710    5759    5808    5857    5906    5955
                        6004    6053    6102    6151    6200    6249    6628#
SPEED   007340          1774#   2027
SPLD    034732          4686    4729    4778    4827    4876    4925    4974    5023    5072    5121    5170    5219    5268
                        5317    5366    5415    5464    5513    5562    5611    5660    5709    5758    5807    5856    5905
                        5954    6003    6052    6101    6150    6199    6248    6587#
STACK   001200           651#    1064    1277    1680    1703    4072    4100    6363
STAT    001240           757#
STAT1   001366           838#    1888*   4118    4156    4641    4654    6457    6516
STAT2   001370           839#    1889*
STAT3   001372           840#    1890*
STRTSW  001236           756#    1100*   1103*   1104    1106    1116    1118    1253    1299    1308    1910    1934*   1964
                        2188
SV05    004402          1520#
SWFLG   010016          1067*   1412    1811*   1838*   1844#
SWMES   007205          1774#   1814
SWMES1  007215          1774#   1817
SWR     001202           730#    1084*   1086    1090*   1100    1258    1263    1368    1375    1399    1414    1614    1619
                        1668    1675    1677    1739    1799    1837*
SWREG   000176           716#    1090    1799    1857
SWOC    000001           632#    1104    1964    2188
```

```
SW01   = 000002      631#   1308   1910   1934
SW02   = 000004      630#
SW03   = 000010      629#   1253
SW04   - 000020      628#
SW05   = 000040      627#
SW06   = 000100      626#   1739
SW07   = 000200      625#
SW08     000400      624#   1675
SW09   = 001000      623#   1399
SW10   - 002000      622#   1677
SW11   - 004000      621#   1375
SW12   - 010000      620#   1414   1614
SW13     020000      619#   1619
SW14     040000      618#
SW15   - 100000      617#
TDATA    024542     4685    4687   4704#
TEMP     001416      858#   1558   1705*  1706*  1747*  1752*  1758*  1770*  6290*  6310*  6341*  6348*
TEMP1    001246      760#   1124*  1776   2200*  2201*  6291*  6311*  6312
TEMP2    001250      761#   1125*  1778
TEMP3    001252      762#   1127*  1153*  1205   1214*  1780   1973   1976   2088*
TEMP4    001254      763#   1128*  1782   1986   1989   1995   1998   2074   2077   2083   2086
TEMP5    001256      764#   1129*  1784   1967*  1980*  2186
TIMER    104416      830#
TKCSR    001204      735#   1435   1801   1846   2242
TKDBR    001206      736#   1437   1443   1804   1806   1848   2244
TLAST    033736     1937    6632#
TPCSR    001210      737#   1419   1441   1616   1849   2245
TPDBR    001212      738#   1421*  1443*  1618*  1851*  2247*
TRPOK    004730     1604#
TSTNO    001226      748#   1078*  1689   1717   1921   1928   1930   2260*  2288*  2307*  2349*  2379*  2409*
                    2439*   2469*  2499*  2529*  2559*  2589*  2619*  2649*  2679*  2709*  2739*  2769*  2799*
                    2829*   2872*  2915*  2957*  2998*  3041*  3066*  3122*  3178*  3230*  3283*  3341*  3401*
                    3453*   3505*  3557*  3609*  3661*  3713*  3765*  3817*  3878*  3929*  3995*  4054*  4083*
                    4112*   4150*  4193*  4226*  4262*  4295*  4334*  4370*  4405*  4440*  4495*  4535*  4578*
                    4634*   4679*  4720*  4769*  4818*  4867*  4916*  4965*  5014*  5063*  5112*  5161*  5210*
                    5259*   5308*  5357*  5406*  5455*  5504*  5553*  5602*  5651*  5700*  5749*  5798*  5847*
                    5896*   5945*  5994*  6043*  6092*  6141*  6190*  6239*  6286*  6337*  6386*
TST1     012320     1924    1942   2260#
TST10    013176     2440    2469#
TST100   025256     4819    4867#
TST101   025432     4868    4916#
TST102   025606     4917    4965#
TST103   025762     4966    5014#
