DMC11 DMC-11 FREE RNGS TSTS AH-8566C-MC
CZDMHC0 FICHE 1 OF 1
NOV 1980

MADE IN USA

```
                        .REM    $
```

IDENTIFICATION

PRODUCT CODE:	AC-8564C-MC
PRODUCT NAME:	CZDMHC0 DMC-11 FREE RUNNING TESTS
DATE:	AUGUST 1980
MAINTAINER:	DIAGNOSTICS-MERRIMACK

PAGE 2

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.

CZDMH TESTS THE DMC11-AR AND DMC11-AL MICRO-PROCESSORS (M8200-YA AND M8200-YB), OR THE KMC11 MICRO-PROCESSOR (M8204). FREE RUNNING TESTS ARE PERFORMED. A LINE UNIT (M8201 OR M8202) MUST BE INSTALLED. CZDMH CAN BE USED AS A HEAT TEST DIAGNOSTIC BY MANUFACTURING.

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

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE FIVE DIAGNOSTICS ARE:

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

NOTE: THE NAMES OF THESE DIAGNOSTICS MAY VARIE AS THEY ARE UPDATED.

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (EXCEPT AN LSI-11) WITH MINIMUM 8K MEMORY
ASR 33 (OR EQUILIVALENT)
DMC11-AR WITH DMC11-DA OR DMC11-FA OR
DMC11-AL WITH DMC11-MA OR DMC11-MD

2.2 STORAGE

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

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA SUCH AS DISK ,MAGTAPE,DECTAPE, OR CASSETTE; FOLLOW INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS *500

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

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER.
(ALSO PLACE 'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW BE LOADING INTO CPU)

4. STARTING PROCEEDURE

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

MAP OF DMC11 STATUS

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

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC. THE ABOVE IS ONLY AN EXAMPLE. THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. IN THIS EXAMPLE THE TABLE CONTAINS THE INFORMATION AND STATUS OF TWO DMC11'S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.
IF THE DIAGNOSTIC WAS STARTED WITH SW00=1 INDICATING MANUAL PARAMETER INPUT THEN THE FOLLOWING SHOWS AN EXAMPLE OF THE QUESTIONS ASKED AND SOME EXAMPLE ANSWERS:

```
HOW MANY DMC11'S TO BE TESTED?1
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 MESSAGE
D: SET A SWITCH FOR EACH DMC DESIRED ACTIVE.
EXAMPLE: IF YOU HAVE 4 DMC'S BUT ONLY WANT TO RUN THE FIRST AND THE LAST SET SWR BITS 0 AND 3 = 1. PRESS CONTINUE
E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
F: SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

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

SCOPE SWITCHES

1. SW06 HALT IN ROMCLK ROUTINE BEFORE CLOCKING MICRO-PROCESSOR INSTRUCTION. THIS ALLOWS THE OPERATOR TO SCOPE A MICRO-PROCESSOR INSTRUCTION IN THE STATIC STATE BEFORE IT IS CLOCKED. HIT CONTINUE TO RESUME RUNNING.
2. SW09 (IF ENABLED BY '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 ENABLED; AND THERE IS A HARD ERROR (CONSTANT); SW08 IS BEST. (SW14=1,0, SW10=0, SW09=0, SW08=1). FOR INTERMITTENT 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.
LINE UNIT(M8201)- JUMPERS W1, W2, AND W4 MUST BE IN. JUMPERS W3, AND W5 MUST BE OUT. SW8 OF E26 MUST BE IN THE ON POSITION.
LINE UNIT (M8202)- JUMPER W1 MUST BE IN. SW8 OF E26 MUST BE IN THE OFF POSITION.

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 CZDMH 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 PERFORM FREE RUNNING TESTS ON KMC
          (MUST BE SET MANUALLY. SEE TEST 1)
        BIT1=0 DMC11-AR (LOW SPEED)
        BIT1=1 DMC11-AL (HIGH SPEED)

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

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

8.5.2 FINDING THE VECTOR AND BR LEVEL

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

8.6 SOFTWARE SWITCH REGISTER

IF THE DIAGNOSTIC IS RUN ON AN 11/04 OR OTHER CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED TO ALLOW USER THE SAME SWITCH OPTIONS AS DESCRIBED PREVIOUSLY. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THIS SOFTWARE SWITCH REGISTER IS USED.
CONTROL:
TO OBTAIN CONTROL AT ANY ALLOWABLE TIME DURING EXECUTION OF THE DIAGNOSTIC THE OPERATOR TYPES A CTRL G ON THE CONSOLE TERMINAL KEYBOARD. AS SOON AS THE CTRL G IS RECOGNIZED, BY THE DIAGNOSTIC, THE FOLLOWING MESSAGE WILL BE DISPLAYED:

```
556
557
558
559                                     SWR=XXXXXX NEW?
560
561                                 WHERE XXXXXX IS THE CURRENT CONTENTS OF  THE  SOFTWARE  SWITCH
562                                 REGISTER  IN  OCTAL.  THE  SOFTWARE CONTROL ROUTINE WILL THEN
563                                 AWAIT OPERATOR ACTION.  AT WHICH TIME THE OPERATOR IS REQUIRED
564                                 TO  TYPE  ONE  OR  MORE OF THE LEGAL CHARACTERS:  1) 0 - 7, 2)
565                                 LINE FEED(<LF>), 3) CARRIAGE  RETURN(<CR>),  OR  4)  CONTROL-U
566                                 (CTRL  U).  NO  CHECK  IS  MADE  FOR  LEGALITY.  IF THE INPUT
567                                 CHARACTER IS NOT A <LF>, <CR>, OR CTRL U IT IS ASSUMED  TO  BE
568                                 AN OCTAL DIGIT.
569
570                                 TO CHANGE THE CONTENTS OF THE SSR THE  OPERATOR  SIMPLY  TYPES
571                                 THE  NEW  DESIRED  VALUE  IN OCTAL - LEADING ZEROS NEED NOT BE
572                                 TYPED.  AND TERMINATES THE INPUT STRING WITH A  <CR>  OR  <LF>
573                                 DEPENDING  ON  THE  PROGRAM ACTION DESIRED AS DESCRIBED BELOW.
574                                 THE INPUT VALUE WILL BE TRUNCATED TO THE LAST 6 DIGITS  TYPED.
575                                 AT  LEAST  ONE  DIGIT  MUST BE TYPED ON ANY GIVEN INPUT STRING
576                                 PRIOR TO THE TERMINATOR BEFORE A CHANGE TO THE SSR WILL OCCUR.
577
578                                 WHEN THE INPUT STRING IS TERMINATED WITH A <CR> THE DIAGNOSTIC
579                                 WILL  CONTINUE  EXECUTION  FROM  THE  POINT  AT  WHICH  IT WAS
580                                 INTERRUPTED.  IF A <CR> IS THE ONLY THING  TYPED  THE  PROGRAM
581                                 WILL CONTINUE WITHOUT CHANGING THE SSR.  THE <LF> DIFFERS FROM
582                                 THE <CR> BY RESTARTING THE PROGRAM AS IF IT WERE RESTARTED  AT
583                                 ADDRESS 200.
584
585                                 IF A CTRL U IS TYPED AT ANY POINT IN THE INPUT STRING PRIOR TO
586                                 THE  TERMINATOR  THE  INPUT  VALUE WILL BE DISREGARDED AND THE
587                                 PROMPT DISPLAYED (SWR = XXXXXX NEW?).
588
589                                 TO SET THE SSR FOR  THE  STARTING  SWITCHES,  FIRST  LOAD  THE
590                                 DIAGNOSTIC, THEN HIT CTRL G, THEN START THE DIAGNOSTIC.
591                     $
```

```
                                ;*CZDMHC   DMC11 FREE RUNNING TESTS
                                ;*COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                ;*-----------------------------------------------------------------

                                ;STARTING PROCEDURE
                                ;LOAD PROGRAM
                                ;LOAD ADDRESS 000200
                                ;SWR=0  AUTOSIZE DMC11
                                ;SW07=1  USE CURRENT DMC11 PARAMETERS
                                ;SW00=1  INPUT NEW DMC11 PARAMETERS
                                ;PRESS START
                                ;PROGRAM WILL TYPE "CZDMHC   DMC11 FREE RUNNING TESTS"
                                ;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
```

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

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

            000024              .=24
000024  005336                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  004750                          .HLT                    ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  004716                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
            000040              .=40
000040  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000042  000000                          0                       ;RETURN ADDRESS IF UNDER ACT-11 OR XXDP
000044  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000046  003522                          $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
            000052              .=52
000052  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  055103  046504  MTITLE: .ASCII  <377><12>/CZDMHC/<377>
(2) 001011     104  041515  030461           .ASCIZ  /DMC11 FREE RUNNING TESTS/<377>
(2)
            001200              .=1200

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

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

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

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

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

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

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

001236  000000          STRTSW: 0               ;SWITCHES AT START OF PROGRAM
001240  000000          STAT:   0               ;DM STATUS WORD STORAGE
001242  000000          CLKX:   0
001244  000000          MASKX:  0
001246  000000          TEMP1:  0               ;TEMPORARY STORAGE
001250  000000          TEMP2:  0               ;TEMPORARY STORAGE
001252  000000          TEMP3:  0               ;TEMPORARY STORAGE
001254  000000          TEMP4:  0               ;TEMPORARY STORAGE
001256  000000          TEMP5:  0               ;TEMPORARY STORAGE
001260  000000          SAVR0:  0               ;R0 STORAGE
001262  000000          SAVR1:  0               ;R1 STORAGE
001264  000000          SAVR2:  0               ;R2 STORAGE
001266  000000          SAVR3:  0               ;R3 STORAGE
001270  000000          SAVR4:  0               ;R4 STORAGE
001272  000000          SAVR5:  0               ;R5 STORAGE
001274  000000          SAVSP:  0               ;STACK POINTER STORAGE
001276  000000          SAVPC:  0               ;PROGRAM COUNTER STORAGE
001300  000000          ZERO:   0
001302  000001          ONE:    1
001304  000000          MEMLIM: 0               ;HIGHEST LOCATION FOR NPR'S
001306  000001          DMACTV: .BLKW 1         ;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

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

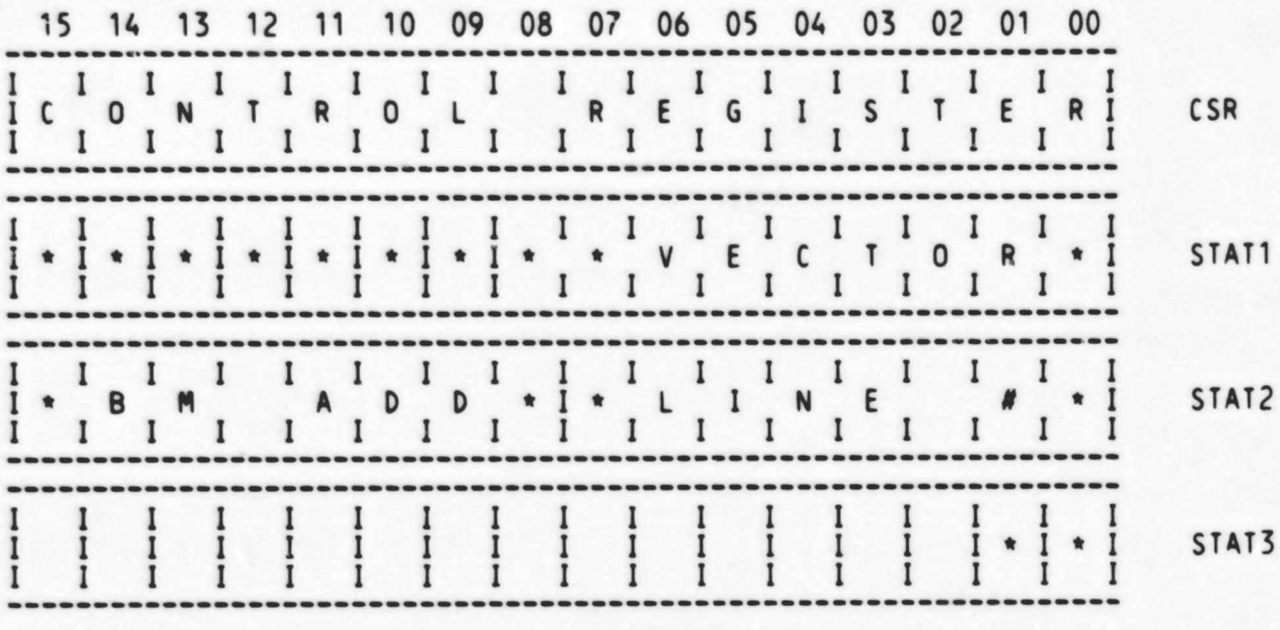
```
                                ;DMC11 CONTROL INDICATORS FOR CURRENT DMC11 UNDER TEST
                                ;-----------------------------------------------------

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)

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

001416  000000                  TEMP:   0
        001460                  .=.+40

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                                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
```

1005

FORMAT OF STATUS TABLE



DEFINITION OF FORMAT

CSR: CONTAINS DMC11 CSR ADDRESS

STAT1: BITS 00-08 IS DMC11 VECTOR ADDRESS
BIT15=1 MICRO-PROCESSOR HAS CRAM
BIT15=0 M!CRO-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

```
 1060
 1061                                        ;PROGRAM INITIALIZATION
 1062                                        ;LOCK OUT INTERRUPTS
 1063                                        ;SET UP PROCESSOR STACK
 1064                                        ;SET UP POWER FAIL VECTOR
 1065                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
 1066                                        ;TYPE TITLE MESSAGE
 1067
 1068  002002  012737  000340  177776  .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
 1069  002010  012706  001200                  MOV     #STACK,SP       ;SET UP STACK
 1070  002014  012737  005336  000024          MOV     #.PFAIL,@#24    ;SET UP POWER FAIL VECTOR
 1071  002022  013737  001310  001314          MOV     DMNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
 1072  002030  005037  010016                  CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
 1073  002034  105037  001325                  CLRB    ERRFLG          ;CLEAR ERROR FLAG
 1074  002040  105037  001327                  CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
 1075  002044  012737  001470  001320          MOV     #DM.MAP-10,CREAM;GET MAP POINTER.
 1076  002052  012737  001676  001322          MOV     #CNT.MAP-4,MILK ;GET PASS COUNT MAP POINTER
 1077  002060  012737  100000  001316          MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
 1078  002066  012700  001702                  MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
 1079  002072  005020                  23$:     CLR     (R0)+           ;CLEAR TABLE
 1080  002074  022700  002002                  CMP     #CNT.MAP+100,R0 ;DONE YET?
 1081  002100  001374                          BNE     23$             ;KEEP GOING
 1082  002102  005037  001234                  CLR     LSTERR          ;CLEAR LAST ERROR POINTER
 1083  002106  012737  000001  001226          MOV     #1,TSTNO        ;SET UP FOR TEST 1
 1084  002114  012737  002002  001214          MOV     #.START,RETURN  ;SET UP FOR POWER FAIL BEFORE
 1085                                                                  ;TESTING STARTS
 1086  002122  013746  000006                  MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
 1087  002126  013746  000004                  MOV     @#4,-(SP)       ;
 1088  002132  012737  002166  000004          MOV     #6$,@#4         ;SET UP FOR TIMEOUT
 1089  002140  012737  177570  001202          MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
 1090  002146  012737  177570  001200          MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
 1091  002154  022777  177777  177020          CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
 1092  002162  001402                          BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
 1093  002164  000407                          BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
 1094  002166  022626                  6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
 1095  002170  012737  000176  001202          MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
 1096  002176  012737  000174  001200          MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
 1097  002204  012637  000004          7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
 1098  002210  012637  000006                  MOV     (SP)+,@#6       ;
 1099  002214  105737  001324                  TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
 1100  002220  001006                          BNE     20$             ;BR IF YES
 1101  002222  022737  003522  000042          CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
 1102  002230  001402                          BEQ     20$
 1103  002232  104402  001000                  TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
 1104  002236  004737  007606          20$:    JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
 1105  002242  017737  176734  001236          MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
 1106  002250  005737  000042                  TST     @#42            ;IS IT RUNNING IN AUTO MODE?
 1107  002254  001402                          BEQ     .+6             ;BR IF NO
 1108  002256  005037  001236                  CLR     STRTSW          ;IF YES, CLEAR SWITCHES
 1109  002262  032737  000001  001236          BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
 1110  002270  001012                          BNE     17$             ;BR IF SW00=1
 1111  002272  105737  001236                  TSTB    STRTSW          ;BIT7=1??
 1112  002276  100007                          BPL     17$             ;BR IF SW07=0
 1113  002300  005737  001306                  TST     DMACTV          ;ARE ANY DEVICES SELECTED?
 1114  002304  001006                          BNE     16$             ;BR IF YES
 1115  002306  104402  007154                  TYPE,   NOACT           ;NO DEVICES SELECTED.
```

```
1116  002312  000000                        HALT                    ;STOP THE SHOW
1117  002314  000776                        BR      .-2             ;DISQUALIFY CONTINUE SWITCH
1118  002316  004737  010512        17$:    JSR     PC,AUTO.SIZE    ;GO DO THE AUTO SIZE
1119  002322  105737  001324        16$:    TSTB    INIFLG          ;FIRST TIME?
1120  002326  001410                        BEQ     21$             ;BR IF YES
1121  002330  105737  001236                TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
1122  002334  100431                        BMI     1$
1123  002336  032737  000006  001236        BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
1124  002344  001403                        BEQ     24$             ;IF NO THEN TYPE STATUS
1125  002346  000424                        BR      1$              ;IF YES DO NOT TYPE STATUS
1126  002350  005137  001324        21$:    COM     INIFLG          ;SET FLAG
1127  002354  104402  006224        24$:    TYPE    ,XHEAD          ;TYPE HEADER
1128  002360  012704  001500                MOV     #DM.MAP,R4      ;SET POINTER
1129  002364  010437  001246        5$:     MOV     R4,TEMP1        ;SET ADDRESS
1130  002370  012437  001250                MOV     (R4)+,TEMP2     ;SET CSR
1131  002374  001411                        BEQ     1$              ;ALL DONE IF ZERO
1132  002376  012437  001252                MOV     (R4)+,TEMP3     ;SET STAT1
1133  002402  012437  001254                MOV     (R4)+,TEMP4     ;SET STAT2
1134  002406  012437  001256                MOV     (R4)+,TEMP5     ;SET STAT3
1135  002412  104410                        CONVRT                  ;TYPE OUT STATUS MAP
1136  002414  007454                        XSTATQ                  ;
1137  002416  000762                        BR      5$
1138  002420  012700  001500        1$:     MOV     #DM.MAP,R0      ;R0 POINTS TO STATUS TABLE
1139
1140                                ;;**************************************************************************
1141                                ;;*AUTO SIZE TEST
1142                                ;;*THIS TEST VERIFYS THAT THE DMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
1143                                ;;*ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
1144                                ;;*CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC).
1145                                ;;*IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE DMC11, THE FIRST
1146                                ;;*DMC11 ADDRESS IS 760070, KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
1147                                ;;*ADDRESS 760000. THIS TEST MAY REQUIRE 2 OR MORE ATTEMPTS TO GET THE
1148                                ;;*RIGHT ADDRESSES. AFTER YOU HAVE CHANGED THE ADDRESS TO WHAT IT TOLD
1149                                ;;*YOU THE FIRST TIME, IT MAY COME BACK AND TELL YOU A DIFFERENT ADDRESS
1150                                ;;*THE NEXT TIME YOU RUN IT. PLEASE HAVE PATIENCE, THE FINAL ADDRESS
1151                                ;;*WILL BE CORRECT (AS LONG AS ALL DEVICES IN FRONT OF THE DMC'S ARE
1152                                ;;*CORRECT).
1153                                ;;**************************************************************************
1154
1155  002424  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
1156  002430  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
1157  002434  005037  000006                CLR     @#6             ;CLEAR VEC+2
1158  002440  005037  001252                CLR     TEMP3           ;CLEAR FLAG
1159  002444  005005                        CLR     R5              ;R5=0=DMC, R5=-1=KMC
1160  002446  011037  001404        AUSTRT: MOV     (R0),DMCSR      ;GET NEXT DMC CSR
1161  002452  001564                        BEQ     AUDONE          ;BR IF DONE
1162  002454  005705                        TST     R5              ;DMC OR KMC?
1163  002456  001005                        BNE     1$              ;BR IF KMC
1164  002460  032760  100000  000002        BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
1165  002466  001061                        BNE     SKIP            ;SKIP IF NOT DMC
1166  002470  000404                        BR      2$              ;ITS A DMC SO CONTINUE
1167  002472  032760  100000  000002  1$:   BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
1168  002500  001454                        BEQ     SKIP            ;SKIP IF NOT KMC
1169  002502  012737  002674  000004  2$:   MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
1170  002510  005705                        TST     R5              ;DMC OR KMC?
1171  002512  001003                        BNE     3$              ;BR IF KMC
```