TST104   026136     5015    5063#
TST105   026312     5064    5112#
TST106   026466     5113    5161#
TST107   026642     5162    5210#
TST11    013274     2470    2499#
TST110   027016     5211    5259#
TST111   027172     5260    5308#
TST112   027346     5309    5357#
TST113   027522     5358    5406#
TST114   027676     5407    5455#
TST115   030052     5456    5504#
TST116   030226     5505    5553#
TST117   030402     5554    5602#
```

```
TST12    013372          2500    2529#
TST120   030556          5603    5651#
TST121   030732          5652    5700#
TST122   031106          5701    5749#
TST123   031262          5750    5798#
TST124   031436          5799    5847#
TST125   031612          5848    5896#
TST126   031766          5897    5945#
TST127   032142          5946    5994#
TST13    013470          2530    2559#
TST130   032316          5995    6043#
TST131   032472          6044    6092#
TST132   032646          6093    6141#
TST133   033022          6142    6190#
TST134   033176          6191    6239#
TST135   033352          6240    6286#
TST136   033544          6287    6337#
TST137   033736          6338    6386#   6432
TST14    013566          2560    2589#
TST140=  ****** U        6387
TST15    013664          2590    2619#
TST16    013762          2620    2649#
TST17    014060          2650    2679#
TST2     012426          2261    2288#
TST20    014156          2680    2709#
TST21    014254          2710    2739#
TST22    014352          2740    2769#
TST23    014450          2770    2799#
TST24    014546          2800    2829#
TST25    014672          2830    2872#
TST26    015016          2873    2915#
TST27    015146          2916    2957#
TST3     012456          2289    2307#
TST30    015306          2958    2998#
TST31    015450          2999    3041#
TST32    015534          3042    3066#
TST33    015734          3067    3122#
TST34    016134          3123    3178#
TST35    016310          3179    3230#
TST36    016464          3231    3283#
TST37    016664          3284    3341#
TST4     012606          2308    2349#
TST40    017104          3342    3401#
TST41    017260          3402    3453#
TST42    017434          3454    3505#
TST43    017610          3506    3557#
TST44    017764          3558    3609#
TST45    020140          3610    3661#
TST46    020314          3662    3713#
TST47    020470          3714    3765#
TST5     012704          2350    2379#
TST50    020644          3766    3817#
TST51    021072          3818    3878#
TST52    021242          3879    3929#
TST53    021510          3930    3995#
TST54    02173[illegible]          3996    4054#
```

```
TST55    022026         4055     4083#
TST56    022120         4084     4112#
TST57    022240         4113     4150#
TST6     013002         2380     2409#
TST60    022404         4151     4193#
TST61    022510         4194     4226#
TST62    022624         4227     4262#
TST63    022726         4263     4295#
TST64    023064         4296     4334#
TST65    023210         4335     4370#
TST66    023320         4371     4405#
TST67    023426         4406     4440#
TST7     013100         2410     2439#
TST70    023624         4442     4495#
TST71    023752         4496     4535#
TST72    024104         4536     4578#
TST73    024304         4579     4634#
TST74    024440         4635     4679#
TST75    024552         4680     4720#
TST76    024726         4721     4769#
TST77    025102         4770     4818#
TTST     003612         1302*    1303*    1305*    1306*    1369#
TWOSYN   010000          683#
TYPDAT   005206         1640     1658     1661#
TYPE     104402          806#    1098     1110     1122     1207     1212     1220     1223     1255     1260     1301     1310     1323
                        1324     1326     1328     1330     1418     1431     1448     1541     1581     1641     1642     1645     1646
                        1648     1650     1654     1659     1708     1814     1817     1869     1915     1933     1939     1977     1999
                        2013     2016     2023     2026     2033     2039     2048     2055     2062     2177
TYPMSG   005106         1638     1641#
VEC      006526         1774#    1991
VECMAP   012010         2176     2188#
WHICH    012002         1979     2184#
WRDCNT   004710         1549*    1582*    1590#
WRKO.F   005174         1653     1656#
XBX      005000         1615     1617     1619#
XCSR     003546         1325     1350#
XERR     003570         1331     1359#
XHEAD    006224         1122     1774#
XLOC     003022         1180*    1198*    1201     123[illegible]     1245#
XPASS    003562         1329     1356#
XSTATQ   007454         1131     1774#
XTSTN    005330         1647     1687#
XVEC     003554         1327     1353#
ZERO     001300          773#
$COD     ****** U        588
$CRAP    177777          588#    2251#    2254     2256#    2280#    2283     2284#    2297#    2300     2303#    2339#    2342     2345#
                        2369#    2372     2375#    2399#    2402     2405#    2429#    2432     2435#    2459#    2462     2465#    2489#
                        2492     2495#    2519#    2522     2525#    2549#    2552     2555#    2579#    2582     2585#    2609#    2612
                        2615#    2639#    2642     2645#    2669#    2672     2675#    2699#    2702     2705#    2729#    2732     2735#
                        2759#    2762     2765#    2789#    2792     2795#    2819#    2822     2825#    2862#    2865     2868#    2905#
                        2908     2911#    2947#    2950     2953#    2988#    2991     2994#    3029#    3032     3037#    3056#    3059
                        3062#    3112#    3115     3118#    3168#    3171     3174#    3220#    3223     3226#    3272#    3275     3279#
                        3329#    3332     3337#    3391#    3394     3397#    3443#    3446     3449#    3495#    3498     3501#    3547#
                        3550     3553#    3599#    3602     3605#    3651#    3654     3657#    3703#    3706     3709#    3755#    3758
                        3761#    3807#    3810     3813#    3868#    3871     3874#    3919#    3922     3925#    3985#    3988     3991#
                        4045#    4048     4050#    40[illegible]4#    40[illegible]     4[illegible]9#    41[illegible]#    4105     4108#    4140#    4143     4146#    4184#
```

```
                4187   4189#  4217#  4220   4222#  4253#  4256   4258#  4285#  4288   4291#  4324#  4327
                4330#  4360#  4363   4366#  4395#  4398   4401#  4430#  4433   4436#  4486#  4489   4491#
                4526#  4529   4531#  4568#  4571   4574#  4624#  4627   4630#  4670#  4673   4675#  4708#
                4711   4716#  4757#  4760   4765#  4806#  4809   4814#  4855#  4858   4863#  4904#  4907
                4912#  4953#  4956   4961#  5002#  5005   5010#  5051#  5054   5059#  5100#  5103   5108#
                5149#  5152   5157#  5198#  5201   5206#  5247#  5250   5255#  5296#  5299   5304#  5345#
                5348   5353#  5394#  5397   5402#  5443#  5446   5451#  5492#  5495   5500#  5541#  5544
                5549#  5590#  5593   5598#  5639#  5642   5647#  5688#  5691   5696#  5737#  5740   5745#
                5786#  5789   5794#  5835#  5838   5843#  5884#  5887   5892#  5933#  5936   5941#  5982#
                5985   5990#  6031#  6034   6039#  6080#  6083   6088#  6129#  6132   6137#  6178#  6181
                6186#  6227#  6230   6235#  6276#  6279   6282#  6318#  6321   6333#  6373#  6376   6382#