```
1172  002514  012703  000006              MOV     #6,R3           ;R3 IS COUNT OF DEVICES BEFORE DMC
1173  002520  000402                      BR      4$              ;GO ON
1174  002522  012703  000010      3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
1175  002526  012702  003010      4$:     MOV     #DEVTAB,R2      ;R2 IS DEVICE TABLE PONTER
1176  002532  012701  160010              MOV     #160010,R1      ;START WITH ADDRESS 160010
1177  002536  005711              FLOAT:  TST     (R1)            ;CHECK ADDRESS IN R1
1178  002540  111204                      MOVB    (R2),R4         ;IF NO TIMEOUT, GET NEXT ADDRESS
1179  002542  060401                      ADD     R4,R1           ;IN R1
1180  002544  005201                      INC     R1              ;
1181  002546  040401                      BIC     R4,R1           ;
1182  002550  005703                      TST     R3              ;ANY MORE DEVICES TO CHECK FOR?
1183  002552  001371                      BNE     FLOAT           ;BR IF YES
1184  002554  012737  002700  000004      MOV     #ERR,@#4        ;OK ONLY DMC'S ARE LEFT, SET UP FOR TIMEOUT
1185  002562  010137  003022              MOV     R1,XLOC         ;SAVE FIRST DMC/KMC ADDRESS
1186  002566  005705              FY:     TST     R5              ;DMC OR KMC?
1187  002570  001005                      BNE     1$              ;BR IF KMC
1188  002572  032760  100000  000002      BIT     #BIT15,2(R0)    ;CHECK FOR DMC CSR
1189  002600  001014                      BNE     SKIP            ;SKIP IF NOT DMC
1190  002602  000404                      BR      2$              ;ITS A DMC SO CONTINUE
1191  002604  032760  100000  000002  1$: BIT     #BIT15,2(R0)    ;CHECK FOR KMC CSR
1192  002612  001407                      BEQ     SKIP            ;SKIP IF NOT KMC
1193  002614  005711              2$:     TST     (R1)            ;CHECK DMC ADDRESS
1194  002616  020137  001404              CMP     R1,DMCSR        ;DOES IT MATCH
1195  002622  001411                      BEQ     OK              ;BR IF YES
1196  002624  062701  000010              ADD     #10,R1          ;GET NEXT DMC ADDRESS
1197  002630  000756                      BR      FY              ;DO IT AGAIN
1198  002632  062700  000010      SKIP:   ADD     #10,R0          ;SKIP TO NEXT CSR IN TABLE
1199  002636  011037  001404              MOV     (R0),DMCSR      ;GET NEXT CSR
1200  002642  001470                      BEQ     AUDONE          ;BR IF DONE
1201  002644  000750                      BR      FY              ;ELSE CONTINUE
1202  002646  062700  000010      OK:     ADD     #10,R0          ;SKIP TO NEXT DMC CSR
1203  002652  062737  000010  003022      ADD     #10,XLOC        ;UPDATE EXPECTED DMC/KMC ADDRESS
1204  002660  011037  001404              MOV     (R0),DMCSR      ;GET NEXT DMC/KMC CSR
1205  002664  001457                      BEQ     AUDONE          ;BR IF DONE
1206  002666  013701  003022              MOV     XLOC,R1         ;GET EXPECTED DMC/KMC ADDRESS
1207  002672  000735                      BR      FY              ;CONTINUE
1208  002674  122243              NODEV:  CMPB    (R2)+,-(R3)     ;ON TIMEOUT, INC R2, DEC R3
1209  002676  000002                      RTI                     ;RETURN
1210  002700  005737  001252      ERR:    TST     TEMP3           ;CHECK FLAG IF = 0 TYPE HEADER
1211  002704  001014                      BNE     1$              ;SKIP HEADER
1212  002706  104402                      TYPE                    ;TYPEOUT HEADER MESSAGE
1213  002710  007223                      CONERR                  ;CONFIGURATION ERROR!!!!
1214  002712  012737  002700  001276      MOV     #ERR,SAVPC      ;SAVE PC FOR TYPEOUT
1215  002720  104411                      CNVRT                   ;TYPE OUT ERROR PC
1216  002722  002770                      ERRPC                   ;
1217  002724  104402                      TYPE                    ;TYPE REST OF HEADER
1218  002726  007277                      CNERR                   ;
1219  002730  012737  177777  001252      MOV     #-1,TEMP3       ;SET FLAG SO IT ONLY GETS TYPED ONCE
1220  002736  010137  001262      1$:     MOV     R1,SAVR1        ;SAVE R1 FOR TYPEOUT
1221  002742  104410                      CONVRT
1222  002744  002776                      CONTAB                  ;TYPE CSR VALUES
1223  002746  005705                      TST     R5              ;DMC OR KMC ?
1224  002750  001003                      BNE     3$              ;BR IF KMC
1225  002752  104402                      TYPE
1226  002754  007320                      DMCM
1227  002756  000402                      BR      4$              ;CONTINUE
```

```
1228  002760  104402              3$:     TYPE
1229  002762  007330                      KMCM
1230  002764  022626              4$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
1231  002766  000727                      BR      OK              ;BR TO GET OUT
1232  002770  000001              ERRPC:  1
1233  002772     006      002             .BYTE   6,2
1234  002774  001276                      SAVPC
1235  002776  000002              CONTAB: 2
1236  003000     006      004             .BYTE   6,4
1237  003002  003022                      XLOC
1238  003004     006      002             .BYTE   6,2
1239  003006  001404                      DMCSR
1240  003010     007              DEVTAB: .BYTE   7               ;DJ
1241  003011     017                      .BYTE   17              ;DH
1242  003012     007                      .BYTE   7               ;DQ
1243  003013     007                      .BYTE   7               ;DU
1244  003014     007                      .BYTE   7               ;DUP
1245  003015     007                      .BYTE   7               ;LK
1246  003016     007                      .BYTE   7               ;DMC
1247  003017     007                      .BYTE   7               ;DZ
1248  003020     007                      .BYTE   7               ;KMC
1249        003022              .EVEN
1250  003022  000000              XLOC:   0
1251  003024  005705              AUDONE: TST     R5              ;DMC?
1252  003026  001005                      BNE     1$              ;BR IF KMC AND ALL DONE
1253  003030  012705  177777              MOV     #-1,R5          ;SET R5 TO -1 (KMC)
1254  003034  012700  001500              MOV     #DM.MAP,R0      ;RESET R0 TO START OF TABLE
1255  003040  000602                      BR      AUSTRT          ;GO DO KMC'S
1256  003042  012637  000006      1$:     MOV     (SP)+,@#6       ;RESTORE LOC 6
1257  003046  012637  000004              MOV     (SP)+,@#4       ;RESTORE LOC 4
1258  003052  032737  000010  001236      BIT     #SW03,STRTSW    ;SELECT SPECIFIC DEVICES??
1259  003060  001422                      BEQ     3$              ;BR IF NO.
1260  003062  104402  006144              TYPE    ,MNEW           ;TYPE THE MESSAGE.
1261  003066  005000                      CLR     R0              ;ZERO DATA LIGHTS
1262  003070  000000                      HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1263  003072  027737  176104  001312      CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
1264  003100  101404                      BLOS    2$              ;BR IF NUMBER IS OK.
1265  003102  104402  006005              TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
1266  003106  000000                      HALT                    ;STOP EVERY THING.
1267  003110  000776                      BR      .-2             ;RESTART THE PROGRAM AGAIN.
1268  003112  017737  176064  001306  2$: MOV     @SWR,DMACTV     ;GET NEW DEVICE PATTERN
1269  003120  013700  001306              MOV     DMACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
1270  003124  000000                      HALT                    ;CONTINUE DYNAMIC SWITCHES.
1271  003126  012700  000300      3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
1272  003132  012701  000302              MOV     #302,R1         ;VECTOR AREA. 300-776
1273  003136  010120              4$:     MOV     R1,(R0)+        ;START PUTTING 'PC+2 - HALT'
1274  003140  005021                      CLR     (R1)+           ;IN VECTOR AREA.
1275  003142  022021                      CMP     (R0)+,(R1)+     ;POP POINTERS
1276  003144  022700  001000              CMP     #1000,R0        ;ALL DONE??
1277  003150  001372                      BNE     4$              ;BR IF NO.
1278
1279                              ;TEST START AND RESTART
1280                              ;----------------------
1281
1282  003152  012706  001200      .BEGIN: MOV     #STACK,SP       ;SET UP STACK
1283  003156  013746  000006              MOV     @#6,-(SP)       ;SAVE LOC 6
```

```
1284  003162  013746  000004                    MOV     @#4,-(SP)       ;SAVE LOC 4
1285  003166  005000                            CLR     R0              ;START AT 0
1286  003170  012737  003234  000004            MOV     #2$,@#4         ;SET UP FOR TIME OUT
1287  003176  005037  000006                    CLR     @#6             ;TO AUTOSIZE MEMORY
1288  003202  005720                    6$:     TST     (R0)+           ;CHECK ADDRESS IN R0
1289  003204  022700  157776                    CMP     #157776,R0      ;IS IT AT LEAST 28K
1290  003210  001374                            BNE     6$              ;BR IF NO
1291  003212  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
1292  003216  010037  001304            7$:     MOV     R0,MEMLIM       ;STORE MEMORY LIMIT
1293  003222  012637  000004                    MOV     (SP)+,@#4       ;RESTORE LOC 4
1294  003226  012637  000006                    MOV     (SP)+,@#6       ;RESTORE LOC 6
1295  003232  000413                            BR      10$             ;CONTINUE
1296  003234  022626                    2$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
1297  003236  162700  000004                    SUB     #4,R0           ;GET LAST GOOD ADDRESS
1298  003242  162700  007776                    SUB     #7776,R0        ;SAVE 2K FOR MONITORS
1299  003246  022700  030000                    CMP     #30000,R0       ;IS IT 8K?
1300  003252  001361                            BNE     7$              ;BR IF NO
1301  003254  012700  037400                    MOV     #37400,R0       ;IF 8K DON'T SAVE 2K
1302  003260  000756                            BR      7$              ;
1303  003262  012737  000340  177776    10$:    MOV     #340,PS         ;LOCK OUT INTERRUPTS
1304  003270  032737  000004  001236            BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
1305  003276  001411                            BEQ     1$              ;BR IF NO LOCK DESIRED.
1306  003300  104402  006043                    TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
1307  003304  012737  000240  003612            MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
1308  003312  012737  000240  003614            MOV     #NOP,TTST+2     ;SET UP TO LOCK
1309  003320  000406                            BR      3$              ;CONTINUE ALONG.
1310  003322  013737  003730  003612    1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
1311  003330  013737  003732  003614            MOV     BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1312  003336  012737  010060  001214    3$:     MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
1313  003344  032737  000002  001236    4$:     BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED?
1314  003352  001002                            BNE     5$              ;BR IF YES
1315  003354  104402  005755                    TYPE    ,MR             ;TYPE R
1316  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
```

```
1373  003604  032777  040000  175370          BIT     #BIT14,@SWR     ;"LOOP ON THIS TEST"?
1374  003612  001407                  TTST:   BEQ     1$              ;BR IF NO.  (IF LOCK SW01=1; THIS LOC =240)
1375  003614  000437                          BR      3$              ;GOTO 3$    (IF LOCK SW01=1; THIS LOC =240)
1376  003616  005737  003734                  TST     DONE            ;WAS TKCSR DONE SET?
1377  003622  001434                          BEQ     3$              ;BR IF NO (LOCKED ON TEST)
1378  003624  005037  003734                  CLR     DONE            ;YES, CLEAR FLAG
1379  003630  000415                          BR      2$              ;GO TO NEXT TEST
1380  003632  032777  004000  175342  1$:     BIT     #SW11,@SWR      ;DELETE ITERATION?  (QUICK PASS)
1381  003640  001011                          BNE     2$              ;BR IF YES
1382  003642  105737  001327                  TSTB    QV.FLG          ;HAVE PASSES BEECOMPLETED?
1383  003646  001406                          BEQ     2$              ;BR IF QUICK PASS.
1384  003650  005237  001224                  INC     LPCNT           ;UPDATE ITERATION COUNTER
1385  003654  023737  001224  001222          CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
1386  003662  101414                          BLOS    3$              ;BR IF NOT YET
1387  003664  105037  001325          2$:     CLRB    ERRFLG          ;PREPARE FOR NEW TEST
1388  003670  005037  001224                  CLR     LPCNT           ;START ICOUNTER AT 0
1389  003674  005037  001220                  CLR     LOCK
1390  003700  012737  000020  001222          MOV     #20,ICOUNT      ;RESET ITERATIONS
1391  003706  013737  001216  001214          MOV     NEXT,RETURN     ;GET NEXT TEST
1392  003714  011600                  3$:     MOV     (SP),R0         ;POP R0 OFF OF THE STACK
1393  003716  022626                          POP2SP                  ;FAKE AN "RTI"
1394  003720  013701  001404                  MOV     DMCSR,R1        ;R1 CONTAINS BASE DMC ADDRESS
1395  003724  000177  175264                  JMP     @RETURN         ;GO DO THE TEST
1396  003730  001407                  BRW:    1407
1397  003732  000437                  BRX:    437
1398  003734  000000                  DONE:   0
1399
1400                                          ;CHECK FOR FREEZE ON CURRENT DATA
1401                                          ;-------------------------------
1402
1403  003736  004737  007606          .SCOP1: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
1404  003742  032777  001000  175232          BIT     #SW09,@SWR      ;IS SW09=1(SET)?
1405  003750  001405                          BEQ     1$              ;BR IF NOT SET.
1406  003752  005737  001220                  TST     LOCK
1407  003756  001402                          BEQ     1$
1408  003760  013716  001220                  MOV     LOCK,(SP)       ;GOTO THE ADDRESS IN LOCK.
1409  003764  000002                  1$:     RTI                     ;GO BACK.
1410
1411                                          ;TELETYPE OUTPUT ROUTINE
1412                                          ;-----------------------
1413
1414  003766  010546                  .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
1415  003770  017605  000002                  MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
1416  003774  062766  000002  000002          ADD     #2,2(SP)        ;POP OVER ADDRESS.
1417  004002  005737  010016          4$:     TST     SWFLG           ;SOFT SWR MESSAGE?
1418  004006  001004                          BNE     1$              ;IF YES TYPE IT OUT REGARDLESS OF SW12
1419  004010  032777  010000  175164          BIT     #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
1420  004016  001012                          BNE     3$              ;BR IF NO PRINT OUT WANTED (SW12=1)
1421  004020  105715                  1$:     TSTB    (R5)            ;IS NUMBER MINUS? (MSB=1(BIT7))
1422  004022  100002                          BPL     2$              ;BR IF NUMBER IS PLUS
1423  004024  104402  005672                  TYPE    ,MCRLF          ;TYPE A CR/LF!
1424  004030  105777  175154          2$:     TSTB    @TPCSR          ;TTY READY?
1425  004034  100375                          BPL     2$              ;BR IF NO.
1426  004036  112577  175150                  MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
1427  004042  001357                          BNE     4$              ;IF NOT ZERO KEEP PRINTING!
1428  004044  012605                  3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
```

```
1429  004046  000002                          RTI                     ;GO HOME
1430                                          ;-----------------------
1431
1432  004050  010346                .INSTR:  MOV     R3,-(SP)        ;SAVE R3 ON STACK
1433  004052  010446                         MOV     R4,-(SP)        ;SAVE R4 ON STACK
1434  004054  017637  000004 004072          MOV     @4(SP),.MSG
1435  004062  062766  000002 000004          ADD     #2,4(SP)
1436  004070  104402                .INST1:  TYPE
1437  004072  000000                .MSG:    0
1438  004074  012704  007502                 MOV     #INBUF,R4
1439  004100  012703  000007                 MOV     #7,R3
1440  004104  105777  175074        1$:      TSTB    @TKCSR
1441  004110  100375                         BPL     1$
1442  004112  117714  175070                 MOVB    @TKDBR,(R4)
1443  004116  142714  000200                 BICB    #200,(R4)
1444  004122  122427  000015                 CMPB    (R4)+,#15
1445  004126  001417                         BEQ     INSTR2
1446  004130  105777  175054        2$:      TSTB    @TPCSR
1447  004134  100375                         BPL     2$
1448  004136  017777  175044 175046          MOV     @TKDBR,@TPDBR
1449  004144  005303                         DEC     R3
1450  004146  001356                         BNE     1$
1451  004150  012604                         MOV     (SP)+,R4
1452  004152  012603                         MOV     (SP)+,R3
1453  004154  104402  005666        .INSTE:  TYPE    ,MQM
1454  004160  010346                         MOV     R3,-(SP)
1455  004162  010446                         MOV     R4,-(SP)
1456  004164  000741                         BR      .INST1
1457  004166  012604                INSTR2:  MOV     (SP)+,R4        ;RESTORE R4
1458  004170  012603                         MOV     (SP)+,R3        ;RESTORE R3
1459  004172  000002                         RTI
1460
1461                                          ;CONVERT ASCII STRING TO OCTAL
1462                                          ;----------------------------
1463
1464  004174  010546                .PARAM:  MOV     R5,-(SP)
1465  004176  010446                         MOV     R4,-(SP)
1466  004200  016605  000004                 MOV     4(SP),R5
1467  004204  012537  004364                 MOV     (R5)+,LOLIM
1468  004210  012537  004366                 MOV     (R5)+,HILIM
1469  004214  012537  004370                 MOV     (R5)+,DEVADR
1470  004220  112537  004372                 MOVB    (R5)+,LOBITS
1471  004224  112537  004373                 MOVB    (R5)+,ADRCNT
1472  004230  010566  000004                 MOV     R5,4(SP)
1473  004234  005005                PARAM1:  CLR     R5
1474  004236  012704  007502                 MOV     #INBUF,R4
1475  004242  122714  000015                 CMPB    #15,(R4)
1476  004246  001420                         BEQ     PARERR
1477  004250  121427  000060        1$:      CMPB    (R4),#60
1478  004254  002415                         BLT     PARERR
1479  004256  121427  000067                 CMPB    (R4),#67
1480  004262  003012                         BGT     PARERR
1481  004264  142714  000060                 BICB    #60,(R4)
1482  004270  152405                         BISB    (R4)+,R5
1483  004272  122714  000015                 CMPB    #15,(R4)
1484  004276  001406                         BEQ     LIMITS
```

```
1485  004300  006305                    ASL     R5
1486  004302  006305                    ASL     R5
1487  004304  006305                    ASL     R5
1488  004306  000760                    BR      1$
1489  004310  104404            PARERR: INSTER
1490  004312  000750                    BR      PARAM1
1491
1492                                    ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
1493                                    ;--------------------------------------
1494
1495  004314  020537  004366    LIMITS: CMP     R5,HILIM
1496  004320  101373                    BHI     PARERR
1497  004322  020537  004364            CMP     R5,LOLIM
1498  004326  103770                    BLO     PARERR
1499  004330  133705  004372            BITB    LOBITS,R5
1500  004334  001365                    BNE     PARERR
1501
1502                                    ;STORE NUMBER AT SPECIFIED ADDRESS
1503
1504  004336  013704  004370            MOV     DEVADR,R4
1505  004342  010524            1$:     MOV     R5,(R4)+
1506  004344  062705  000002            ADD     #2,R5
1507  004350  105337  004373            DECB    ADRCNT
1508  004354  001372                    BNE     1$
1509  004356  012604                    MOV     (SP)+,R4
1510  004360  012605                    MOV     (SP)+,R5
1511  004362  000002                    RTI
1512  004364  000000            LOLIM:  0
1513  004366  000000            HILIM:  0
1514  004370  000000            DEVADR: 0
1515  004372  000000            LOBITS: 0
1516          004373            ADRCNT=LOBITS+1
1517
1518                                    ;SAVE PC OF TEST THAT FAILED AND R0-R5
1519                                    ;--------------------------------------
1520
1521  004374  016637  000004  001276  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1522
1523                                    ;SAVE R0-R5
1524
1525  004402  010537  001272    SV05:   MOV     R5,SAVR5        ;SAVE R5
1526  004406  010437  001270            MOV     R4,SAVR4        ;SAVE R4
1527  004412  010337  001266            MOV     R3,SAVR3        ;SAVE R3
1528  004416  010237  001264            MOV     R2,SAVR2        ;SAVE R2
1529  004422  010137  001262            MOV     R1,SAVR1        ;SAVE R1
1530  004426  010037  001260            MOV     R0,SAVR0        ;SAVE R0
1531  004432  000002                    RTI                     ;LEAVE.
1532
1533                                    ;RESTORE R0-R5
1534
1535  004434  013700  001260  .RES05:  MOV     SAVR0,R0        ;RESTORE R0
1536  004440  013701  001262            MOV     SAVR1,R1        ;RESTORE R1
1537  004444  013702  001264            MOV     SAVR2,R2        ;RESTORE R2
1538  004450  013703  001266            MOV     SAVR3,R3        ;RESTORE R3
1539  004454  013704  001270            MOV     SAVR4,R4        ;RESTORE R4
1540  004460  013705  001272            MOV     SAVR5,R5        ;RESTORE R5
```