$ENDAD  [illegible]  [illegible]710  1096   1342#  1666
$N      [illegible]  588#   2251   2256   2258   2264#  2280   2284   2286   2291#  2297   2303   2305   2311
                2312#  2339   2345   2347   2352   2353#  2369   2375   2377   2382   2383#  2399   2405
                2407   2412   2413#  2429   2435   2437   2442   2443#  2459   2465   2467   2472   2473#
                2489   2495   2497   2502   2503#  2519   2525   2527   2532   2533#  2549   2555   2557
                2562   2563#  2579   2585   2587   2592   2593#  2609   2615   2617   2622   2623#  2639
                2645   2647   2652   2653#  2669   2675   2677   2682   2683#  2699   2705   2707   2712
                2713#  2729   2735   2737   2742   2743#  2759   2765   2767   2772   2773#  2789   2795
                2797   2802   2803#  2819   2825   2827   2832   2833#  2862   2868   2870   2875   2876#
                2905   2911   2913   2919   2920#  2947   2953   2955   2961   2962#  2988   2994   2996
                3002   3003#  3029   3037   3039   3044   3045#  3056   3062   3064   3070   3071#  3112
                3118   3120   3126   3127#  3168   3174   3176   3182   3183#  3220   3226   3228   3234
                3235#  3272   3279   3281   3287   3288#  3329   3337   3339   3345   3346#  3391   3397
                3399   3405   3406#  3443   3449   3451   3457   3458#  3495   3501   3503   3509   3510#
                3547   3553   3555   3561   3562#  3599   3605   3607   3613   3614#  3651   3657   3659
                3665   3666#  3703   3709   3711   3717   3718#  3755   3761   3763   3769   3770#  3807
                3813   3815   3821   3822#  3868   3874   3876   3882   3883#  3919   3925   3927   3933
                3934#  3985   3991   3993   3999   4000#  4045   4050   4052   4057   4059#  4074   4079
                4081   4086   4087#  4102   4108   4110   4115   4116#  4140   4146   4148   4153   4154#
                4184   4189   4191   4196   4198#  4217   4222   4224   4229   4230#  4253   4258   4260
                4265   4266#  4285   4291   4293   4298   4299#  4324   4330   4332   4337   4338#  4360
                4366   4368   4373   4374#  4395   4401   4403   4408   4409#  4430   4436   4438   4444
                4445#  4486   4491   4493   4499   4500#  4526   4531   4533   4539   4540#  4568   4574
                4576   4582   4583#  4624   4630   4632   4637   4638#  4670   4675   4677   4683   4684#
                4708   4716   4718   4724   4725#  4757   4765   4767   4773   4774#  4806   4814   4816
                4822   4823#  4855   4863   4864   4865   4871   4872#  4904   4912   4913   4914   4920
                4921#  4953   4961   4962   4963   4969   4970#  5002   5010   5011   5012   5018   5019#
                5051   5059   5060   5061   5067   5068#  5100   5108   5109   5110   5116   5117#  5149
                5157   5158   5159   5165   5166#  5198   5206   5207   5208   5214   5215#  5247   5255
                5256   5257   5263   5264#  5296   5304   5305   5306   5312   5313#  5345   5353   5354
                5355   5361   5362#  5394   5402   5403   5404   5410   5411#  5443   5451   5452   5453
                5459   5460#  5492   5500   5501   5502   5508   5509#  5541   5549   5550   5551   5557
                5558#  5590   5598   5599   5600   5606   5607#  5639   5647   5648   5649   5655   5656#
                5688   5696   5697   5698   5704   5705#  5737   5745   5746   5747   5753   5754#  5786
                5794   5795   5796   5802   5803#  5835   5843   5844   5845   5851   5852#  5884   5892
                5893   5894   5900   5901#  5933   5941   5942   5943   5949   5950#  5982   5990   5991
                5992   5998   5999#  6031   6039   6040   6041   6047   6048#  6080   6088   6089   6090