```
1541  004464  000002                        RTI                     ;LEAVE
1542
1543                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1544                                        ;-----------------------------------------------------
1545
1546  004466  104402  005672        .CONVR: TYPE    ,MCRLF
1547  004472  010046                .CNVRT: MOV     R0,-(SP)
1548  004474  010146                        MOV     R1,-(SP)
1549  004476  010346                        MOV     R3,-(SP)
1550  004500  010446                        MOV     R4,-(SP)
1551  004502  010546                        MOV     R5,-(SP)
1552  004504  017601  000012                MOV     @12(SP),R1
1553  004510  062766  000002  000012        ADD     #2,12(SP)
1554  004516  012137  004710                MOV     (R1)+,WRDCNT
1555  004522  112137  004712        1$:     MOVB    (R1)+,CHRCNT
1556  004526  112137  004713                MOVB    (R1)+,SPACNT
1557  004532  013137  004714                MOV     @(R1)+,BINWRD
1558  004536  122737  000003  004712        CMPB    #3,CHRCNT
1559  004544  001003                        BNE     2$
1560  004546  042737  177400  004714        BIC     #177400,BINWRD
1561  004554  013704  004714        2$:     MOV     BINWRD,R4
1562  004560  113705  004712                MOVB    CHRCNT,R5
1563  004564  012700  001416                MOV     #TEMP,R0
1564  004570  010403                3$:     MOV     R4,R3
1565  004572  042703  177770                BIC     #177770,R3
1566  004576  062703  000060                ADD     #060,R3
1567  004602  110320                        MOVB    R3,(R0)+
1568  004604  000241                        CLC
1569  004606  006004                        ROR     R4
1570  004610  000241                        CLC
1571  004612  006004                        ROR     R4
1572  004614  000241                        CLC
1573  004616  006004                        ROR     R4
1574  004620  005305                        DEC     R5
1575  004622  001362                        BNE     3$
1576  004624  012703  007544                MOV     #MDATA,R3
1577  004630  114023                4$:     MOVB    -(R0),(R3)+
1578  004632  105337  004712                DECB    CHRCNT
1579  004636  001374                        BNE     4$
1580  004640  105737  004713                TSTB    SPACNT
1581  004644  001405                        BEQ     6$
1582  004646  112723  000040        5$:     MOVB    #040,(R3)+
1583  004652  105337  004713                DECB    SPACNT
1584  004656  001373                        BNE     5$
1585  004660  105013                6$:     CLRB    (R3)
1586  004662  104402  007544                TYPE    ,MDATA
1587  004666  005337  004710                DEC     WRDCNT
1588  004672  001313                        BNE     1$
1589  004674  012605                        MOV     (SP)+,R5
1590  004676  012604                        MOV     (SP)+,R4
1591  004700  012603                        MOV     (SP)+,R3
1592  004702  012601                        MOV     (SP)+,R1
1593  004704  012600                        MOV     (SP)+,R0
1594  004706  000002                        RTI
1595  004710  000000            WRDCNT:     0
1596  004712  000000            CHRCNT:     0
```

```
1597            004713                  SPACNT=CHRCNT+1
1598  004714   000000                   BINWRD: 0
1599
1600
1601                                            ;TRAP DISPATCH SERVICE
1602                                            ;ARGUMENT OF TRAP IS EXTRACTED
1603                                            ;AND USED AS OFFSET TO OBTAIN POINTER
1604                                            ;TO SELECTED SUBROUTINE
1605
1606  004716   011646                  .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
1607  004720   162716   000002                 SUB     #2,(SP)         ;=PC OF TRAP
1608  004724   017616   000000                 MOV     @(SP),(SP)      ;GET TRP
1609  004730   006316                  TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
1610  004732   042716   177001                 BIC     #177001,(SP)    ;CLEAR UNWANTED BITS
1611  004736   062716   001330                 ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
1612  004742   017616   000000                 MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
1613  004746   000136                         JMP     @(SP)+          ;GO TO SUBROUTINE
1614
1615                                            ;ERROR HANDLER
1616                                            ;-------------
1617
1618  004750   004737   007606          .HLT:   JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
1619  004754   032777   010000  174220          BIT     #SW12,@SWR      ;BELL ON ERROR?
1620  004762   001406                         BEQ     XBX             ;BR IF NO BELL
1621  004764   105777   174220                 TSTB    @TPCSR          ;TTY READY.
1622  004770   100003                         BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
1623  004772   112777   000207  174212         MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
1624  005000   032777   020000  174174  XBX:   BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1625  005006   001105                         BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
1626  005010   021637   001234                 CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
1627  005014   001404                         BEQ     1$              ;BR IF YES
1628  005016   011637   001234                 MOV     (SP),LSTERR     ;RECORD BEING HERE
1629  005022   105037   001325                 CLRB    ERRFLG          ;PREPARE HEADER
1630  005026   104406                  1$:     SAV05                   ;SAVE ALL PROC REGISTERS
1631  005030   011605                         MOV     (SP),R5         ;GET THE PC OF ERROR
1632  005032   162705   000002                 SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1633  005036   011504                         MOV     (R5),R4         ;GET HLT INSTRUCTION
1634  005040   006304                         ASL     R4              ;MULT BY TWO
1635  005042   061504                         ADD     (R5),R4         ;DOUBLE IT
1636  005044   006304                         ASL     R4              ;MULT AGAIN
1637  005046   042704   177001                 BIC     #177001,R4      ;CLEAR JUNK
1638  005052   062704   023404                 ADD     #.ERRTAB,R4     ;GET POINTER
1639  005056   012437   005172                 MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1640  005062   012437   005204                 MOV     (R4)+,DATAHD    ;GET DATA HEADRER
1641  005066   011437   005216                 MOV     (R4),DATABP     ;GET DATA TABLE
1642  005072   105737   001325                 TSTB    ERRFLG          ;TYPE HEADREER
1643  005076   001403                         BEQ     TYPMSG          ;BR IF YES
1644  005100   005737   005216                 TST     DATABP          ;DOES DATA TABLE EXIST?
1645  005104   001040                         BNE     TYPDAT          ;BR IF YES.
1646  005106   104402   005672          TYPMSG: TYPE    ,MCRLF
1647  005112   104402   005672                 TYPE    ,MCRLF
1648  005116   005737   001220                 TST     LOCK
1649  005122   001402                         BEQ     1$
1650  005124   104402   006142                 TYPE    ,MASTEK
1651  005130   104402   006130          1$:    TYPE    ,MTSTN
1652  005134   104411   005330                 CNVRT   ,XTSTN          ;SHOW IT
```

```
1653  005140  104402  006217            TYPE    ,MERRPC         ;TYPE PC.
1654  005144  104411  005322            CNVRT   ,ERTABO         ;SHOW IT
1655  005150  104402  005672            TYPE    ,MCRLF          ;GIVE A CR/LF
1656  005154  112737  177777  001325    MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1657  005162  005737  005172            TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1658  005166  001402                    BEQ     WRKO.FM         ;BR IF NO.
1659  005170  104402                    TYPE                    ;TYPE
1660  005172  000000            ERRMSG: 0                       ;    ERROR MESSAGE
1661  005174                    WRKO.FM:                        ;
1662  005174  005737  005204            TST     DATAHD          ;DATA HEADER?
1663  005200  001402                    BEQ     TYPDAT          ;BR IF NO
1664  005202  104402                    TYPE                    ;TYPE
1665  005204  000000            DATAHD: 0                       ;    DATA HEADER
1666  005206  005737  005216    TYPDAT: TST     DATABP          ;DATA TABLE?
1667  005212  001402                    BEQ     RESREG          ;BR IF NO.
1668  005214  104410                    CONVRT                  ;SHOW
1669  005216  000000            DATABP: 0                       ;    DATA TABLE
1670  005220  104407            RESREG: RES05                   ;RESTORE PROC REGISTERS
1671  005222  022737  003522  000042  HALTS:  CMP     #$ENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, HALT!!
1672  005230  001403                    BEQ     1$
1673  005232  005777  173744            TST     @SWR            ;HALT ON ERROR?
1674  005236  100005                    BPL     EXITER          ;BR IF NO HALT ON ERROR
1675  005240  010046            1$:     PUSHR0                  ;SAVE R0
1676  005242  016600  000002            MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
1677  005246  000000                    HALT                    ;HALT
1678  005250  012600                    POPR0                   ;GET R0
1679  005252  005237  001232    EXITER: INC     ERRCNT          ;UPDATE ERROR COUNT
1680  005256  032777  000400  173716    BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1681  005264  001007                    BNE     1$              ;BR IF YES
1682  005266  032777  002000  173706    BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1683  005274  001411                    BEQ     2$              ;BR IF NO
1684  005276  013737  001216  001214    MOV     NEXT,RETURN     ;SET FOR NEXT TEST
1685  005304  012706  001200    1$:     MOV     #STACK,SP       ;RESET SP
1686  005310  013701  001404            MOV     DMCSR,R1        ;SET UP R1
1687  005314  000177  173674            JMP     @RETURN         ;GOTO SPECIFIED TEST
1688  005320  000002            2$:     RTI                     ;RETURN
1689  005322  000001            ERTABO: 1
1690  005324     006     002            .BYTE   6,2
1691  005326  001276                    SAVPC
1692  005330  000001            XTSTN:  1
1693  005332     003     002            .BYTE   3,2
1694  005334  001226                    TSTNO
1695                                    ;ENTER HERE ON POWER FAILURE
1696                                    ;--------------------------
1697
1698
1699  005336                    .PFAIL:
1700  005336  012737  005350  000024            MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
1701  005344  000000                            HALT                            ;HALT ON POWER DOWN NORMAL
1702  005346  000777                            BR      .
1703
1704                                    ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1705
1706  005350                    RESTAR:
1707  005350  012737  005336  000024            MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
1708  005356  012706  001200                    MOV     #STACK,SP       ;RESET THE STACK POINTER
```

```
 1709  005362  013701  001404                  MOV     DMCSR,R1        ;RESTORE R1
 1710  005366  005037  001416                  CLR     TEMP            ;READY FOR TIMMER
 1711  005372  005237  001416                  INC     TEMP            ;PLUS ONE TO THE TIMER!
 1712  005376  001375                          BNE     .-4             ;BR IF MORE TO GO
 1713  005400  104402  005675                  TYPE    ,MPFAIL         ;TYPE THE MESSAGE
 1714  005404  104411  005430                  CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
 1715  005410  105037  001325                  CLRB    ERRFLG          ;START CLEAN
 1716  005414  005037  001234                  CLR     LSTERR          ;'''''''''''''
 1717  005420  005011                          CLR     (R1)            ;CLEAR MAINT BITS
 1718  005422  104412                          MSTCLR                  ;START CLEAN UP OF DEVICE
 1719  005424  000177  173564                  JMP     @RETURN         ;START DOING THAT TEST AGAIN.
 1720  005430  000001                  PFTAB:  1
 1721  005432     003     002          .BYTE   3,2
 1722  005434  001226                          TSTNO
 1723
 1724  005436                          .DELAY:
 1725  005436  012777  000020  173746          MOV     #20,@DMP04
 1726  005444  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1727  005446  121111                          121111                  ;POKE CLOCK DELAY BIT
 1728  005450                          1$:
 1729  005450  104414                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1730  005452  121224                          121224                  ;PORT4_IBUS*11
 1731  005454  032777  000020  173730          BIT     #BIT4,@DMP04    ;IS CLOCK BIT SET?
 1732  005462  001772                          BEQ     1$              ;BR IF NO
 1733  005464  000002                          RTI
 1734
 1735  005466                          .MSTCLR:
 1736  005466  152777  000100  173712          BISB    #BIT6,@DMCSRH   ;SET MASTER CLEAR
 1737  005474  142777  000300  173704          BICB    #BIT6!BIT7,@DMCSRH  ;CLEAR MASTER CLEAR AND RUN
 1738  005502  000002                          RTI                     ;RETURN
 1739
 1740  005504                          .ROMCLK:
 1741  005504  152777  000002  173674          BISB    #BIT1,@DMCSRH   ;SET ROMI
 1742  005512  013677  173676                  MOV     @(SP)+,@DMP06   ;LOAD INSTRUCTION IN SEL6
 1743  005516  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1744  005522  032777  000100  173452          BIT     #SW06,@SWR      ;HALT IF SW06 =1
 1745  005530  001401                          BEQ     1$              ;BR IF SW06 =0
 1746  005532  000000                          HALT                    ;HALT BEFORE CLOCKING INSTRUCTION
 1747  005534  152777  000003  173644  1$:     BISB    #BIT1!BIT0,@DMCSRH ;CLOCK INSTRUCTION
 1748  005542  142777  000007  173636          BICB    #BIT2!BIT1!BIT0,@DMCSRH  ;CLEAR ROMO, ROMI, STEP
 1749  005550  000002                          RTI
 1750
 1751  005552                          .DATACLK:
 1752  005552  013637  001416                  MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
 1753  005556  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1754  005562  152777  000020  173616  1$:     BISB    #BIT4,@DMCSRH   ;SET STEP LU
 1755  005570  027777  173610  173606          CMP     @DMCSR,@DMCSR   ;WASTE TIME
 1756  005576  142777  000020  173602          BICB    #BIT4,@DMCSRH   ;CLEAR STEP LU
 1757  005604  005337  001416                  DEC     TEMP            ;DEC TICK COUNT
 1758  005610  001364                          BNE     1$              ;BR IF NOT DONE
 1759  005612  000002                          RTI                     ;RETURN
 1760  005614  000001                  3$:     .BLKW 1
 1761
 1762  005616                          .TIMER:
 1763  005616  013637  001416                  MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
 1764  005622  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
```

```
 1765  005626                                1$:
 1766  005626  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1767  005630  021364                              021364                  ;PORT4 IBUS* REG11
 1768  005632  032777  000002  173552              BIT     #2,@DMP04       ;IS PGM CLOCK BIT CLEAR?
 1769  005640  001772                              BEQ     1$              ;BR IF YES
 1770  005642                                2$:
 1771  005642  104414                              ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1772  005644  021364                              021364                  ;PORT4 IBUS* REG11
 1773  005646  032777  000002  173536              BIT     #2,@DMP04       ;IS PGM CLOCK BIT SET?
 1774  005654  001372                              BNE     2$              ;BR IF YES
 1775  005656  005337  001416                      DEC     TEMP            ;DEC COUNT
 1776  005662  001361                              BNE     1$              ;BR IF NOT DONE
 1777  005664  000002                              RTI                     ;RETURN
 1778
 1779  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 CZDMH  /
  (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
 1780  007456     006      003                 .BYTE   6,3
 1781  007460  001246                          TEMP1
```

```
1782  007462  000006  000003                .BYTE   6,3
1783  007464  001250                        TEMP2
1784  007466  000006  000003                .BYTE   6,3
1785  007470  001252                        TEMP3
1786  007472  000006  000003                .BYTE   6,3
1787  007474  001254                        TEMP4
1788  007476  000006  000002                .BYTE   6,2
1789  007500  001256                        TEMP5
1790                                .EVEN
1791
1792                                      ;BUFFERS FOR INPUT-OUTPUT
1793
1794  007502  000000                INBUF:  0
1795          007544                .=.+40
1796  007544  000000                MDATA:  0
1797          007606                .=.+40
1798
1799
1800                                      ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1801                                      ;REGISTER USING THE CONSOLE TERMINAL
1802                                      ;---------------------------------------
1803
1804  007606  022737  000176  001202  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
1805  007614  001077                        BNE     CKSWR5          ;BR IF NO
1806  007616  105777  171362                TSTB    @TKCSR          ;IS DONE SET?
1807  007622  100003                        BPL     2$              ;GO ON IF NOT SET
1808  007624  012737  177777  003734        MOV     #-1,DONE        ;IF DONE SET, SET FLAG
1809  007632  022777  000007  171346  2$:   CMP     #7,@TKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
1810  007640  001404                        BEQ     1$              ;BR IF YES
1811  007642  022777  000207  171336        CMP     #207,@TKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII)
1812  007650  001061                        BNE     CKSWR5          ;BR IF NO
1813  007652  010246                1$:     MOV     R2,-(SP)        ;STORE R2
1814  007654  010346                        MOV     R3,-(SP)        ;STORE R3
1815  007656  010446                        MOV     R4,-(SP)        ;STORE R4
1816  007660  012737  177777  010016        MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
1817  007666  005002                CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
1818  007670  012704  177777                MOV     #-1,R4          ;SET FLAG TO ALL ONES
1819  007674  104402  007205                TYPE    ,SWMES          ;TYPE "SWR= "
1820  007700  104411                CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
1821  007702  010052                        SOFTSW                  ;OF SOFT SWITCH REGISTER
1822  007704  104402  007215        CKSWR3: TYPE    ,SWMES1         ;TYPE "NEW? "
1823  007710  004737  010020        CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
1824  007714  022703  000015                CMP     #15,R3          ;WAS IT A CR?
1825  007720  001424                        BEQ     5$              ;BR IF YES
1826  007722  022703  000012                CMP     #12,R3          ;WAS IT A LF?
1827  007726  001416                        BEQ     4$              ;BR IF YES
1828  007730  022703  000025                CMP     #25,R3          ;WAS IT CTRL U?
1829  007734  001754                        BEQ     CKSWR1          ;BR IF YES(START OVER)
1830  007736  022703  000007                CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
1831  007742  001762                        BEQ     CKSWR4
1832  007744  005004                        CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
1833  007746  042703  177770                BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1834  007752  006302                        ASL     R2              ;SHIFT R2 3 TIMES
1835  007754  006302                        ASL     R2
1836  007756  006302                        ASL     R2
1837  007760  050302                        BIS     R3,R2           ;ADD LAST DIGIT
```

```
 1838  007762  000752                        BR      CKSWR4          ;GET NEXT CHARACTER
 1839  007764  012766  002002  000006  4$:   MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
 1840  007772  005704                  5$:   TST     R4              ;IS FLAG CLEAR?
 1841  007774  001002                        BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
 1842  007776  010277  171200                MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
 1843  010002  005037  010016          6$:   CLR     SWFLG           ;CLEAR TYPEOUT FLAG
 1844  010006  012604                        MOV     (SP)+,R4        ;RESTORE R4
 1845  010010  012603                        MOV     (SP)+,R3        ;RESTORE R3
 1846  010012  012602                        MOV     (SP)+,R2        ;RESTORE R2
 1847  010014  000207                  CKSWR5: RTS     PC              ;RETURN
 1848
 1849  010016  000000                  SWFLG:  0
 1850
 1851  010020  105777  171160          INCHAR: TSTB    @TKCSR
 1852  010024  100375                        BPL     .-4
 1853  010026  017703  171154                MOV     @TKDBR,R3
 1854  010032  105777  171152                TSTB    @TPCSR
 1855  010036  100375                        BPL     .-4
 1856  010040  010377  171146                MOV     R3,@TPDBR
 1857  010044  042703  000200                BIC     #BIT7,R3
 1858  010050  000207                        RTS     PC
 1859
 1860  010052  000001                  SOFTSW: 1
 1861  010054     006     002                .BYTE   6,2
 1862  010056  000176                        SWREG
```

```
1863
1864                                        ;
1865                                        ;ROUTINE USED TO "CYCLE" THROUGH UP TO 16 DMC11'S
1866                                        ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1867                                        ;AND RUNS THE SPECIFIED DMC11'S.   THIS ROUTINE *MUST*
1868                                        ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1869                                        ;SETUP NECESSARY.
1870                                        ;
1871
1872  010060  005737  001306        CYCLE:  TST     DMACTV          ;ARE ANY DMC11'S TO BE TESTED?
1873  010064  001004                        BNE     1$              ;BR IF OK.
1874  010066  104402  007154                TYPE    ,NOACT          ;NO DMC11'S SELECTED!!
1875  010072  000000                        HALT                    ;STOP THE SHOW.
1876  010074  000776                        BR      .-2             ;DISQUALIFY CONT. SW.
1877  010076  000241                1$:     CLC                     ;CLEAR PROC. CARRY BIT.
1878  010100  006137  001316                ROL     RUN             ;UPDATE POINTER
1879  010104  005537  001316                ADC     RUN             ;CATCH CARRY FROM RUN
1880  010110  062737  000004  001322        ADD     #4,MILK         ;UPDATE POINTER
1881  010116  062737  000010  001320        ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
1882  010124  022737  001700  001320        CMP     #DM.MAP+200,CREAM
1883  010132  001006                        BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
1884  010134  012737  001500  001320        MOV     #DM.MAP,CREAM   ;RESET ADDRESS POINTER.
1885  010142  012737  001702  001322        MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
1886  010150  033737  001316  001306  2$:   BIT     RUN,DMACTV      ;IS THIS ONE ACTIVE?
1887  010156  001747                        BEQ     1$              ;BR IF NO
1888  010160  013700  001320                MOV     CREAM,R0        ;GET ADDRESS POINTER
1889  010164  013702  001322                MOV     MILK,R2         ;GET PASS COUNT POINTER
1890  010170  012037  001404                MOV     (R0)+,DMCSR     ;LOAD SYSTEM CTRL. REG
1891  010174  011037  001374                MOV     (R0),DMRVEC     ;LOAD VECTOR
1892  010200  042737  177000  001374        BIC     #177000,DMRVEC  ;CLEAR UNWANTED BITS
1893  010206  012037  001366                MOV     (R0)+,STAT1     ;LOAD STAT1
1894  010212  012037  001370                MOV     (R0)+,STAT2     ;LOAD STAT2
1895  010216  012037  001372                MOV     (R0)+,STAT3     ;LOAD STAT3
1896  010222  012237  001230                MOV     (R2)+,PASCNT    ;LOAD PASS COUNT
1897  010226  012237  001232                MOV     (R2)+,ERRCNT    ;LOAD ERROR COUNT
1898  010232  012700  000002                MOV     #2,R0           ;SAVE CORE THIS WAY!
1899  010236  013737  001404  001406        MOV     DMCSR,DMCSRH
1900  010244  005237  001406                INC     DMCSRH
1901  010250  013737  001406  001410        MOV     DMCSRH,DMCTL
1902  010256  005237  001410                INC     DMCTL
1903  010262  013737  001410  001412        MOV     DMCTL,DMP04
1904  010270  060037  001412                ADD     R0,DMP04
1905  010274  013737  001412  001414        MOV     DMP04,DMP06
1906  010302  060037  001414                ADD     R0,DMP06
1907
1908  010306  013737  001374  001376        MOV     DMRVEC,DMRLVL   ;PTY LVL
1909  010314  060037  001376                ADD     R0,DMRLVL       ;
1910  010320  013737  001376  001400        MOV     DMRLVL,DMTVEC   ;TX VEC
1911  010326  060037  001400                ADD     R0,DMTVEC       ;
1912  010332  013737  001400  001402        MOV     DMTVEC,DMTLVL   ;TX LVL
1913  010340  060037  001402                ADD     R0,DMTLVL
1914
1915  010344  032737  000002  001236        BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
1916  010352  001450                        BEQ     7$              ;BR IF NO
1917  010354                         4$:
1918  010354  005737  000042                TST     @#42            ;RUNNING IN AUTO MODE?
```

```
1919  010360  001045                        BNE     7$              ;BR IF YES
1920  010362  104402  005672                TYPE    ,MCRLF
1921  010366  104403                        INSTR                   ;GET TEST NO.
1922  010370  006130                        MTSTN
1923  010372  104405                        PARAM
1924  010374  000001                        1
1925  010376  001000                        1000
1926  010400  001226                        TSTNO
1927  010402     000              .BYTE   0
1928  010403     001              .BYTE   1
1929  010404  012700  012320              MOV     #TST1,R0
1930  010410  022710                5$:   CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
1931  010412  012737                      MOV     (PC)+,@(PC)+
1932  010414  001020                      BNE     6$              ;BR IF NOT SAME
1933  010416  023760  001226  000002      CMP     TSTNO,2(R0)     ;DOES TSTNO MATCH?
1934  010424  001014                      BNE     6$              ;BR IF NO
1935  010426  022760  001226  000004      CMP     #TSTNO,4(R0)    ;IS LAST WORD OK?
1936  010434  001010                      BNE     6$              ;BR IF NO
1937  010436  010037  001214              MOV     R0,RETURN       ;IT IS A LEGAL TEST SO DO IT
1938  010442  104402  005755              TYPE    ,MR
1939  010446  042737  000002  001236      BIC     #SW01,STRTSW
1940  010454  000412                      BR      8$
1941  010456  005720                6$:   TST     (R0)+           ;POP R0
1942  010460  020027  016214              CMP     R0,#TLAST+10    ;AT END YET?
1943  010464  001351                      BNE     5$              ;BR IF NO
1944  010466  104402  005666              TYPE    ,MQM            ;YES ILLEGAL TEST NO.
1945  010472  000730                      BR      4$              ;TRY AGAIN
1946
1947  010474  012737  012320  001214  7$: MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
1948  010502  013701  001404      8$:     MOV     DMCSR,R1        ;R1 = BASE DMC11 ADDRESS
1949  010506  000177  170502              JMP     @RETURN         ;GO START TESTING.
1950
1951
1952                                  ;ROUTINE USED TO "AUTO SIZE" THE DMC11
1953                                  ;CSR AND VECTOR.
1954                                  ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
1955                                  ;       ADDRESS RANGE (160000:164000)
1956                                  ;       AND THE VECTOR MAY BE ANY WHERE IN THE
1957                                  ;       FLOATING VECTOR RANGE (300:770)
1958                                  ;
1959
1960  010512                  AUTO.SIZE:
1961  010512  000005                  RESET                   ;INSURE A BUS INIT.
1962  010514  012702  001500  CSRMAP: MOV     #DM.MAP,R2      ;LOAD MAP POINTER.
1963  010520  005022          1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
1964  010522  022702  001700          CMP     #DM.END,R2      ;ALL DONE?
1965  010526  001374                  BNE     1$              ;BR IF NO
1966  010530  005037  001310          CLR     DMNUM           ;SET OCTAL NUMBER OF DMC11'S TO 0
1967  010534  012702  001500          MOV     #DM.MAP,R2      ;R2 POINTS TO DMC MAP
1968  010540  005037  001306          CLR     DMACTV          ;CLEAR ACTIVE
1969  010544  032737  000001  001236  BIT     #SW00,STRTSW    ;QUESTIONS?
1970  010552  001002                  BNE     .+6             ;BR IF YES
1971  010554  000137  011252          JMP     7$              ;IF NO SKIP QUESTIONS
1972  010560  012737  000001  001256  MOV     #1,TEMP5        ;START WITH 1
1973  010566  104403                  INSTR
1974  010570  006450                  NUM
```

```
 1975  010572  104405                          PARAM
 1976  010574  000001                          1
 1977  010576  000020                          16.
 1978  010600  001252                          TEMP3
 1979  010602     000                          .BYTE   0
 1980  010603     001                          .BYTE   1
 1981  010604  013737  001252  001310          MOV     TEMP3,DMNUM     ;DMNUM = HOW MANY
 1982  010612  104402  005672        12$:      TYPE    ,MCRLF
 1983  010616  104410                          CONVRT                  ;TYPE WHICH DMC IS BEING DONE
 1984  010620  012002                          WHICH                   ;TEMP5 IS WHICH DMC
 1985  010622  005237  001256                  INC     TEMP5
 1986  010626  104403                          INSTR
 1987  010630  006510                          CSR
 1988  010632  104405                          PARAM
 1989  010634  160000                          160000
 1990  010636  164000                          164000
 1991  010640  001254                          TEMP4
 1992  010642     000                          .BYTE   0
 1993  010643     001                          .BYTE   1
 1994  010644  013722  001254                  MOV     TEMP4,(R2)+     ;STORE CSR IN MAP
 1995  010650  104403                          INSTR
 1996  010652  006526                          VEC
 1997  010654  104405                          PARAM
 1998  010656  000000                          0
 1999  010660  000776                          776
 2000  010662  001254                          TEMP4
 2001  010664     000                          .BYTE   0
 2002  010665     001                          .BYTE   1
 2003  010666  013712  001254                  MOV     TEMP4,(R2)      ;STORE VECTOR IN MAP
 2004  010672  104402              10$:        TYPE
 2005  010674  006547                          PRIO                    ;ASK WHAT BR LEVEL
 2006  010676  004737  012266                  JSR     PC,INTTY        ;GET RESPONSE
 2007  010702  022703  000024                  CMP     #24,R3          ;
 2008  010706  101014                          BHI     50$             ;BR IF LESS THAN 4
 2009  010710  022703  000027                  CMP     #27,R3          ;
 2010  010714  103411                          BLO     50$             ;BR IF GREATER THAN 7
 2011  010716  012704  000011                  MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
 2012  010722  006303                          ASL     R3              ;SHIFT R3 LEFT
 2013  010724  005304                          DEC     R4              ;DEC SHIFT COUNT
 2014  010726  001375                          BNE     .-4             ;BR IF NOT DONE
 2015  010730  042703  170777                  BIC     #170777,R3      ;BIC UNWANTED BITS
 2016  010734  050312                          BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
 2017  010736  000403                          BR      8$              ;CONTINUE
 2018  010740  104402              50$:        TYPE
 2019  010742  005666                          MQM                     ;RESPONSE IS OUT OF LIMITS
 2020  010744  000752                          BR      10$             ;TRY AGAIN
 2021  010746  104402              8$:         TYPE
 2022  010750  006606                          CRAM                    ;DOES DMC HAVE CRAM?
 2023  010752  004737  012266                  JSR     PC,INTTY        ;GET REPLY
 2024  010756  022703  000131                  CMP     #131,R3
 2025  010762  001427                          BEQ     9$              ;YES
 2026  010764  022703  000116                  CMP     #116,R3         ;NO
 2027  010770  001403                          BEQ     40$             ;NOT A Y OR N
 2028  010772  104402                          TYPE
 2029  010774  005666                          MQM                     ;TYPE "?"
 2030  010776  000763                          BR      8$              ;ASK AGAIN
```

```
2031  011000  104402                    40$:    TYPE
2032  011002  007340                            SPEED                   ;DMC11-AR OR DMC11-AL?
2033  011004  004737  012266                    JSR     PC,INTTY        ;GET RESPONSE
2034  011010  022703  000122                    CMP     #122,R3         ;IS IT R
2035  011014  001414                            BEQ     16$             ;BR IF REMOTE
2036  011016  022703  000114                    CMP     #114,R3         ;IS IT L
2037  011022  001403                            BEQ     41$             ;BR IF LOCAL
2038  011024  104402                            TYPE
2039  011026  005666                            MQM
2040  011030  000763                            BR      40$             ;TRY AGAIN
2041  011032  052762  000002  000004  41$:     BIS     #BIT1,4(R2)     ;SET BIT1 IN STAT3
2042  011040  000402                            BR      16$             ;CONTINUE
2043  011042  052712  100000            9$:     BIS     #BIT15,(R2)     ;SET BIT 15 IF CRAM
2044  011046  104402                    16$:    TYPE
2045  011050  006704                            MODU                    ;ASK WHICH LINE UNIT
2046  011052  004737  012266                    JSR     PC,INTTY        ;GET REPLY
2047  011056  022703  000021                    CMP     #21,R3          ;''1''
2048  011062  001417                            BEQ     30$
2049  011064  022703  000022                    CMP     #22,R3          ;''2''
2050  011070  001412                            BEQ     31$
2051  011072  022703  000116                    CMP     #116,R3         ;''N''
2052  011076  001403                            BEQ     32$
2053  011100  104402                            TYPE
2054  011102  005666                            MQM                     ;IF NOT A 1,2 OR N TYPE ''?''
2055  011104  000760                            BR      16$             ;TRY AGIAN
2056  011106  052722  010000            32$:    BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
2057  011112  022222                            CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
2058  011114  000447                            BR      33$
2059  011116  052712  020000            31$:    BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF M8202
2060  011122  104402                    30$:    TYPE
2061  011124  007114                            CONN                    ;ASK IF LOOP-BACK IS ON
2062  011126  004737  012266                    JSR     PC,INTTY        ;GET REPLY
2063  011132  022703  000131                    CMP     #131,R3         ;Y
2064  011136  001406                            BEQ     17$
2065  011140  022703  000116                    CMP     #116,R3         ;N
2066  011144  001406                            BEQ     18$
2067  011146  104402                            TYPE
2068  011150  005666                            MQM                     ;IF NOT Y OR N TYPE ''?''
2069  011152  000763                            BR      30$             ;TRY AGAIN
2070  011154  052722  040000            17$:    BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
2071  011160  000402                            BR      19$
2072  011162  042722  040000            18$:    BIC     #BIT14,(R2)+    ;NO TURNAROUND
2073  011166                            19$:
2074  011166  104403                            INSTR
2075  011170  007016                            LINE
2076  011172  104405                            PARAM
2077  011174  000000                            0
2078  011176  000377                            377
2079  011200  001254                            TEMP4
2080  011202     000                            .BYTE   0
2081  011203     001                            .BYTE   1
2082  011204  113722  001254                    MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
2083  011210  104403                            INSTR
2084  011212  007054                            BM
2085  011214  104405                            PARAM
2086  011216  000000                            0
```

```
2087  011220  000377                        377
2088  011222  001254                        TEMP4
2089  011224     000                        .BYTE   0
2090  011225     001                        .BYTE   1
2091  011226  113722  001254                MOVB    TEMP4,(R2)+     ;STORE SWITCH PAC IN MAP
2092  011232  005722                        TST     (R2)+           ;POP OVER STAT3
2093  011234  005337  001252          33$:  DEC     TEMP3           ;DEC DMC COUNT
2094  011240  001402                        BEQ     34$             ;BR IF DONE
2095  011242  000137  010612                JMP     12$             ;JUMP IF NOT
2096  011246  000137  011702          34$:  JMP     13$             ;CONTINUE
2097  011252  012701  160000          7$:   MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
2098  011256  012737  011774  000004        MOV     #6$,@#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
2099  011264  005011                  2$:   CLR     (R1)            ;CLEAR SEL0
2100  011266  005711                        TST     (R1)            ;IF DMC11 DMCSR S/B 0
2101  011270  001172                        BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DMC11
2102  011272  005061  000006                CLR     6(R1)           ;CLEAR SEL6
2103  011276  005761  000006                TST     6(R1)           ;IF DMC11 THEN DMRIC S/B =0!
2104  011302  001165                        BNE     3$              ;BR IF NOT DMC11
2105  011304  012711  002000                MOV     #BIT10,(R1)     ;SET ROMO
2106  011310  005061  000004                CLR     4(R1)           ;CLEAR SEL4
2107  011314  012761  125252  000006        MOV     #125252,6(R1)   ;WRITE THIS TO SEL6
2108  011322  052711  020000                BIS     #BIT13,(R1)     ;WRITE IT!
2109  011326  022761  125252  000004        CMP     #125252,4(R1)   ;WAS IT WRITTEN?
2110  011334  001004                        BNE     21$             ;IF NO IT IS NOT CRAM
2111  011336  052762  100000  000002        BIS     #BIT15,2(R2)    ;SET BIT15 IF CRAM
2112  011344  000431                        BR      22$
2113  011346  012711  001000          21$:  MOV     #BIT9,(R1)      ;SET ROMI
2114  011352  012761  100430  000006        MOV     #100430,6(R1)   ;PUT INSTRUCTION IN SEL6
2115  011360  012711  001400                MOV     #BIT9!BIT8,(R1) ;CLOCK INSTRUCTION (MICRO PROC PC TO 0)
2116  011364  012711  002000                MOV     #BIT10,(R1)     ;SET ROMO
2117  011370  022761  016472  000006        CMP     #016472,6(R1)   ;IS IT LOCAL CROM?
2118  011376  001411                        BEQ     23$             ;BR IF YES
2119  011400  022761  016461  000006        CMP     #016461,6(R1)   ;IS IT REMOTE CROM?
2120  011406  001410                        BEQ     22$             ;BR IF YES
2121  011410  022761  177777  000006        CMP     #-1,6(R1)       ;NO CROM?
2122  011416  001404                        BEQ     22$             ;BR IF YES
2123  011420  000516                        BR      3$              ;NOT A DMC
2124  011422  052762  000002  000006  23$:  BIS     #BIT1,6(R2)     ;SET BIT 1 IN STAT3
2125                                  ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DMC11 CSR ADDRESS.
2126  011430  010122                  22$:  MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
2127  011432  012711  001000          15$:  MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
2128  011436  005061  000004                CLR     4(R1)           ;CLEAR PORT4
2129  011442  012761  122113  000006        MOV     #122113,6(R1)   ;LOAD INSTRUCTION (CLR DTR)
2130  011450  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
2131  011454  012761  021264  000006        MOV     #021264,6(R1)   ;LOAD INSTRUCTION
2132  011462  052711  000400                BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
2133  011466  122761  000377  000004        CMPB    #377,4(R1)      ;IS IT ALL ONES?
2134  011474  001003                        BNE     .+10            ;BR IF NO
2135  011476  052712  010000                BIS     #BIT12,(R2)     ;IF YES, NO LINE UNIT, SET STATUS BIT
2136  011502  000436                        BR      20$
2137  011504  032761  000002  000004        BIT     #BIT1,4(R1)     ;IS SWITCH A ONE?
2138  011512  001403                        BEQ     .+10            ;BR IF M8201
2139  011514  052712  060000                BIS     #BIT13!BIT14,(R2) ;M8202 ASSUME CONNECTOR
2140  011520  000427                        BR      20$             ;CONNECTOR ON)
2141  011522  032761  000010  000004        BIT     #BIT3,4(R1)     ;IS MRDY SET
2142  011530  001023                        BNE     20$             ;BR IF M8201 NO CONNECTOR (ON LINE)
```

```
2143  011532  012761  000100  000004          MOV     #BIT6,4(R1)      ;LOAD PORT4
2144  011540  012761  122113  000006          MOV     #122113,6(R1)    ;LOAD INSTRUCTION
2145  011546  052711  000400                  BIS     #BIT8,(R1)       ;CLOCK INSTRUCTION(SET DTR)
2146  011552  012761  021264  000006          MOV     #021264,6(R1)    ;LOAD INSTRUCTION
2147  011560  052711  000400                  BIS     #BIT8,(R1)       ;CLOCK INSTRUCTION(READ MODEM REG)
2148  011564  032761  000010  000004          BIT     #BIT3,4(R1)      ;IS MRDY SET NOW?
2149  011572  001402                          BEQ     20$              ;BR IF NO CONNECTOR
2150  011574  052712  040000                  BIS     #BIT14,(R2)      ;SET STATUS BIT FOR CONNECTOR
2151  011600  005722                  20$:    TST     (R2)+            ;POP POINTER
2152  011602  012761  021324  000006          MOV     #021324,6(R1)    ;PUT INSTRUCTION IN PORT6
2153  011610  012711  001400                  MOV     #BIT9!BIT8,(R1)  ;PORT4_LU 15
2154  011614  156122  000004                  BISB    4(R1),(R2)+      ;STORE DDCMP LINE # IN TABLE
2155  011620  012761  021344  000006          MOV     #021344,6(R1)    ;PORT6_INSTRUCTION
2156  011626  012711  001400                  MOV     #BIT8!BIT9,(R1)  ;CLOCK INSTR.
2157  011632  156122  000004                  BISB    4(R1),(R2)+      ;STORE BM873 ADD IN TABLE
2158  011636  005722                          TST     (R2)+            ;POP OVER STAT3
2159  011640  005011                          CLR     (R1)             ;CLEAR ROMI
2160  011642  005237  001310                  INC     DMNUM            ;UPDATE DEVICE COUNTER
2161  011646  022737  000020  001310          CMP     #20,DMNUM        ;ARE MAX. NO. OF DEV FOUND?
2162  011654  001412                          BEQ     13$              ;YES DON'T LOOK FOR ANY MORE.
2163  011656  005011                  3$:     CLR     (R1)             ;CLEAR BIT 10
2164  011660  005061  000006                  CLR     6(R1)            ;CLEAR SEL 6
2165  011664  062701  000010          14$:    ADD     #10,R1           ;UPDATE CSR POINTER ADDRESS
2166  011670  022701  164000                  CMP     #164000,R1
2167  011674  001402                          BEQ     13$              ;BR IF DONE
2168  011676  000137  011264                  JMP     2$               ;JUMP IF NOT
2169  011702  005037  001306          13$:    CLR     DMACTV
2170  011706  005737  001310                  TST     DMNUM            ;WERE ANY DMC11'S FOUND AT ALL?
2171  011712  001423                          BEQ     5$               ;ERROR AUTO SIZER FOUND NO DMC11'S IN THIS SYS.
2172  011714  013701  001310                  MOV     DMNUM,R1
2173  011720  010137  001314                  MOV     R1,SAVNUM        ;SAVE NUMBER OF DEVICES
2174  011724  000241                  4$:     CLC
2175  011726  006137  001306                  ROL     DMACTV           ;GENERATE ACTIVE REGISTER OF DEVICES.
2176  011732  005237  001306                  INC     DMACTV           ;SET THE BIT
2177  011736  005301                          DEC     R1
2178  011740  001371                          BNE     4$               ;BR IF MORE TO GENERATE
2179  011742  012737  000006  000004          MOV     #6,@#4           ;RESTORE TRAP VECTOR
2180  011750  013737  001306  001312          MOV     DMACTV,SAVACT    ;SAVE ACTIVE REGISTER
2181  011756  000137  012010                  JMP     VECMAP           ;GO FIND THE VECTOR NOW.
2182  011762  104402  005760          5$:     TYPE    ,MERR2           ;NOTIFY OPR THAT NO DMC11'S FOUND.
2183  011766  005000                          CLR     R0               ;MAKE DATA LIGHTS ZERO
2184  011770  000000                          HALT                     ;STOP THE SHOW
2185  011772  000776                          BR      .-2              ;DISABLE CONT. SW.
2186  011774  012716  011664          6$:     MOV     #14$,(SP)        ;ENTERED BY NON-EXISTANT TIME-OUT.
2187  012000  000002                          RTI                      ;RETURN TO MAINSTREAM
2188
2189  012002  000001                  WHICH:  1
2190  012004     002     002                  .BYTE   2,2
2191  012006  001256                          TEMP5
2192
2193  012010  032737  000001  001236  VECMAP: BIT     #SW00,STRTSW
2194  012016  001114                          BNE     5$
2195  012020  012737  000340  000022          MOV     #340,@#22        ;SET IOT TRAP PRIO TO 7
2196  012026  012737  012202  000020          MOV     #4$,@#20         ;SET IOT TRAP VECTOR
2197  012034  012702  001500                  MOV     #DM.MAP,R2       ;SET SOFTWARE POINTER
2198  012040  012700  000300                  MOV     #300,R0          ;FLOATING VECTORS START HERE.
```