                6096   6097#  6129   6137   6138   6139   6145   6146#  6178   6186   6187   6188   6194
                6195#  6227   6235   6236   6237   6243   6244#  6276   6282   6283   6284   6289   6290#
                6318   6333   6334   6335   6340   6341#  6373   6382   6383   6384   6389   6390#  6632#
$       [illegible]  588#   2261   2264#  2289   2291#  2308   2312#  2350   2353#  2380   2383#  2410   2413#
                2440   2443#  2470   2473#  2500   2503#  2530   2533#  2560   2563#  2590   2593#  2620
                [illegible]  2650   2653#  2680   2683#  2710   2713#  2740   2743#  2770   2773#  2800   2803#
                [illegible]  2833#  2873   2876#  [illegible]  2920#  2958   2962#  2999   3003#  3042   3045#  3067
                [illegible]  [illegible]  [illegible]  [illegible]  3183#  [illegible]  [illegible]  [illegible]  [illegible]  [illegible]  3346#  [illegible]  3406#
```

```
                         3454   3458#  3506   3510#  3558   3562#  3610   3614#  3662   3666#  3714   3718#  3766
                         3770#  3818   3822#  3879   3883#  3930   3934#  3996   4000#  4055   4059#  4084   4087#
                         4113   4116#  4151   4154#  4194   4198#  4227   4230#  4263   4266#  4296   4299#  4335
                         4338#  4371   4374#  4406   4409#  4442   4445#  4496   4500#  4536   4540#  4579   4583#
                         4635   4638#  4680   4684#  4721   4725#  4770   4774#  4819   4823#  4868   4872#  4917
                         4921#  4966   4970#  5015   5019#  5064   5068#  5113   5117#  5162   5166#  5211   5215#
                         5260   5264#  5309   5313#  5358   5362#  5407   5411#  5456   5460#  5505   5509#  5554
                         5558#  5603   5607#  5652   5656#  5701   5705#  5750   5754#  5799   5803#  5848   5852#
                         5897   5901#  5946   5950#  5995   5999#  6044   6048#  6093   6097#  6142   6146#  6191
                         6195#  6240   6244#  6287   6290#  6338   6341#  6387   6390#
$Y          000[illegible]    588#   794#   802    804#   806#   808#   810#   812#   814#   816#   818#   820#   822#
                          824#   826#   828#   830#   832#
.           036534        695#   696    699#   706#   711#   714#   718#   722#   724#   726#   777#   778#   779#
                          859#   864#   866#   867#   868#   869#   871#   872#   873#   874#   876#   877#   878#
                          879#   881#   882#   883#   884#   886#   887#   888#   889#   891#   892#   893#   894#
                          896#   897#   898#   899#   901#   902#   903#   904#   906#   907#   908#   909#   911#
                          912#   913#   914#   916#   917#   918#   919#   921#   922#   923#   924#   926#   927#
                          928#   929#   931#   932#   933#   934#   936#   937#   938#   939#   941#   942#   943#
                          944#  1102   1112   1244#  1262   1697   1707   1755#  1790#  1792#  1847   1850   1871
                         1965   2009   2129   2133   2180   2211   2243   2246   4452   4642   4655   6368   6458
                         6475   6493   6508   6517   6525   6533#  6536#
.BEGIN      003152       1277#
.CNVRT      004472        821   1542#
.CONVR      004466        819   1541#
.DATAC      005552        829   1746#
.DELAY      005436        825   1719#
.EOP        003364       1318#  6387
.ERRTA      036330       1633   6632#
.HLT        004750        702   1613#
.INSTE      004154        811   1448#
.INSTR      004050        809   1427#
.INST1      004070       1431#  1451
.MSG        004072       1429#  1432#
.MSTCL      005466        823   1730#
.PARAM      004174        813   1459#