```
2199  012044  012701  000302              MOV     #302,R1          ;PC OF IOT INSTR.
2200  012050  010120                1$:   MOV     R1,(R0)+         ;START FILLING VECTOR AREA
2201  012052  012721  000004              MOV     #4,(R1)+         ;WITH .+2; IOT
2202  012056  022021                      CMP     (R0)+,(R1)+      ;ADD 2 TO R0 +R1
2203  012060  020127  001000              CMP     R1,#1000
2204  012064  101771                      BLOS    1$               ;BR IF MORE TO FILL
2205  012066  013737  001306  001246      MOV     DMACTV,TEMP1     ;STORE TEMPORALLY
2206  012074  006037  001246        2$:   ROR     TEMP1            ;BRING OUT A BIT
2207  012100  103063                      BCC     5$               ;BR IF ALL DONE
2208  012102  012704  000012              MOV     #12,R4           ;R4 IS INDEX REGISTER
2209  012106  016437  012252  177776      MOV     BRLVL(R4),PS     ;SET PS TO 7
2210  012114  011201                      MOV     (R2),R1
2211  012116  012761  000200  000004      MOV     #200,4(R1)
2212  012124  012711  001000              MOV     #BIT9,(R1)       ;SET ROMI
2213  012130  012761  121111  000006      MOV     #121111,6(R1)    ;PUT INSTRUCTION IN PORT6
2214  012136  012711  001400              MOV     #BIT9!BIT8,(R1)  ;FORCE AN INTERRUPT
2215  012142  105200                7$:   INCB    R0               ;STALL
2216  012144  001376                      BNE     .-2              ;FOR TIME TO INTERUPT
2217  012146  162704  000002              SUB     #2,R4            ;GET NEXT LOWEST PS LEVEL
2218  012152  001404                      BEQ     6$               ;BR IF R4 = 0
2219  012154  016437  012252  177776      MOV     BRLVL(R4),PS     ;MOVE NEXT LOWER LEVEL IN PS
2220  012162  000767                      BR      7$               ;BR TO DELAY
2221  012164  052762  005300  000002  6$: BIS     #5300,2(R2)      ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX DMC11 LATER
2222  012172  005011                3$:   CLR     (R1)             ;CLEAR ROMI
2223  012174  062702  000010              ADD     #10,R2           ;POP SOFTWARE POINTER
2224  012200  000735                      BR      2$               ;KEEP GOING
2225  012202  051662  000002        4$:   BIS     (SP),2(R2)       ;GET VECTOR ADDRESS
2226  012206  042762  000007  000002      BIC     #7,2(R2)         ;CLEAR JUNK
2227  012214  016405  012254              MOV     BRLVL+2(R4),R5   ;GET BR LEVEL OF DMC11
2228  012220  006305                      ASL     R5               ;SHIFT LEVEL 4 PLACES
2229  012222  006305                      ASL     R5               ;TO THE LEFT FOR THE
2230  012224  006305                      ASL     R5               ;STATUS TABLE
2231  012226  006305                      ASL     R5
2232  012230  042705  170777              BIC     #170777,R5       ;CLEAR UNWANTED BITS
2233  012234  050562  000002              BIS     R5,2(R2)         ;PUT BR LEVEL IN STATUS TABLE
2234  012240  022626                      CMP     (SP)+,(SP)+      ;POP IOT JUNK OFF STACK
2235  012242  012716  012172              MOV     #3$,(SP)         ;SET FOR RETURN
2236  012246  000002                      RTI
2237  012250  000207                5$:   RTS     PC               ;ALL DONE WITH "AUTO SIZING"
2238
2239  012252  000000                BRLVL:  0       ;LEVEL 0
2240  012254  000000                        0       ;LEVEL 0
2241  012256  000200                        200     ;LEVEL 4
2242  012260  000240                        240     ;LEVEL 5
2243  012262  000300                        300     ;LEVEL 6
2244  012264  000340                        340     ;LEVEL 7
2245
2246
2247  012266  105777  166712        INTTY:  TSTB    @TKCSR           ;WAIT FOR DONE
2248  012272  100375                        BPL     .-4
2249  012274  017703  166706              MOV     @TKDBR,R3        ;PUT CHAR IN R3
2250  012300  105777  166704              TSTB    @TPCSR           ;WAIT UNTIL PRINTER IS READY
2251  012304  100375                      BPL     .-4
2252  012306  010377  166700              MOV     R3,@TPDBR        ;ECHO CHAR
2253  012312  042703  000240              BIC     #BIT7!BIT5,R3    ;MASK OFF LOWER CASE
2254  012316  000207                      RTS     PC               ;RETURN
```

```
2255
```

```
2256
2257
2258                                    ;*************************** TEST 1 ****************************
2259                                    ;*FREE RUNNING FLAG MODE DATA TEST
2260                                    ;*TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA
2261                                    ;*LINE UNIT LOOP IS SET FOR THIS TEST.
2262                                    ;*ALL FOLLOWING TESTS ARE FREE RUNNING AND ARE PERFORMED
2263                                    ;*ONLY ON DMC'S WITH LINE UNITS. IF YOU WISH TO PERFORM
2264                                    ;*THESE FREE RUNNING TESTS ON A KMC (NORMALLY THE FREE RUNNING TESTS
2265                                    ;*WILL FAIL ON A KMC, THE TIMER IS TOO FAST) THEN YOU MUST
2266                                    ;*MANUALLY SET BIT0 OF STAT3 IN THE STATUS MAP. ALSO THE KMC
2267                                    ;*MUST HAVE THE MICRO-CODE LOADED BY PREVIOUSLY RUNNING
2268                                    ;*DZDMG TEST 2 AND THEN LOADING AND STARTING CZDMHC
2269                                    ;* WITH SWITCH 7 = 1
2270                                    ;;*************************************************************************
2271
2272                                    ;  TEST 1
2273                                    ;--------------
2274  012320  012737  000001  001226  TST1:  MOV     #1,TSTNO
2275  012326  012737  013404  001216          MOV     #TST2,NEXT
2276                                                          ;R1 CONTAINS BASE DMC11 ADDRESS
2277  012334  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT A DMC?
2278  012342  001406                          BEQ     .+16            ;BR IF YES
2279  012344  032737  000001  001372          BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2280  012352  001002                          BNE     .+6             ;BR IF YES
2281  012354  000137  013402                  JMP     14$             ;SKIP TEST
2282  012360  032737  010000  001366          BIT     #BIT12,STAT1    ;LU PRESENT?
2283  012366  001372                          BNE     .-12            ;BR IF NO
2284  012370  013700  021360                  MOV     RCOUNT,R0       ;CLEAR RECEIVER BUFFER
2285  012374  062700  000002                  ADD     #2,R0           ;CLEAR 2 MORE LOCATIONS
2286  012400  012702  021362                  MOV     #RBUF,R2        ;CLEAR OUT RECEIVE BUFFER
2287  012404  105022                  10$:    CLRB    (R2)+           ;CLEAR BUFFER
2288  012406  005300                          DEC     R0              ;DONE YET!
2289  012410  001375                          BNE     10$             ;NO
2290  012412  005037  021306                  CLR     TFLAG           ;SET TFLAG TO 0
2291  012416  005037  021310                  CLR     RFLAG           ;SET RFLAG TO 0
2292  012422  012711  040000                  MOV     #BIT14,(R1)     ;MASTER CLEAR
2293  012426  032737  100000  001366          BIT     #BIT15,STAT1    ;CRAM?
2294  012434  001402                          BEQ     .+6             ;BR IF NO
2295  012436  012711  100000                  MOV     #BIT15,(R1)     ;IF CRAM SET RUN
2296  012442  105227  000000                  INCB    #0              ;DELAY
2297  012446  001375                          BNE     .-4             ;DELAY
2298  012450  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2299  012454  005711                  1$:     TST     (R1)            ;RUN SET?
2300  012456  100405                          BMI     .+14            ;BR IF YES
2301  012460  005237  001416                  INC     TEMP            ;INC DELAY
2302  012464  001373                          BNE     1$              ;BR IF NOT DONE
2303  012466  104014                          HLT     14              ;ERROR RUN NOT SET
2304  012470  000771                          BR      1$              ;TRY AGAIN
2305  012472  052711  004043                  BIS     #4043,(R1)      ;BASE I, LU LOOP
2306  012476  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2307  012502  105711                  2$:     TSTB    (R1)            ;RDI SET?
2308  012504  100404                          BMI     .+12            ;BR IF YES
2309  012506  005237  001416                  INC     TEMP            ;INC DELAY
2310  012512  001373                          BNE     2$              ;BR IF NOT DONE
2311  012514  104014                          HLT     14              ;ERROR,RDI NOT SET
```

```
2312  012516  012761  021430  000004          MOV     #BASE,4(R1)     ;SET UP BASE ADDRESS
2313  012524  005061  000006                  CLR     6(R1)           ;CLEAR COUNT
2314  012530  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
2315  012534  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2316  012540  105711                  3$:     TSTB    (R1)            ;IS RDI GONE?
2317  012542  100020                          BPL     8$              ;BR IF YES
2318  012544  005237  001416                  INC     TEMP            ;INC DELAY
2319  012550  001373                          BNE     3$              ;BR IF NOT DONE
2320  012552  105761  000002                  TSTB    2(R1)           ;IS THERE A CNTL O ERROR
2321  012556  100011                          BPL     18$             ;BR IF NO
2322  012560  016137  000004  001252          MOV     4(R1),TEMP3     ;SAVE SEL4 FOR TYPEOUT
2323  012566  016137  000006  001254          MOV     6(R1),TEMP4     ;SAVE SEL6 FOR TYPEOUT
2324  012574  104016                          HLT     16              ;CNTL O ERROR
2325  012576  000137  013402                  JMP     14$             ;FATAL ERROR STOP
2326  012602  104014                  18$:    HLT     14              ;ERROR RDI STILL SET
2327  012604                          8$:
2328  012604  152711  000041                  BISB    #41,(R1)        ;ASK FOR CNTL I
2329  012610  105711                  64$:    TSTB    (R1)            ;WAIT FOR RDI
2330  012612  100376                          BPL     64$             ;BR IF NOT SETY
2331  012614  005061  000006                  CLR     6(R1)           ;SET FULL DUPLEX
2332  012620  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
2333  012624  105711                  65$:    TSTB    (R1)            ;RDI UP?
2334  012626  100776                          BMI     65$             ;BR IF YES
2335  012630  152711  000044                  BISB    #44,(R1)        ;REC BA/CC
2336  012634  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2337  012640  105711                  4$:     TSTB    (R1)            ;IS RDI SET?
2338  012642  100404                          BMI     .+12            ;BR IF YES
2339  012644  005237  001416                  INC     TEMP            ;INC DELAY
2340  012650  001373                          BNE     4$              ;BR IF DELAY NOT DONE
2341  012652  104014                          HLT     14              ;ERROR RDI NOT SET
2342  012654  012761  021362  000004          MOV     #RBUF,4(R1)     ;LOAD REC BA
2343  012662  013761  021360  000006          MOV     RCOUNT,6(R1)    ;LOAD REC COUNT
2344  012670  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
2345  012674  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2346  012700  105711                  5$:     TSTB    (R1)            ;RDI GONE?
2347  012702  100004                          BPL     .+12            ;BR IF YES
2348  012704  005237  001416                  INC     TEMP            ;INC DELAY
2349  012710  001373                          BNE     5$              ;BR IF NO DONE
2350  012712  104014                          HLT     14              ;ERROR RDI STILL SET
2351  012714  152711  000040                  BISB    #40,(R1)        ;XMIT BA/CC
2352  012720  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2353  012724  105711                  6$:     TSTB    (R1)            ;RDI SET?
2354  012726  100404                          BMI     .+12            ;BR IF YES
2355  012730  005237  001416                  INC     TEMP            ;INC DELAY
2356  012734  001373                          BNE     6$              ;BR IF NOT DONE
2357  012736  104014                          HLT     14              ;ERROR RDI NOT SET
2358  012740  012761  021314  000004          MOV     #TBUF,4(R1)     ;LOAD XMIT BUFFER
2359  012746  013761  021312  000006          MOV     TCOUNT,6(R1)    ;LOAD COUNT
2360  012754  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
2361  012760  005037  001416                  CLR     TEMP            ;GET SET TO DELAY
2362  012764  105711                  7$:     TSTB    (R1)            ;RDI GONE?
2363  012766  100004                          BPL     .+12            ;BR IF YES
2364  012770  005237  001416                  INC     TEMP            ;INC DELAY
2365  012774  001373                          BNE     7$              ;BR IF NOT DONE DELAY
2366  012776  104014                          HLT     14              ;ERROR RDI STILL SET
2367  013000  005037  001416          16$:    CLR     TEMP            ;GET SET TO DELAY
```

```
2368  013004  012737  000022  001246          MOV     #22,TEMP1       ;GET SET FOR LONG DELAY
2369  013012  105761  000002          11$:     TSTB    2(R1)           ;RDO SET?
2370  013016  100407                          BMI     17$             ;BR IF YES
2371  013020  005237  001416                  INC     TEMP            ;INC DELAY
2372  013024  001372                          BNE     11$             ;BR IF DELAY NOT DONE
2373  013026  005337  001246                  DEC     TEMP1           ;DEC DELAY COUNT
2374  013032  001367                          BNE     11$             ;BR IF NOT DONE DELAY
2375  013034  104014                          HLT     14              ;ERROR RDO NOT SET
2376  013036  016137  000002  001250  17$:     MOV     2(R1),TEMP2     ;SAVE SEL2
2377  013044  001001                          BNE     .+4             ;BR IF OK
2378  013046  104014                          HLT     14              ;ERROR!!! SEL2 = 0!!!!!!
2379  013050  032761  000004  000002          BIT     #BIT2,2(R1)     ;REC OR XMIT?
2380  013056  001032                          BNE     13$             ;BR IF REC
2381  013060  005737  021306          12$:     TST     TFLAG           ;FIRST TIME HERE?
2382  013064  001401                          BEQ     .+4             ;BR IF YES
2383  013066  104014                          HLT     14              ;ERROR MULTIPLE XMIT DONES
2384  013070  012737  177777  021306          MOV     #-1,TFLAG       ;SET TFLAG TO -1
2385  013076  132761  000001  000002          BITB    #BIT0,2(R1)     ;IS IT CONTROL 0
2386  013104  001401                          BEQ     .+4             ;BR IF NO
2387  013106  104014                          HLT     14              ;XMIT ERROR
2388  013110  022761  021314  000004          CMP     #TBUF,4(R1)     ;XMIT BA CORRECT?
2389  013116  001401                          BEQ     .+4             ;BR IF YES
2390  013120  104014                          HLT     14              ;XMIT BA ERROR
2391  013122  023761  021312  000006          CMP     TCOUNT,6(R1)    ;COUNT OK?
2392  013130  001401                          BEQ     .+4             ;BR IF YES
2393  013132  104014                          HLT     14              ;XMIT COUNT ERROR
2394  013134  142761  000207  000002          BICB    #207,2(R1)      ;CLEAR RDO AND BITS 0-2
2395  013142  000453                          BR      15$             ;CONTINUE
2396  013144  005737  021310          13$:     TST     RFLAG           ;FIRST TIME HERE?
2397  013150  001401                          BEQ     .+4             ;BR IF YES
2398  013152  104014                          HLT     14              ;ERROR MULTIPLE REC DONES
2399  013154  012737  177777  021310          MOV     #-1,RFLAG       ;SET RFLAG TO -1
2400  013162  132761  000001  000002          BITB    #BIT0,2(R1)     ;IS IT CNTL 0
2401  013170  001401                          BEQ     .+4             ;BR IF NO
2402  013172  104014                          HLT     14              ;RECEIVE ERROR
2403  013174  022761  021362  000004          CMP     #RBUF,4(R1)     ;REC BA CORRECT?
2404  013202  001401                          BEQ     .+4             ;BR IF YES
2405  013204  104014                          HLT     14              ;REC BA ERROR
2406  013206  023761  021360  000006          CMP     RCOUNT,6(R1)    ;COUNT OK?
2407  013214  001401                          BEQ     .+4             ;BR IF YES
2408  013216  104014                          HLT     14              ;REC COUNT ERROR
2409  013220  013700  021360                  MOV     RCOUNT,R0       ;GET SET TO CHECK DATA
2410  013224  012702  021314                  MOV     #TBUF,R2        ;R2 POINTS TO GOOD DATA
2411  013230  012703  021362                  MOV     #RBUF,R3        ;R3 POINTS TO RECEIVE DATA
2412  013234  010337  001252          9$:      MOV     R3,TEMP3        ;SAVE ADDRESS FOR TYPEOUT
2413  013240  112205                          MOVB    (R2)+,R5        ;R5 = XMIT DATA
2414  013242  112304                          MOVB    (R3)+,R4        ;R4 = RECEIVE DATA
2415  013244  120504                          CMPB    R5,R4           ;CHECK DATA
2416  013246  001401                          BEQ     .+4             ;BR IF OK
2417  013250  104013                          HLT     13              ;DATA ERROR
2418  013252  005300                          DEC     R0              ;DEC COUNT
2419  013254  001367                          BNE     9$              ;BR IF NOT DONE
2420  013256  005713                          TST     (R3)            ;THIS SHOULD BE 0, ELSE
2421  013260  001401                          BEQ     .+4             ;IT RECEIVED TO MUCH!!
2422  013262  104014                          HLT     14              ;ERROR
2423  013264  142761  000207  000002          BICB    #207,2(R1)      ;CLEAR RDO AND BITS 0-2
```