.PFAIL      005336        700   1065   1694#  [illegible]  [illegible]  [illegible]
.RES05      004434        817   1530#
.ROMCL      005504        827   1735#
.SAV05      004374        815   1516#
.SCOPE      003576        803   1366#
.SCOP1      003736        805   1398#
.START      002002        719   1063#  1079   1834
.TIMER      005616        831   1757#
.TRPSR      004716        704   1601#
.TRPTA      001330        801#  1606
.TYPE       003[illegible]        807   14[illegible]#
```

DMEND	588#	1312													
DMFRNT	588#														
HLT	662#	2278	2295	2319	2332	2359	2366	2389	2396	2419	2426	2449	2456	2479	2486
	2509	2516	2539	2546	2569	2576	2599	2606	2629	2636	2659	2666	2689	2696	2719
	2726	2749	2756	2779	2786	2809	2816	2841	2855	2884	2898	2927	2940	2968	2976
	2985	3009	3017	3026	3054	3085	3105	3141	3161	3195	3213	3247	3265	3302	3322
	3362	3384	3418	3436	3470	3488	3522	3540	3574	3592	3626	3644	3678	3696	3730
	3748	3782	3800	3842	3860	3894	3912	3951	3974	4018	4038	4069	4071	4097	4099
	4138	4173	4181	4213	4249	4281	4322	4358	4393	4428	4468	4517	4558	4599	4618
	4665	4701	4746	4795	4844	4893	4942	4991	5040	5089	5138	5187	5236	5285	5334
	5383	5432	5481	5530	5579	5628	5677	5726	5775	5824	5873	5922	5971	6020	6069
	6118	6167	6216	6265	6304	6316	6350	6361	6369	6476	6494	6509	6526		
$ADD1	588#	2251													
$ADD2	588#	2280													
$ADD3	588#	2297	2905												
$ADD4	588#	2339	2369	2399	2429	2459	2489	2519	2549	2579	2609	2639	2669	2699	2729
	2759	2789													
$ALU	588#	4708	4757	4806	4855	4904	4953	5002	5051	5100	5149	5198	5247	5296	5345
	5394	5443	5492	5541	5590	5639	5688	5737	5786	5835	5884	5933	5982	6031	6080
	6129	6178	6227												
$ALU0	588#	4670													
$AUTO	588#	1134													
$BR	588#	3883	3900												
$BUFFE	588#	1786													
$COMP	588#	6300													
$CYCLE	588#	1858													
$EOP	588#	1312													
$FINI	588#	6632													
$FLOAT	588#	3072	3091	3128	3147	3184	3201	3236	3253	3289	3308	3347	3368	3407	3424
	3459	3476	3511	3528	3563	3580	3615	3632	3667	3684	3719	3736	3771	3788	
$GETPA	588#														
$HEADE	588#														
$INTR	588#	4045	4074												
$IR	588#	2947	2988												
$MARHI	588#	4641	4654	6457	6516										
$MEMFI	588#	4500	4540												
$MEM0	588#	4526													
$MEM1	588#	4486													
$MEM2	588#	4568													
$MEM3	588#	4624													
$MUCK	588#														
$MSG	588#	1774													
$NOISE	588#	6373													
$NPRBI	588#	4430													
$NPR1	588#	4184	4217	4253											
$NPR2	588#	4285	4324												
$NPR3	588#	4360	4395												
$PFAIL	588#	1690													
$POWER	588#	6318													
$PRIO	588#	4102	4140												
$PROC1	588#	3029													
$PROC2	588#	3056	3112	3168	3220	3272	3329	3391	3443	3495	3547	3599	3651	3703	3755
$PROC3	588#	3807													
$PROC4	588#	3868													
$PROC5	588#														
$QUEST	588#	1968	1981	1990	2069	2078									

```
$RAMCL    588#   1718
$RCLK     588#   1721   1724   1761   1766   3076   3078   3096   3098   3132   3134   3152   3154   3187   3189
          3205   3207   3239   3241   3257   3259   3293   3295   3313   3315   3351   3353   3373   3375   3410
          3412   3428   3430   3462   3464   3480   3482   3514   3516   3532   3534   3566   3568   3584   3586
          3618   3620   3636   3638   3670   3672   3688   3690   3722   3724   3740   3742   3774   3776   3792
          3794   3828   3836   3854   3886   3888   3904   3906   3939   3943   3945   3962   3966   3968   4006
          4010   4012   4030   4032   4065   4093   4129   4169   4206   4238   4242   4244   4274   4307   4315
          4317   4351   4353   4380   4383   4386   4415   4418   4421   4461   4504   4507   4509   4511   4545