```
2424  013272  005737  021310          15$:    TST     RFLAG           ;REC DONE?
2425  013276  001640                          BEQ     16$             ;BR IF NO
2426  013300  005737  021306                  TST     TFLAG           ;XMIT DONE?
2427  013304  001635                          BEQ     16$             ;BR IF NO
2428  013306  004737  022472                  JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
2429  013312  012700  013340                  MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
2430  013316  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
2431  013322  012702  000010                  MOV     #10,R2          ;COUNT
2432  013326  122021                 22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
2433  013330  001007                          BNE     23$             ;IF ERROR BR 23$
2434  013332  005302                          DEC     R2              ;DEC COUNT
2435  013334  001374                          BNE     22$             ;CONTINUE CHECKING IF NOT DONE
2436  013336  000421                          BR      24$             ;ALL COUNTS OK, GET OUT
2437  013340     000     000     000  25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
2438  013343     000     000     000
2439  013346     000     000
2440  013350  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
2441  013356  113737  021435  001252          MOVB    BASE+5,TEMP3
2442  013364  113737  021437  001254          MOVB    BASE+7,TEMP4
2443  013372  113737  021441  001256          MOVB    BASE+11,TEMP5
2444  013400  104017                          HLT     17
2445  013402                          24$:
2446  013402  104400                  14$:    SCOPE                   ;SCOPE THIS TEST
2447
2448
2449                                          ;***************************** TEST 2 *****************************
2450                                          ;*OVERUN TEST
2451                                          ;*IN FREE RUNNING MODE SEND MESSAGE WITH NO RECEIVE
2452                                          ;*BUFFER AVAILABLE, VERIFY THAT AN OVERRUN ERROR OCCURS
2453                                          ;:*********************************************************
2454
2455                                          ;  TEST 2
2456                                          ;-------------
2457  013404  012737  000002  001226  TST2:   MOV     #2,TSTNO
2458  013412  012737  013744  001216          MOV     #TST3,NEXT
2459                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2460  013420  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT A DMC?
2461  013426  001406                          BEQ     .+16            ;BR IF YES
2462  013430  032737  000001  001372          BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2463  013436  001002                          BNE     .+6             ;BR IF YES
2464  013440  000137  013726                  JMP     10$             ;SKIP TEST
2465  013444  032737  010000  001366          BIT     #BIT12,STAT1    ;LU PRESENT?
2466  013452  001372                          BNE     .-12            ;BR IF NO
2467  013454  004737  022030                  JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
2468  013460  004537  022440                  JSR     R5,XFRELD       ;LOAD XMIT BA/CC
2469  013464  021314                          TBUF                    ;BA
2470  013466  000044                          44                      ;CC
2471  013470  012700  000010                  MOV     #10,R0          ;R0 = RETRANSMISSION COUNT
2472  013474  012703  000015                  MOV     #15,R3          ;DELAY COUNT
2473  013500  005037  001416                  CLR     TEMP            ;CLEAR DELAY COUNTER
2474  013504  105761  000002          1$:     TSTB    2(R1)           ;IS RDY 0 SET?
2475  013510  100407                          BMI     .+20            ;BR IF SET
2476  013512  005237  001416                  INC     TEMP            ;INC DELAY COUNTER
2477  013516  001372                          BNE     1$              ;BR IF NOT DONE DELAY
2478  013520  005303                          DEC     R3              ;DEC DELAY COUNT
2479  013522  001370                          BNE     1$              ;BR IF DELAY NOT DONE
```

```
2480  013524  104014                          HLT     14              ;ERROR, RDY 0 NOT SET
2481  013526  000477                          BR      10$             ;GET OUT
2482  013530  132761  000001  000002          BITB    #BIT0,2(R1)     ;IS IT CNTL 0?
2483  013536  001002                          BNE     11$             ;BR IF YES
2484  013540  104014                          HLT     14              ;ERROR, NOT CNTL 0
2485  013542  000471                          BR      10$             ;CONTINUE
2486  013544  012705  000004          11$:    MOV     #BIT2,R5        ;PUT "EXPECTED" IN R5
2487  013550  016104  000006                  MOV     6(R1),R4        ;PUT "FOUND" IN R4
2488  013554  020504                          CMP     R5,R4           ;IS ORUN SET?
2489  013556  001404                          BEQ     12$             ;BR IF YES
2490  013560  022704  000001                  CMP     #1,R4           ;DATA CK ERROR?
2491  013564  001461                          BEQ     13$             ;BR IF YES
2492  013566  104015                          HLT     15              ;ERROR, ORUN NOT SET
2493  013570  042761  000207  000002  12$:    BIC     #207,2(R1)      ;CLEAR RDO
2494  013576  005037  001416                  CLR     TEMP            ;RESET DELAY
2495  013602  005300                          DEC     R0              ;DEC RETRANS COUNT
2496  013604  001337                          BNE     1$              ;COUNTINUE
2497  013606  004737  022472                  JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
2498  013612  012700  013654                  MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS (LOW SPEED)
2499  013616  032737  000002  001372          BIT     #BIT1,STAT3     ;IS IT HIGH OR LOW
2500  013624  001402                          BEQ     21$             ;BR IF LOW
2501  013626  012700  013664                  MOV     #26$,R0         ;POINTER TO EXPECTED SOFT COUNTS (HIGH SPEED)
2502  013632  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
2503  013636  012702  000010                  MOV     #10,R2          ;COUNT
2504  013642  122021                  22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
2505  013644  001013                          BNE     23$             ;IF ERROR BR 23$
2506  013646  005302                          DEC     R2              ;DEC COUNT
2507  013650  001374                          BNE     22$             ;CONTINUE CHECKING IF NOT DONE
2508  013652  000425                          BR      24$             ;ALL COUNTS OK, GET OUT
2509  013654     000     000     077  25$:    .BYTE   0,0,077,100,0,0,0,0     ;EXPECTED ERROR COUNTS (LOW SPEED)
2510  013657     100     000     000
2511  013662     000     000
2512  013664     000     000     077  26$:    .BYTE   0,0,77,100,0,0,0,0      ;EXPECTED ERROR COUNTS (HIGH SPEED)
2513  013667     100     000     000
2514  013672     000     000
2515  013674  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
2516  013702  113737  021435  001252          MOVB    BASE+5,TEMP3
2517  013710  113737  021437  001254          MOVB    BASE+7,TEMP4
2518  013716  113737  021441  001256          MOVB    BASE+11,TEMP5
2519  013724  104017                          HLT     17
2520  013726                          24$:
2521  013726  104400                  10$:    SCOPE                   ;SCOPE THIS TEST
2522  013730  042761  000207  000002  13$:    BIC     #207,2(R1)      ;IGNOR THIS ERROR
2523  013736  005037  001416                  CLR     TEMP            ;RESET DELAY
2524  013742  000660                          BR      1$              ;CONTINUE
2525
2526
2527                                          ;***************************** TEST 3 ****************************
2528                                          ;*LOST DATA TEST
2529                                          ;*IN FREE RUNNING MODE SEND A MESSAGE LONGER THAN THE RECEIVE
2530                                          ;*BUFFER, VERIFY THAT A LOST DATA ERROR OCCURS.
2531                                          ;;*********************************************************************
2532
2533                                          ;  TEST 3
2534                                          ;--------------
2535  013744  012737  000003  001226  TST3:   MOV     #3,TSTNO
```

```
2536  013752  012737  014226  001216          MOV     #TST4,NEXT
2537
2538  013760  104412                          MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
2539  013762  032737  100000  001366          BIT     #BIT15,STAT1    ;MASTER CLEAR DMC11
2540  013770  001406                          BEQ     .+16            ;IS IT A DMC?
2541  013772  032737  000001  001372          BIT     #BIT0,STAT3     ;BR IF YES
2542  014000  001002                          BNE     .+6             ;KMC WITH BIT0 SET?
2543  014002  000137  014224                  JMP     10$             ;BR IF YES
2544  014006  032737  010000  001366          BIT     #BIT12,STAT1    ;SKIP TEST
2545  014014  001372                          BNE     .-12            ;LU PRESENT?
2546  014016  004737  022030                  JSR     PC,BASELD       ;BR IF NO
2547  014022  004537  022406                  JSR     R5,RFRELD               ;LOAD DMC BASE ADDRESS
2548  014026  021362                          RBUF                    ;LOAD RECEIVE BA/CC
2549  014030  000020                          20                      ;BA
2550  014032  004537  022440                  JSR     R5,XFRELD       ;CC
2551  014036  021314                          TBUF                    ;LOAD XMIT BA/CC
2552  014040  000044                          44                      ;BA
2553  014042  012703  000015                  MOV     #15,R3          ;CC
2554  014046  005037  001416                  CLR     TEMP            ;DELAY COUNT
2555  014052  105761  000002          1$:     TSTB    2(R1)           ;CLEAR DELAY COUNTER
2556  014056  100407                          BMI     .+20            ;IS RDY 0 SET?
2557  014060  005237  001416                  INC     TEMP            ;BR IF SET
2558  014064  001372                          BNE     1$              ;INC DELAY COUNTER
2559  014066  005303                          DEC     R3              ;BR IF NOT DONE DELAY
2560  014070  001370                          BNE     1$              ;DEC DELAY COUNT
2561  014072  104014                          HLT     14              ;BR IF DELAY NOT DONE
2562  014074  000453                          BR      10$             ;ERROR, RDY 0 NOT SET
2563  014076  132761  000001  000002          BITB    #BIT0,2(R1)     ;GET OUT
2564  014104  001002                          BNE     11$             ;IS IT CNTL 0?
2565  014106  104014                          HLT     14              ;BR IF YES
2566  014110  000445                          BR      10$             ;ERROR NOT CNTL 0
2567  014112  012705  000020          11$:    MOV     #BIT4,R5        ;CONTINUE
2568  014116  016104  000006                  MOV     6(R1),R4        ;PUT "EXPECTED" IN R5
2569  014122  020504                          CMP     R5,R4           ;PUT "FOUND" IN R4
2570  014124  001401                          BEQ     12$             ;IS LOST DATA SET?
2571  014126  104015                          HLT     15              ;BR IF YES
2572  014130  004737  022472          12$:    JSR     PC,SHUTDOWN     ;ERROR, LOST DATA NOT SET
2573  014134  012700  014162                  MOV     #25$,R0         ;SHUTDOWN DMC
2574  014140  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO EXPECTED SOFT COUNTS
2575  014144  012702  000010                  MOV     #10,R2          ;POINTER TO ACTUAL COUNTS
2576  014150  122021                  22$:    CMPB    (R0)+,(R1)+     ;COUNT
2577  014152  001007                          BNE     23$             ;COMPARE SOFT ERROR COUNTS
2578  014154  005302                          DEC     R2              ;IF ERROR BR 23$
2579  014156  001374                          BNE     22$             ;DEC COUNT
2580  014160  000421                          BR      24$             ;CONTINUE CHECKING IF NOT DONE
2581  014162     000     000     000  25$:    .BYTE   0,0,0,0,0,0,0,0 ;ALL COUNTS OK, GET OUT
2582  014165     000     000     000                                  ;EXPECTED ERROR COUNTS
2583  014170     000     000
2584  014172  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
2585  014200  113737  021435  001252          MOVB    BASE+5,TEMP3
2586  014206  113737  021437  001254          MOVB    BASE+7,TEMP4
2587  014214  113737  021441  001256          MOVB    BASE+11,TEMP5
2588  014222  104017                          HLT     17
2589  014224                          24$:
2590  014224  104400                  10$:    SCOPE                   ;SCOPE THIS TEST
2591
```

```
                                        ;*************************** TEST 4 ****************************
                                        ;*TRANSMIT NON-EXISTENT MEMORY TEST
                                        ;*IN FREE RUNNING MODE, LOAD A TRANSMIT BA THAT WILL TIME OUT
                                        ;*VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS
                                        ;;***************************************************************

                                        ;  TEST 4
                                        ;--------------
014226  012737  000004  001226  TST4:   MOV     #4,TSTNO
014234  012737  014500  001216          MOV     #TST5,NEXT
                                                                ;R1 CONTAINS BASE DMC11 ADDRESS
014242  104412                          MSTCLR                  ;MASTER CLEAR DMC11
014244  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT A DMC?
014252  001406                          BEQ     .+16            ;BR IF YES
014254  032737  000001  001372          BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
014262  001002                          BNE     .+6             ;BR IF YES
014264  000137  014476                  JMP     10$             ;SKIP TEST
014270  032737  010000  001366          BIT     #BIT12,STAT1    ;LU PRESENT?
014276  001372                          BNE     .-12            ;BR IF NO
014300  004737  022030                  JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
014304  004537  022440                  JSR     R5,XFRELD       ;LOAD XMIT BA/CC
014310  177320                          177320                  ;BA
014312  140044                          140044                  ;CC
014314  012703  000015                  MOV     #15,R3          ;DELAY COUNT
014320  005037  001416                  CLR     TEMP            ;CLEAR DELAY COUNTER
014324  105761  000002          1$:     TSTB    2(R1)           ;IS RDY 0 SET?
014330  100407                          BMI     .+20            ;BR IF SET
014332  005237  001416                  INC     TEMP            ;INC DELAY COUNTER
014336  001372                          BNE     1$              ;BR IF NOT DONE DELAY
014340  005303                          DEC     R3              ;DEC DELAY COUNT
014342  001370                          BNE     1$              ;BR IF DELAY NOT DONE
014344  104014                          HLT     14              ;ERROR, RDY 0 NOT SET
014346  000453                          BR      10$             ;GET OUT
014350  132761  000001  000002          BITB    #BIT0,2(R1)     ;IS IT CNTL 0?
014356  001002                          BNE     11$             ;BR IF YES
014360  104014                          HLT     14              ;ERROR, NOT CNTL 0
014362  000445                          BR      10$             ;CONTINUE
014364  012705  000400          11$:    MOV     #BIT8,R5        ;PUT "EXPECTED" IN R5
014370  016104  000006                  MOV     6(R1),R4        ;PUT "FOUND" IN R4
014374  020504                          CMP     R5,R4           ;IS NON-EX-MEM SET?
014376  001401                          BEQ     .+4             ;BR IF YES
014400  104015                          HLT     15              ;ERROR NON-EX-MEM NOT SET
014402  004737  022472                  JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
014406  012700  014434                  MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
014412  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
014416  012702  000010                  MOV     #10,R2          ;COUNT
014422  122021                  22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
014424  001007                          BNE     23$             ;IF ERROR BR 23$
014426  005302                          DEC     R2              ;DEC COUNT
014430  001374                          BNE     22$             ;CONTINUE CHECKING IF NOT DONE
014432  000421                          BR      24$             ;ALL COUNTS OK, GET OUT
014434    000     000     000   25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
014437    000     000     000
014442    000     000
014444  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
```

```
014452  113737  021435  001252          MOVB    BASE+5,TEMP3
014460  113737  021437  001254          MOVB    BASE+7,TEMP4
014466  113737  021441  001256          MOVB    BASE+11,TEMP5
014474  104017                          HLT     17
014476                          24$:
014476  104400                  10$:    SCOPE                           ;SCOPE THIS TEST


                                        ;****************************** TEST 5 ******************************
                                        ;*RECEIVE NON-EXISTENT MEMORY TEST
                                        ;*IN FREE RUNNING MODE, LOAD A RECEIVE BA THAT WILL TIME OUT
                                        ;*VERIFY THAT A NON-EXISTENT MEMORY ERROR OCCURS
                                        ;;*******************************************************************

                                        ;  TEST 5
                                        ;---------------
014500  012737  000005  001226  TST5:   MOV     #5,TSTNO
014506  012737  014762  001216          MOV     #TST6,NEXT
                                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
014514  104412                          MSTCLR                          ;MASTER CLEAR DMC11
014516  032737  100000  001366          BIT     #BIT15,STAT1            ;IS IT A DMC?
014524  001406                          BEQ     .+16                    ;BR IF YES
014526  032737  000001  001372          BIT     #BIT0,STAT3             ;KMC WITH BIT0 SET?
014534  001002                          BNE     .+6                     ;BR IF YES
014536  000137  014760                  JMP     10$                     ;SKIP TEST
014542  032737  010000  001366          BIT     #BIT12,STAT1            ;LU PRESENT?
014550  001372                          BNE     .-12                    ;BR IF NO
014552  004737  022030                  JSR     PC,BASELD                       ;LOAD DMC BASE ADDRESS
014556  004537  022406                  JSR     R5,RFRELD               ;LOAD RECEIVE BA/CC
014562  177320                          177320                          ;BA
014564  140044                          140044                          ;CC
014566  004537  022440                  JSR     R5,XFRELD               ;LOAD XMIT BA/CC
014572  021314                          TBUF                            ;BA
014574  000044                          44                              ;CC
014576  012703  000015                  MOV     #15,R3                  ;DELAY COUNT
014602  005037  001416                  CLR     TEMP                    ;CLEAR DELAY COUNTER
014606  105761  000002          1$:     TSTB    2(R1)                   ;IS RDY O SET?
014612  100407                          BMI     .+20                    ;BR IF SET
014614  005237  001416                  INC     TEMP                    ;INC DELAY COUNTER
014620  001372                          BNE     1$                      ;BR IF NOT DONE DELAY
014622  005303                          DEC     R3                      ;DEC DELAY COUNT
014624  001370                          BNE     1$                      ;BR IF DELAY NOT DONE
014626  104014                          HLT     14                      ;ERROR, RDY O NOT SET
014630  000453                          BR      10$                     ;GET OUT
014632  132761  000001  000002          BITB    #BIT0,2(R1)             ;IS IT CNTL O?
014640  001002                          BNE     11$                     ;BR IF YES
014642  104014                          HLT     14                      ;ERROR, NOT CNTL O
014644  000445                          BR      10$                     ;CONTINUE
014646  012705  000400          11$:    MOV     #BIT8,R5                ;PUT "EXPECTED" IN R5
014652  016104  000006                  MOV     6(R1),R4                ;PUT "FOUND" IN R4
014656  020504                          CMP     R5,R4                   ;IS NON-EX-MEM SET?
014660  001401                          BEQ     .+4                     ;BR IF YES
014662  104015                          HLT     15                      ;ERROR NON-EX-MEM NOT SET
014664  004737  022472                  JSR     PC,SHUTDOWN             ;SHUTDOWN DMC
014670  012700  014716                  MOV     #25$,R0                 ;POINTER TO EXPECTED SOFT COUNTS
014674  012701  021433          21$:    MOV     #BASE+3,R1              ;POINTER TO ACTUAL COUNTS
```

```
                                                        ;COUNT
                                                        ;COMPARE SOFT ERROR COUNTS
                                                        ;IF ERROR BR 23$
                                                        ;DEC COUNT
                                                        ;CONTINUE CHECKING IF NOT DONE
                                                        ;ALL COUNTS OK, GET OUT
                                                        ;EXPECTED ERROR COUNTS
                                                        ;SCOPE THIS TEST

                                                        ;*************************** TEST 6 ****************************
                                                        ;*PROCESSOR ERROR TEST
                                                        ;*IN FREE RUNNING MODE, DO A BASE TRANSFER REQUEST AFTER A
                                                        ;*BASE HAS BEEN SET UP, VERIFY THAT A PROCESSOR ERROR OCCURS.
                                                        ;;*************************************************************

                                                        ;  TEST 6
                                                        ;--------------
                                                        ;R1 CONTAINS BASE DMC11 ADDRESS
                                                        ;MASTER CLEAR DMC11
                                                        ;IS IT A DMC?
                                                        ;BR IF YES
                                                        ;KMC WITH BIT0 SET?
                                                        ;BR IF YES
                                                        ;SKIP TEST
                                                        ;LU PRESENT?
                                                        ;BR IF NO
                                                        ;LOAD BASE ADDRESS
                                                        ;2ND BASE REQUEST
                                                        ;RDI SET?
                                                        ;BR IF NO
                                                        ;CLEAR RQI
                                                        ;GET SET TO DELAY
                                                        ;RDO SET?
                                                        ;BR IF YES
                                                        ;INC DELAY
                                                        ;BR IF NOT DONE DELAY
                                                        ;ERROR, RDO NOT SET
                                                        ;TRY AGAIN
                                                        ;IS IS CNTL 0?
                                                        ;BR IF YES
                                                        ;ERROR NOT CNTL 0
                                                        ;CONTINUE
                                                        ;PUT "EXPECTED" IN R5
                                                        ;PUT "FOUND" IN R4
                                                        ;IS PROC ERROR SET?
```

```
2760  015126  001401                          BEQ     .+4             ;BR IF YES
2761  015130  104015                          HLT     15              ;ERROR, PROC ERROR NOT SET
2762  015132  012700  015160                  MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
2763  015136  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
2764  015142  012702  000010                  MOV     #10,R2          ;COUNT
2765  015146  122021                  22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
2766  015150  001007                          BNE     23$             ;IF ERROR BR 23$
2767  015152  005302                          DEC     R2              ;DEC COUNT
2768  015154  001374                          BNE     22$             ;CONTINUE CHECKING IF NOT DONE
2769  015156  000421                          BR      24$             ;ALL COUNTS OK, GET OUT
2770  015160     000     000     000  25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
2771  015163     000     000     000
2772  015166     000     000
2773  015170  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
2774  015176  113737  021435  001252          MOVB    BASE+5,TEMP3
2775  015204  113737  021437  001254          MOVB    BASE+7,TEMP4
2776  015212  113737  021441  001256          MOVB    BASE+11,TEMP5
2777  015220  104017                          HLT     17
2778  015222                          24$:
2779  015222  104400                  10$:    SCOPE                   ;SCOPE THIS TEST
2780
2781
2782                                          ;***************************** TEST 7 *****************************
2783                                          ;*PROCESSOR ERROR TEST
2784                                          ;*IN FREE RUNNING MODE DO A RQI WITH AN ILLEGAL 10 CODE
2785                                          ;*VERIFY THAT A PROCESSOR ERROR OCCURS
2786                                          ;;*************************************************************
2787
2788                                          ;  TEST 7
2789                                          ;--------------
2790  015224  012737  000007  001226  TST7:   MOV     #7,TSTNO
2791  015232  012737  015466  001216          MOV     #TST10,NEXT
2792                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2793  015240  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2794  015242  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT A DMC?
2795  015250  001406                          BEQ     .+16            ;BR IF YES
2796  015252  032737  000001  001372          BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2797  015260  001002                          BNE     .+6             ;BR IF YES
2798  015262  000137  015464                  JMP     10$             ;SKIP TEST
2799  015266  032737  010000  001366          BIT     #BIT12,STAT1    ;LU PRESENT?
2800  015274  001372                          BNE     .-12            ;BR IF NO
2801  015276  004737  022030                  JSR     PC,BASELD               ;LOAD DMC BASE ADDRESS
2802  015302  152711  000046                  BISB    #46,(R1)        ;RQI AND ILLEGAL CODE
2803  015306  105711                          TSTB    (R1)            ;WAIT FOR RDI
2804  015310  100376                          BPL     .-2             ;BR IF NO RDI
2805  015312  142711  000040                  BICB    #40,(R1)        ;CLEAR RQI
2806  015316  005037  001416                  CLR     TEMP            ;CLEAR COUNTER
2807  015322  105761  000002          1$:     TSTB    2(R1)           ;RDY 0 SET?
2808  015326  100405                          BMI     .+14            ;BR IF YES
2809  015330  005237  001416                  INC     TEMP            ;BUMP COUNTER DELAY
2810  015334  001372                          BNE     1$              ;BR IF NOT DONE
2811  015336  104014                          HLT     14              ;ERROR NO RDY 0
2812  015340  000770                          BR      1$              ;TRY AGAIN
2813  015342  132761  000001  000002          BITB    #BIT0,2(R1)     ;IS IT CNTL 0
2814  015350  001002                          BNE     11$             ;BR IF YES
2815  015352  104014                          HLT     14              ;ERROR, NOT CNTL 0
```

```
2816  015354  000443                          BR      10$             ;CONTINUE
2817  015356  012705  001000          11$:    MOV     #BIT9,R5        ;PUT "EXPECTED" IN R5
2818  015362  016104  000006                  MOV     6(R1),R4        ;PUT "FOUND" IN R4
2819  015366  020504                          CMP     R5,R4           ;IS PROC ERROR SET?
2820  015370  001401                          BEQ     .+4             ;BR IF YES
2821  015372  104015                          HLT     15              ;ERROR PROC ERROR NOT SET
2822  015374  012700  015422                  MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS
2823  015400  012701  021433          21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
2824  015404  012702  000010                  MOV     #10,R2          ;COUNT
2825  015410  122021                  22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
2826  015412  001007                          BNE     23$             ;IF ERROR BR 23$
2827  015414  005302                          DEC     R2              ;DEC COUNT
2828  015416  001374                          BNE     22$             ;CONTINUE CHECKING IF NOT DONE
2829  015420  000421                          BR      24$             ;ALL COUNTS OK, GET OUT
2830  015422     000     000     000  25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS
2831  015425     000     000     000
2832  015430     000     000
2833  015432  113737  021433  001250  23$:    MOVB    BASE+3,TEMP2
2834  015440  113737  021435  001252          MOVB    BASE+5,TEMP3
2835  015446  113737  021437  001254          MOVB    BASE+7,TEMP4
2836  015454  113737  021441  001256          MOVB    BASE+11,TEMP5
2837  015462  104017                          HLT     17
2838  015464                          24$:
2839  015464  104400                  10$:    SCOPE                   ;SCOPE THIS TEST
2840
2841
2842                                          ;*************************** TEST 10 ****************************
2843                                          ;*HALF DUPLEX TEST
2844                                          ;*IN FREE RUNNING MODE, SET HALF DUPLEX AND L U LOOP
2845                                          ;*SEND A MESSAGE AND VERIFY THAT THERE ARE NO DONES
2846                                          ;:****************************************************************
2847
2848                                          ;  TEST 10
2849                                          ;-------------
2850  015466  012737  000010  001226  TST10:  MOV     #10,TSTNO
2851  015474  012737  015622  001216          MOV     #TST11,NEXT
2852                                                                  ;R1 CONTAINS BASE DMC11 ADDRESS
2853  015502  104412                          MSTCLR                  ;MASTER CLEAR DMC11
2854  015504  032737  100000  001366          BIT     #BIT15,STAT1    ;IS IT A DMC?
2855  015512  001406                          BEQ     .+16            ;BR IF YES
2856  015514  032737  000001  001372          BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2857  015522  001002                          BNE     .+6             ;BR IF YES
2858  015524  000137  015614                  JMP     10$             ;SKIP TEST
2859  015530  032737  010000  001366          BIT     #BIT12,STAT1    ;LU PRESENT?
2860  015536  001372                          BNE     .-12            ;BR IF NO
2861  015540  004737  022146                  JSR     PC,BASELH       ;LOAD BASE AND HALF DUPLEX
2862  015544  004537  022406                  JSR     R5,RFRELD       ;LOAD RECEIVE BUFFER
2863  015550  021362                          RBUF                    ;BA
2864  015552  000044                          44                      ;CC
2865  015554  004537  022440                  JSR     R5,XFRELD       ;LOAD TRANSMIT BUFFER
2866  015560  021314                          TBUF                    ;BA
2867  015562  000044                          44                      ;CC
2868  015564  012703  000003                  MOV     #3,R3           ;LOAD DELAY COUNT
2869  015570  005037  001416                  CLR     TEMP            ;CLEAR DELAY
2870  015574  105761  000002          4$:     TSTB    2(R1)           ;IS DONE SET?
2871  015600  100'06                          BMI     5$              ;BR IF YES (ERROR)
```

```
2872  015602  005237  001416              INC     TEMP            ;INC DELAY
2873  015606  001372                      BNE     4$              ;BR IF DELAY NOT DONE
2874  015610  005303                      DEC     R3              ;DEC DELAY COUNT
2875  015612  001370                      BNE     4$              ;BR IF DELAY NOT DONE
2876  015614  104400              10$:    SCOPE                   ;SCOPE THIS TEST
2877  015616  104014              5$:     HLT     14              ;ERROR DONE WITH HALF-DUPLEX
2878  015620  000775                      BR      10$             ;GET OUT
2879
2880
2881                                      ;***************************** TEST 11 ****************************
2882                                      ;*RESUME TEST
2883                                      ;*THIS TEST SENDS AND RECEIVES A BUFFER AND SHUTS DOWN THE
2884                                      ;*DMC. THEN A MASTER CLEAR IS ISSUED AND A BASE WITH RESUME
2885                                      ;*BIT SET IS GIVEN, ANOTHER BUFFER IS SENT AND RECEIVED.
2886                                      ;*DATA IS CHECKED.
2887                                      ;:*************************************************************************
2888
2889                                      ;  TEST 11
2890                                      ;--------------
2891  015622  012737  000011  001226  TST11:  MOV     #11,TSTNO
2892  015630  012737  016204  001216          MOV     #TST12,NEXT
2893                                                              ;R1 CONTAINS BASE DMC11 ADDRESS
2894  015636  104412                      MSTCLR                  ;MASTER CLEAR DMC11
2895  015640  032737  100000  001366      BIT     #BIT15,STAT1    ;IS IT A DMC?
2896  015646  001406                      BEQ     .+16            ;BR IF YES
2897  015650  032737  000001  001372      BIT     #BIT0,STAT3     ;KMC WITH BIT0 SET?
2898  015656  001002                      BNE     .+6             ;BR IF YES
2899  015660  000137  016202              JMP     10$             ;SKIP TEST
2900  015664  032737  010000  001366      BIT     #BIT12,STAT1    ;LU PRESENT?
2901  015672  001372                      BNE     .-12            ;BR IF NO
2902  015674  005037  020050              CLR     RESUME          ;CLR RESUME FLAG
2903  015700  005737  020050      1$:     TST     RESUME          ;FIRST OR SECOND PASS?
2904  015704  001003                      BNE     2$              ;BR IF SECOND
2905  015706  004737  022030              JSR     PC,BASELD       ;BASE
2906  015712  000402                      BR      3$              ;CONTINUE
2907  015714  004737  022266      2$:     JSR     PC,RESUM        ;BASE WITH RESUME BIT
2908  015720  004537  022406      3$:     JSR     R5,RFRELD       ;RECEIVE BUFFER
2909  015724  021362                      RBUF                    ;BA
2910  015726  000044                      44                      ;CC
2911  015730  004537  022440              JSR     R5,XFRELD       ;XMIT BUFFER
2912  015734  021314                      TBUF                    ;BA
2913  015736  000044                      44                      ;CC
2914  015740  012703  000030              MOV     #30,R3          ;DELAY COUNT
2915  015744  012700  000002              MOV     #2,R0           ;NEED TWO DONES
2916  015750  005037  001416              CLR     TEMP            ;CLEAR DELAY COUNTER
2917  015754  105761  000002      4$:     TSTB    2(R1)           ;IS RDY 0 SET?
2918  015760  100407                      BMI     .+20            ;BR IF SET
2919  015762  005237  001416              INC     TEMP            ;INC DELAY COUNTER
2920  015766  001372                      BNE     4$              ;BR IF NOT DONE DELAY
2921  015770  005303                      DEC     R3              ;DEC DELAY COUNT
2922  015772  001370                      BNE     4$              ;BR IF DELAY NOT DONE
2923  015774  104014                      HLT     14              ;ERROR, RDY 0 NOT SET
2924  015776  000501                      BR      10$             ;GET OUT
2925  016000  042761  000207  000002      BIC     #207,2(R1)      ;CLEAR DONE
2926  016006  005300                      DEC     R0              ;TWO DONES YET?
2927  016010  001361                      BNE     4$              ;BR IF NOT
```

```
                                        MOV     #TBUF,R2        ;ADDRESS OF GOOD DATA
                                        MOV     #RBUF,R3        ;ADDRESS OF RECEIVED DATA
                                        MOV     #44,R0          ;COUNT
                                6$:     MOVB    (R2)+,R5        ;LOAD GOOD DATA
                                        MOVB    (R3)+,R4        ;LOAD FOUND DATA
                                        CMPB    R5,R4           ;COMPARE DATA
                                        BEQ     7$              ;BR IF OK
                                        HLT     12              ;DATA ERROR
                                7$:     DEC     R0              ;DONE YET?
                                        BNE     6$              ;BR IF NOT
                                        JSR     PC,SHUTDOWN     ;SHUTDOWN DMC
                                        TST     RESUME          ;
                                        BNE     8$              ;BR IF ALL DONE
                                        MOV     #-1,RESUME      ;SET FLAG FOR SECOND PASS
                                        BR      1$              ;CONTINUE
                                8$:
                                        MOV     #25$,R0         ;POINTER TO EXPECTED SOFT COUNTS (LOW SPEED)
                                        BIT     #BIT1,STAT3     ;IS IT HIGH OR LOW
                                        BEQ     21$             ;BR IF LOW
                                        MOV     #26$,R0         ;POINTER TO EXPECTED SOFT COUNTS (HIGH SPEED)
                                21$:    MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
                                        MOV     #10,R2          ;COUNT
                                22$:    CMPB    (R0)+,(R1)+     ;COMPARE SOFT ERROR COUNTS
                                        BNE     23$             ;IF ERROR BR 23$
                                        DEC     R2              ;DEC COUNT
                                        BNE     22$             ;CONTINUE CHECKING IF NOT DONE
                                        BR      24$             ;ALL COUNTS OK, GET OUT
                                25$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS (LOW SPEED)
                                26$:    .BYTE   0,0,0,0,0,0,0,0 ;EXPECTED ERROR COUNTS (HIGH SPEED)
                                23$:    MOVB    BASE+3,TEMP2
                                        MOVB    BASE+5,TEMP3
                                        MOVB    BASE+7,TEMP4
                                        MOVB    BASE+11,TEMP5
                                        HLT     17
                                24$:
                                10$:    SCOPE                   ;SCOPE THIS TEST


                                ;****************************** TEST 12 ******************************
                                ;*FREE RUNNING DATA TEST (INTERRUPT DRIVEN EXERCISER)
                                ;*THIS TEST REPEATEDLY QUEUES UP 7 RECEIVE BUFFERS AND
                                ;*7 TRANSMIT BUFFERS AND CHECKS DATA WHEN ALL 7 BUFFERS
                                ;*ARE RECEIVED. TRANSMIT COUNTS RANGE FROM 2 TO 104.
                                ;*DATA IS A BINARY COUNT PATTERN. THE RESUME FUNCTION
                                ;*IS CHECKED IN THIS TEST. THIS TEST USES THE TURNAROUND CONNECTOR
                                ;*IF IT IS PRESENT, OTHERWISE LINE UNIT LOOP IS SET.
                                ;:*************************************************************

                                ;  TEST 12
                                ;--------------
                        TST12:  MOV     #12,TSTNO
                                MOV     #.EOP,NEXT
```

```
2984
2985  016220  104412                          MSTCLR                  ;R1 CONTAINS BASE DMC11 ADDRESS
2986  016222  032737  100000  001366          BIT     #BIT15,STAT1    ;MASTER CLEAR DMC11
2987  016230  001406                          BEQ     .+16            ;IS IT A DMC?
2988  016232  032737  000001  001372          BIT     #BIT0,STAT3     ;BR IF YES
2989  016240  001002                          BNE     .+6             ;KMC WITH BIT0 SET?
2990  016242  000137  017034                  JMP     ENDEX1          ;BR IF YES
2991  016246  032737  010000  001366          BIT     #BIT12,STAT1    ;SKIP TEST
2992  016254  001372                          BNE     .-12            ;LU PRESENT?
2993  016256  012737  000340  177776          MOV     #340,PS         ;BR IF NO
2994  016264  013700  001366                  MOV     STAT1,R0        ;LOCK OUT INTERRUPTS
2995  016270  006200                          ASR     R0              ;GET BR LEVEL
2996  016272  006200                          ASR     R0              ;SHIFT RIGHT 4 TIMES
2997  016274  006200                          ASR     R0
2998  016276  006200                          ASR     R0
2999  016300  042700  177437                  BIC     #177437,R0      ;PUT BR LEVEL IN R0
3000  016304  012777  017122  163062          MOV     #IISR,@DMRVEC   ;LOAD INPUT VECTOR
3001  016312  010077  163060                  MOV     R0,@DMRLVL      ;LOAD LEVEL
3002  016316  012777  017412  163054          MOV     #OISR,@DMTVEC   ;LOAD OUTPUT VECTOR
3003  016324  010077  163052                  MOV     R0,@DMTLVL      ;LOAD LEVEL
3004
3005                                          ;INITIALIZE ALL BUFFER LISTS AND COUNT LISTS
3006
3007  016330  012737  000104  021306          MOV     #104,TFLAG      ;TFLAG CONTAINS COUNT
3008  016336  012700  020054                  MOV     #XMITBA+2,R0    ;R0 POINTS TO BA LIST
3009  016342  012703  020346                  MOV     #RBUFF,R3       ;R3 CONTAINS BUFFER ADDRESS
3010  016346  010320                  1$:     MOV     R3,(R0)+        ;LOAD BA LIST WITH REC BA
3011  016350  062703  000104                  ADD     #104,R3         ;UPDATE BUFFER ADDRESS
3012  016354  022700  020072                  CMP     #XMITBA+20,R0   ;END OF REC BUFFERS?
3013  016360  001372                          BNE     1$              ;NO LOAD NEXT ONE
3014  016362  012720  020110          2$:     MOV     #TBUFF,(R0)+    ;LOAD BA LIST WITH XMIT BA
3015  016366  022700  020110                  CMP     #XMITBA+36,R0   ;END OF XMIT BUFFERS?
3016  016372  001373                          BNE     2$              ;NO LOAD NEXT BUFFER
3017  016374  012700  020222                  MOV     #RCNTAB+2,R0    ;R0 POINTS TO COUNT LIST
3018  016400  013720  021306          3$:     MOV     TFLAG,(R0)+     ;LOAD COUNT OF 104
3019  016404  022700  020240                  CMP     #RCNTAB+20,R0   ;END OF REC COUNT LIST?
3020  016410  001373                          BNE     3$              ;BR IF NO
3021  016412  012737  000005  021304          MOV     #5,FLAG ;LOOP COUNT
3022  016420  012711  040000                  MOV     #BIT14,(R1)     ;SET MASTER CLEAR
3023  016424  032737  100000  001366          BIT     #BIT15,STAT1    ;IOP?
3024  016432  001402                          BEQ     .+6             ;BR IF NO
3025  016434  012711  100000                  MOV     #BIT15,(R1)     ;SET RUN ON IOP
3026  016440  012700  177777                  MOV     #-1,R0          ;R0 IS INPUT DONE COUNTER
3027  016444  005037  020050          CLRTAB: CLR     RESUME          ;CLEAR RESUME FLAG
3028  016450  012705  020256                  MOV     #RDNTAB,R5      ;GET READY TO CLEAR ALL RECEIVE
3029  016454  005025                  2$:     CLR     (R5)+           ;BUFFERS
3030  016456  022705  021302                  CMP     #RBUFFE,R5      ;END OF BUFFER?
3031  016462  001374                          BNE     2$              ;BR IF NO
3032  016464  012704  020240                  MOV     #XCNTAB,R4      ;R4 POINTS TO XMIT COUNT LIST
3033  016470  013724  021306          4$:     MOV     TFLAG,(R4)+     ;LOAD XMIT CHAR COUNT
3034  016474  022704  020256                  CMP     #XCNTAB+16,R4   ;DONE?
3035  016500  001373                          BNE     4$              ;BR IF NO
3036  016502  005002                  5$:     CLR     R2              ;R2 IS OUTPUT DONE COUNTER
3037  016504  005004                          CLR     R4              ;R4 IS USED AS INDEX IN OISR
3038  016506  005711                          TST     (R1)            ;IS RUN SET?
3039  016510  100376                          BPL     .-2             ;WAIT FOR RUN
```

```
3040  016512  152761  000100  000002          BISB    #BIT6,2(R1)     ;SET IEO
3041  016520  032737  040000  001366          BIT     #BIT14,STAT1    ;LOOP BACK CONNECTOR?
3042  016526  001002                          BNE     .+6             ;BR IF YES
3043  016530  052711  004000                  BIS     #BIT11,(R1)     ;SET LINE UNIT LOOP
3044  016534  022737  000005  021304          CMP     #5,FLAG         ;FIRST TIME?
3045  016542  001003                          BNE     1$              ;BR IF NOT
3046  016544  052711  000143                  BIS     #143,(R1)       ;SET IEI,RQI,BASE I
3047  016550  000402                          BR      3$              ;CONTINUE
3048  016552  052711  000144          1$:     BIS     #144,(R1)       ;SET IEI, RQI, REC BA/CC
3049  016556  005037  001416          3$:     CLR     TEMP            ;SET UP FOR DELAY COUNT
3050  016562  012737  000022  001250          MOV     #22,TEMP2       ;GET SET FOR DELAY
3051  016570  005037  177776                  CLR     PS              ;ALLOW INTERRUPTS
3052  016574  022700  000020          SCAN:   CMP     #20,R0          ;INPUT DONE?
3053  016600  001402                          BEQ     SCAN2           ;BR IF YES
3054  016602  000137  017072                  JMP     SCAN1           ;BR IF NO
3055  016606  022702  000034          SCAN2:  CMP     #34,R2          ;XMIT DONE FOR ALL MESSAGES?
3056  016612  001402                          BEQ     8$              ;BR IF YES
3057  016614  000137  017072                  JMP     SCAN1           ;BR IF NO
3058  016620  022704  000034          8$:     CMP     #34,R4          ;REC DONE FOR ALL MESSAGES?
3059  016624  001402                          BEQ     9$              ;BR IF YES
3060  016626  000137  017072                  JMP     SCAN1           ;BR IF NO
3061  016632                          9$:
3062  016632  012700  020256                  MOV     #RDNTAB,R0      ;GET FIRST REC BUFFER
3063  016636  012002                  5$:     MOV     (R0)+,R2        ;R2 POINTS TO BUFFER
3064  016640  005005                          CLR     R5              ;R5=EXPECTED
3065  016642  005003                          CLR     R3              ;R3 = COUNT
3066  016644  010237  001252          6$:     MOV     R2,TEMP3        ;SAVE ADDRESS FOR TYPEOUT
3067  016650  112204                          MOVB    (R2)+,R4        ;GET RECEIVE DATA
3068  016652  120504                          CMPB    R5,R4           ;IS IT CORRECT?
3069  016654  001401                          BEQ     .+4             ;BR IF YES
3070  016656  104013                          HLT     13              ;DATA ERROR
3071  016660  005205                          INC     R5              ;NEXT CHARACTER
3072  016662  005203                          INC     R3              ;INC COUNT
3073  016664  021003                          CMP     (R0),R3         ;DONE YET?
3074  016666  001366                          BNE     6$              ;BR IF NO
3075  016670  062700  000002                  ADD     #2,R0           ;GET NEXT REC BUFFER
3076  016674  022700  020312                  CMP     #RDNTAB+34,R0   ;DONE YET?
3077  016700  001356                          BNE     5$              ;BR IF NO
3078  016702  012700  000001                  MOV     #1,R0           ;SET R0 TO 1
3079  016706  032737  000001  021304  4$:     BIT     #BIT0,FLAG      ;CHANGE CHAR COUNT FOR NEXT LOOP
3080  016714  001003                          BNE     1$              ;BR TO SUB 40
3081  016716  005337  021306                  DEC     TFLAG           ;DEC BY ONE
3082  016722  000403                          BR      2$              ;CONTINUE
3083  016724  162737  000040  021306  1$:     SUB     #40,TFLAG       ;SUBTRACT 40 FRON XMIT COUNT
3084  016732  005337  021304          2$:     DEC     FLAG            ;DEC LOOP COUNT
3085  016736  001242                          BNE     CLRTAB          ;GO DO IT AGAIN
3086  016740  152711  000146          ENDEX:  BISB    #146,(R1)       ;SHUT DOWN DMC
3087  016744  005737  021304          1$:     TST     FLAG            ;HAS INTERRUPT OCCURED?
3088  016750  001775                          BEQ     1$              ;BR IF NO
3089  016752  012700  017014                  MOV     #10$,R0         ;R0 POINTS TO LO SPEED COUNTS
3090  016756  032737  000002  001372          BIT     #BIT1,STAT3     ;IS IT LO SPEED?
3091  016764  001402                          BEQ     2$              ;BR IF YES
3092  016766  012700  017024                  MOV     #11$,R0         ;R0 POINTS TO HI COUNTS
3093  016772  012701  021433          2$:     MOV     #BASE+3,R1      ;POINTER TO ACTUAL COUNTS
3094  016776  012702  000010                  MOV     #10,R2          ;10 COUNTS TO CHECK
3095  017002  122021                  3$:     CMPB    (R0)+,(R1)+     ;CHECK COUNT
```