          4548   4550   4552   4586   4589   4591   4593   4608   4610   4612   4639   4643   4646   4652   4656
          4659   4689   4691   4693   4695   4734   4738   4740   4783   4787   4789   4832   4836   4838   4881
          4885   4887   4930   4934   4936   4979   4983   4985   5028   5032   5034   5077   5081   5083   5126
          5130   5132   5175   5179   5181   5224   5228   5230   5273   5277   5279   5322   5326   5328   5371
          5375   5377   5420   5424   5426   5469   5473   5475   5518   5522   5524   5567   5571   5573   5616
          5620   5622   5665   5669   5671   5714   5718   5720   5763   5767   5769   5812   5816   5818   5861
          5865   5867   5910   5914   5916   5959   5963   5965   6008   6012   6014   6057   6061   6063   6106
          6110   6112   6155   6159   6161   6204   6208   6210   6253   6257   6259   6306   6394   6407   6410
          6413   6426   6429   6432   6438   6441   6444   6447   6455   6459   6471   6489   6501   6504   6514
          6518   6521   6561   6577   6580   6597   6608   6610   6618   6620
$SCOPE    588#   1362
$SIMBC    588#
$SOFTC    588#   1794
$SPF      588#   3935   3957
$SP1      588#   3919
$SP2      588#   3985
$TIMER    588#   6276
$TRPDE    588#    802    804    806    808    810    812    814    816    818    820    822    824    826    828
           830
$TSTN     588#   2258   2286   2305   2347   2377   2407   2437   2467   2497   2527   2557   2587   2617   2647
          2677   2707   2737   2767   2797   2827   2870   2913   2955   2996   3039   3064   3120   3176   3228
          3281   3339   3399   3451   3503   3555   3607   3659   3711   3763   3815   3876   3927   3993   4052
          4081   4110   4148   4191   4224   4260   4293   4332   4368   4403   4438   4493   4533   4576   4632
          4677   4718   4767   4816   4865   4914   4963   5012   5061   5110   5159   5208   5257   5306   5355
          5404   5453   5502   5551   5600   5649   5698   5747   5796   5845   5894   5943   5992   6041   6090
          6139   6188   6237   6284   6335   6384
$VARIA    588#    721
$WRFLT    588#   2834   2847   2877   2890
$WR46     588#   2819   2862
$XZ       588#   2251   2256   2280   2284   2297   2303   2339   2345   2369   2375   2399   2405   2429   2435
          2459   2465   2489   2495   2519   2525   2549   2555   2579   2585   2609   2615   2639   2645   2669
          2675   2699   2705   2729   2735   2759   2765   2789   2795   2819   2825   2862   2868   2905   2911
          2947   2953   2988   2994   3029   3037   3056   3062   3112   3118   3168   3174   3220   3226   3272
          3279   3329   3337   3391   3397   3443   3449   3495   3501   3547   3553   3599   3605   3651   3657
          3703   3709   3755   3761   3807   3813   3868   3874   3919   3925   3985   3991   4045   4050   4074
          4079   4102   4108   4140   4146   4184   4189   4217   4222   4253   4258   4285   4291   4324   4330
          4360   4366   4395   4401   4430   4436   4486   4491   4526   4531   4568   4574   4624   4630   4670
          4675   4708   4716   4757   4765   4806   4814   4855   4863   4904   4912   4953   4961   5002   5010
          5051   5059   5100   5108   5149   5157   5198   5206   5247   5255   5296   5304   5345   5353   5394
          5402   5443   5451   5492   5500   5541   5549   5590   5598   5639   5647   5688   5696   5737   5745
          5786   5794   5835   5843   5884   5892   5933   5941   5982   5990   6031   6039   6080   6088   6129
          6137   6178   6186   6227   6235   6276   6282   6318   6333   6373   6382

. ABS.  036534     000

ERRORS DETECTED:  0
```

```
CZDMCC/I,CZDMCC.SEQ/SOL/CRF=CZDMCC.P11
RUN-TIME: 16 23 1 SECONDS
RUN-TIME RATIO: 65/41=1.5
CORE USED: 28K (55 PAGES)
```