```
3096  017004  103414                        BLO     ENDEX2          ;BR IF ERROR
3097  017006  005302                        DEC     R2              ;DEC COUNT
3098  017010  001374                        BNE     3$              ;BR IF NOT DONE
3099  017012  000410                        BR      ENDEX1          ;ALL OK GET OUT
3100  017014     000     000     000  10$:  .BYTE   0,0,0,0,0,0,5,5 ;EXPECTED LO SPEED COUNTS
3101  017017     000     000     000
3102  017022     005     005
3103  017024     000     000     005  11$:  .BYTE   0,0,5,0,0,0,5,5 ;EXPECTED HI SPEED COUNTS
3104  017027     000     000     000
3105  017032     005     005
3106  017034  104400                  ENDEX1: SCOPE                 ;SCOPE THIS TEST
3107  017036  113737  021433  001250  ENDEX2: MOVB    BASE+3,TEMP2    ;SAVE ALL ODD ADDRESSES
3108  017044  113737  021435  001252          MOVB    BASE+5,TEMP3    ;FOR TYPEOUT
3109  017052  113737  021437  001254          MOVB    BASE+7,TEMP4    ;
3110  017060  113737  021441  001256          MOVB    BASE+11,TEMP5
3111  017066  104017                        HLT     17              ;NON ZERO ERROR COUNT
3112  017070  000761                        BR      ENDEX1          :GET OUT
3113  017072  005337  001416          SCAN1: DEC     TEMP            ;DECREMENT DELAY COUNTER
3114  017076  001402                        BEQ     1$              ;BR IF ZERO
3115  017100  000137  016574                JMP     SCAN            ;BR IF NOT DONE DELAY
3116  017104  005337  001250          1$:   DEC     TEMP2           ;DEC DELAY COUNT
3117  017110  001402                        BEQ     2$              ;BR IF DONE DELAY
3118  017112  000137  016574                JMP     SCAN            ;BR IF NOT DONE
3119  017116  104001                  2$:   HLT     1               ;ERROR HUNG
3120  017120  000745                        BR      ENDEX1          ;GET OUT
3121
3122                                        ;INPUT INTERRUPT SERVICE ROUTINE
3123
3124  017122  022700  000017          IISR: CMP     #17,R0          ;PROC. ERROR DONE?
3125  017126  001421                        BEQ     12$             ;BR IF YES
3126  017130  005737  020050                TST     RESUME          ;IS THIS A RESUME INTERRUPT
3127  017134  001432                        BEQ     8$              ;BR IF NO
3128  017136  032711  000002                BIT     #BIT1,(R1)      ;CNTL OR BASE?
3129  017142  001407                        BEQ     13$             ;BR IF CNTL I
3130  017144  012761  021430  000004        MOV     #BASE,4(R1)     ;LOAD BASE ADDRESS
3131  017152  012761  010000  000006        MOV     #BIT12,6(R1)    ;WITH RESUME BIT SET
3132  017160  000404                        BR      12$             ;CONTINUE
3133  017162  005061  000006          13$:  CLR     6(R1)           ;SELECT FULL DUPLEX
3134  017166  005037  020050                CLR     RESUME          ;CLEAR RESUME FLAG
3135  017172  142711  000040          12$:  BICB    #40,(R1)        ;CLEAR RQI
3136  017176  105711                        TSTB    (R1)            ;IS RDI GONE?
3137  017200  100776                        BMI     .-2             ;BR IF NO
3138  017202  005737  020050                TST     RESUME          ;BASE OR CNTL I?
3139  017206  001403                        BEQ     14$             ;BR IF IT WAS CNTL I
3140  017210  152711  000041                BISB    #41,(R1)        ;ASK FOR CNTL I
3141  017214  000002                        RTI                     ;RETURN
3142  017216  105011                  14$:  CLRB    (R1)            ;CLEAR BSEL 0
3143  017220  000002                        RTI                     ;RETURN
3144  017222  005700                  8$:   TST     R0              ;FIRST TIME HERE?
3145  017224  100006                        BPL     7$              ;LOAD BASE IF MINUS
3146  017226  012761  021430  000004        MOV     #BASE,4(R1)     ;SET UP BASE ADDRESS
3147  017234  005061  000006                CLR     6(R1)           ;CLEAR COUNT
3148  017240  000434                        BR      3$              ;CONTINUE
3149  017242  001003                  7$:   BNE     1$              ;CNTL I FULL DUPLEX IF 0
3150  017244  005061  000006                CLR     6(R1)           ;SELECT FULL DUPLEX
3151  017250  000430                        BR      3$              ;CONTINUE
```

```
3152  017252  032700  000010          1$:   BIT   #BIT3,R0          ;XMIT?
3153  017256  001013                        BNE   2$                ;BR IF YES
3154  017260  000241                        CLC                     ;CLEAR CARRY
3155  017262  006100                        ROL   R0                ;MAKE R0 EVEN
3156  017264  016061  020052  000004        MOV   RECBA(R0),4(R1)   ;LOAD REC BUFFER
3157  017272  016061  020220  000006        MOV   RCNTAB(R0),6(R1);LOAD COUNT
3158  017300  000241                        CLC                     ;CLEAR CARRY
3159  017302  006000                        ROR   R0                ;GET R0 BACK
3160  017304  000412                        BR    3$                ;CONTINUE
3161  017306  000241                  2$:   CLC                     ;CLEAR CARRY
3162  017310  006100                        ROL   R0                ;MAKE IT EVEN
3163  017312  016061  020052  000004        MOV   XMITBA(R0),4(R1);LOAD XMIT BUFFER
3164  017320  016061  020220  000006        MOV   RCNTAB(R0),6(R1);LOAD COUNT
3165  017326  000241                        CLC                     ;CLEAR CARRY
3166  017330  006000                        ROR   R0                ;PUT IT BACK
3167  017332  142711  000040          3$:   BICB  #40,(R1)          ;CLEAR RQI
3168  017336  105711                        TSTB  (R1)              ;WAIT FOR
3169  017340  100776                        BMI   .-2               ;RDI TO GO AWAY
3170  017342  005200                        INC   R0                ;INC COUNT
3171  017344  001003                        BNE   6$                ;IF 0 ASK FOR CNTL I
3172  017346  152711  000041              BISB  #41,(R1)          ;ASK FOR CNTL I
3173  017352  000002                        RTI                     ;RETURN
3174  017354  022700  000017          6$:   CMP   #17,R0            ;DONE YET?
3175  017360  001411                        BEQ   4$                ;BR IF YES
3176  017362  032700  000010              BIT   #BIT3,R0          ;XMIT?
3177  017366  001003                        BNE   5$                ;BR IF YES
3178  017370  152711  000044              BISB  #44,(R1)          ;ASK FOR REC BA/CC
3179  017374  000002                        RTI                     ;RETURN
3180  017376  152711  000040          5$:   BISB  #40,(R1)          ;ASK FOR XMIT BA/CC
3181  017402  000002                        RTI                     ;RETURN
3182  017404  152711  000046          4$:   BISB  #46,(R1)          ;FORCE PROC. ERROR
3183  017410  000002                        RTI                     ;RETURN
3184
3185                                        ;OUTPUT INTERRUPT SERVICE ROUTINE
3186
3187  017412  032761  000001  000002  0ISR:  BIT   #BIT0,2(R1)      ;IS THIS AN ERROR?
3188  017420  001467                        BEQ   1$                ;BR IF NO
3189  017422  005737  021304              TST   FLAG              ;IS THIS SHUT DOWN INTERRUPT?
3190  017426  001006                        BNE   9$                ;BR IF NO
3191  017430  005237  021304              INC   FLAG              ;YES MAKE FLAG NON-ZERO
3192  017434  022761  001000  000006        CMP   #BIT9,6(R1)       ;SHUT DOWN BIT SET?
3193  017442  001531                        BEQ   10$               ;YES ALL IS OK
3194  017444  022700  000017          9$:   CMP   #17,R0            ;RESUME INTERRUPT?
3195  017450  001041                        BNE   11$               ;BR IF NO
3196  017452  022761  001000  000006        CMP   #BIT9,6(R1)       ;PROC. ERROR BIT SET?
3197  017460  001035                        BNE   11$               ;BR IF NO
3198  017462  005200                        INC   R0                ;BUMP COUNTER (TO 20)
3199  017464  012711  040000              MOV   #BIT14,(R1)       ;MASTER CLEAR DEVICE
3200  017470  032737  100000  001366        BIT   #BIT15,STAT1      ;DMC OR KMC?
3201  017476  001405                        BEQ   .+14              ;BR IF DMC
3202  017500  012711  100000              MOV   #BIT15,(R1)       ;SET RUN ON KMC
3203  017504  105227  000000              INCB  #0                ;DELAY ON KMC
3204  017510  001375                        BNE   .-4
3205  017512  012737  177777  020050        MOV   #-1,RESUME        ;SET RESUME FLAG
3206  017520  005711                        TST   (R1)              ;RUN SET?
3207  017522  100376                        BPL   .-2               ;BR IF NO
```

```
3208  017524  012761  000100  000002          MOV   #BIT6,2(R1)       ;SET IEO
3209  017532  032737  040000  001366          BIT   #BIT14,STAT1      ;LOOP BACK CONNECTOR?
3210  017540  001002                          BNE   .+6               ;BR IF YES
3211  017542  052711  004000                  BIS   #BIT11,(R1)       ;SET LINE UNIT LOOP
3212  017546  052711  000143                  BIS   #143,(R1)         ;ASK FOR PORT (BASE REQUEST)
3213  017552  000002                          RTI                     ;RETURN
3214  017554  016137  000004  001252  11$:    MOV   4(R1),TEMP3       ;SAVE FOR ERROR TYPEOUT
3215  017562  016137  000006  001254          MOV   6(R1),TEMP4       ;SAVE FOR ERROR TYPEOUT
3216  017570  104016                          HLT   16                ;CNTL O ERROR
3217  017572  022626                          CMP   (SP)+,(SP)+       ;ADJUST STACK
3218  017574  000137  017034                  JMP   ENDEX1            ;GET OUT
3219  017600  032761  000004  000002  1$:     BIT   #BIT2,2(R1)       ;RECEIVE?
3220  017606  001053                          BNE   2$                ;BR IF YES
3221  017610  022761  020110  000004          CMP   #TBUFF,4(R1)      ;IS XMIT BA CORRECT?
3222  017616  001412                          BEQ   4$                ;BR IF OK
3223  017620  022761  020111  000004          CMP   #TBUFF+1,4(R1)    ;IS XMIT BA CORRECT?
3224  017626  001406                          BEQ   4$                ;BR IF YES
3225  017630  012705  020110                  MOV   #TBUFF,R5         ;R5 = EXPECTED
3226  017634  016137  000004  001252          MOV   4(R1),TEMP3       ;SAVE FOUND FOR TYPEOUT
3227  017642  104002                          HLT   2                 ;XMIT BA ERROR
3228  017644  005005                  4$:     CLR   R5                ;R5 IS INDEX REG
3229  017646  026561  020240  000006  5$:     CMP   XCNTAB(R5),6(R1);IS CHAR COUNT OK?
3230  017654  001406                          BEQ   6$                ;BR IF YES
3231  017656  062705  000002                  ADD   #2,R5             ;INC INDEX
3232  017662  022705  000016                  CMP   #16,R5            ;DONE LIST YET?
3233  017666  001367                          BNE   5$                ;BR IF NO
3234  017670  104003                          HLT   3                 ;XMIT COUNT ERROR
3235  017672  016162  000004  020312  6$:     MOV   4(R1),XDNTAB(R2);STORE XMIT DONE BA
3236  017700  062702  000002                  ADD   #2,R2             ;INC INDEX
3237  017704  016162  000006  020312          MOV   6(R1),XDNTAB(R2);STORE XMIT DONE CC
3238  017712  062702  000002                  ADD   #2,R2             ;INC INDEX
3239  017716  142761  000207  000002          BICB  #207,2(R1)        ;CLEAR RDO
3240  017724  000002                          RTI                     ;RETURN
3241  017726  105011                  10$:    CLRB  (R1)              ;CLEAR SEL0
3242  017730  105061  000002                  CLRB  2(R1)             ;CLEAR SEL2
3243  017734  000002                          RTI                     ;RETURN
3244  017736  012705  000002          2$:     MOV   #2,R5             ;SET UP R5 AS INDEX
3245  017742  026561  020052  000004          CMP   RECBA(R5),4(R1)   ;COMPARE WITH LIST OF CORRECT BA'S
3246  017750  001406                          BEQ   3$                ;BR IF OK?
3247  017752  062705  000002                  ADD   #2,R5             ;INCREMENT R5
3248  017756  022705  000020                  CMP   #20,R5            ;END OF LIST?
3249  017762  001367                          BNE   2$+4              ;BR IF NO
3250  017764  104004                          HLT   4                 ;REC BA ERROR
3251  017766  005005                  3$:     CLR   R5                ;R5 IS INDEX
3252  017770  026561  020240  000006  7$:     CMP   XCNTAB(R5),6(R1);CHECK FOR CORRECT REC COUNT
3253  017776  001406                          BEQ   8$                ;BR IF YES
3254  020000  062705  000002                  ADD   #2,R5             ;INCREMENT R5
3255  020004  022705  000016                  CMP   #16,R5            ;END OF LIST?
3256  020010  001367                          BNE   7$                ;BR IF NOT
3257  020012  104005                          HLT   5                 ;REC COUNT ERROR
3258  020014  016164  000004  020256  8$:     MOV   4(R1),RDNTAB(R4)  ;STORE REC BA
3259  020022  062704  000002                  ADD   #2,R4             ;INC INDEX
3260  020026  016164  000006  020256          MOV   6(R1),RDNTAB(R4);STORE REC DONE CC
3261  020034  062704  000002                  ADD   #2,R4             ;INC INDEX
3262  020040  142761  000207  000002          BICB  #207,2(R1)        ;CLEAR RDO
3263  020046  000002                          RTI                     ;RETURN
```

```
                                        ;BUFFERS

020050  000000                RESUME: 0
020052                      RECBA:
020052  000017              XMITBA: .BLKW   17      ;REC & XMIT BA LIST

020110                      TBUFF:                  ;TRANSMIT DATA
020110    000     001     002   .BYTE   0,1,2,3,4,5,6,7
020113    003     004     005
020116    006     007
020120    010     011     012   .BYTE   10,11,12,13,14,15,16,17
020123    013     014     015
020126    016     017
020130    020     021     022   .BYTE   20,21,22,23,24,25,26,27
020133    023     024     025
020136    026     027
020140    030     031     032   .BYTE   30,31,32,33,34,35,36,37
020143    033     034     035
020146    036     037
020150    040     041     042   .BYTE   40,41,42,43,44,45,46,47
020153    043     044     045
020156    046     047
020160    050     051     052   .BYTE   50,51,52,53,54,55,56,57
020163    053     054     055
020166    056     057
020170    060     061     062   .BYTE   60,61,62,63,64,65,66,67
020173    063     064     065
020176    066     067
020200    070     071     072   .BYTE   70,71,72,73,74,75,76,77
020203    073     074     075
020206    076     077
020210    100     101     102   .BYTE   100,101,102,103,104,105,106,107
020213    103     104     105
020216    106     107

020220  000010              RCNTAB: .BLKW   10              ;RECEIVE COUNT TABLE
020240  000007              XCNTAB: .BLKW   7               ;TRANSMIT COUNT TABLE

020256  000016              RDNTAB: .BLKW   16              ;RECEIVE DONE TABLE (BA/CC)
020312  000016              XDNTAB: .BLKW   16              ;XMIT DONE TABLE (BA/CC)

020346                      RBUFF:                          ;RECEIVER BUFFERS
020346  000104              RBUFF1:  .BLKB 104
020452  000104              RBUFF2:  .BLKB 104
020556  000104              RBUFF3:  .BLKB 104
020662  000104              RBUFF4:  .BLKB 104
020766  000104              RBUFF5:  .BLKB 104
021072  000104              RBUFF6:  .BLKB 104
021176  000104              RBUFF7:  .BLKB 104
021302  000000              RBUFFE: 0                       ;END OF RECEIVER BUFFERS


                                        ;BUFFER AREA
```

```
                                                ;-----------

                                FLAG:   0
                                TFLAG:  0
                                RFLAG:  0
                                TCOUNT: 44
021314  041101  042103  043105  TBUF:   .ASCII/ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/
021322  044107  045111  046113
021330  047115  050117  051121
021336  052123  053125  054127
021344  055131  030460  031462
021352  032464  033466  034470
                                .EVEN
021360  000044                RCOUNT: 44
021362  021430                RBUF:   .=.+46
                                .EVEN
021430  022030                BASE:   .=.+256.


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


022030                          BASELD:
                                        ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
                                        ;AND PUTS DMC INTO FULL-DUPLEX MODE

022030  012711  040000                    MOV     #BIT14,(R1)     ;MASTER CLEAR
022034  032737  100000  001366            BIT     #BIT15,STAT1    ;CRAM?
022042  001402                            BEQ     .+6             ;BR IF NO
022044  012711  100000                    MOV     #BIT15,(R1)     ;IF CRAM SET RUN
022050  105227  000000                    INCB    #0              ;DELAY
022054  001375                            BNE     .-4             ;BR IF NOT DONE DELAY
022056  005711                  1$:       TST     (R1)            ;IS RUN SET?
022060  100376                            BPL     1$              ;BR IF NO
022062  052711  004000                    BIS     #BIT11,(R1)     ;SET LU LOOP
022066  152711  000043                    BISB    #43,(R1)        ;BASE REQUEST
022072  105711                  2$:       TSTB    (R1)            ;RDY I SET?
022074  100376                            BPL     2$              ;BR IF NO
022076  012761  021430  000004            MOV     #BASE,4(R1)     ;LOAD BASE ADDRESS
022104  005061  000006                    CLR     6(R1)           ;CLEAR CC
022110  142711  000040                    BICB    #40,(R1)        ;CLEAR RQI
022114  105711                  3$:       TSTB    (R1)            ;RDY I CLEAR?
022116  100776                            BMI     3$              ;BR IF NO
022120  152711  000041          BISB      #41,(R1)        ;ASK FOR CNTL I
022124  105711          64$:    TSTB      (R1)            ;WAIT FOR RDI
022126  100376          BPL     64$                       ;BR IF NOT SETY
022130  005061  000006  CLR     6(R1)                     ;SET FULL DUPLEX
022134  142711  000040  BICB    #40,(R1)                  ;CLEAR RQI
022140  105711          65$:    TSTB      (R1)            ;RDI UP?
022142  100776          BMI     65$                       ;BR IF YES
022144  000207          RTS     PC                        ;RETURN

022146                          BASELH:
                                        ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
                                        ;AND PUTS DMC INTO HALF-DUPLEX MODE
```

```
3376
3377  022146  012711  040000                        MOV     #BIT14,(R1)     ;MASTER CLEAR
3378  022152  032737  100000  001366                BIT     #BIT15,STAT1    ;CRAM?
3379  022160  001402                                BEQ     .+6             ;BR IF NO
3380  022162  012711  100000                        MOV     #BIT15,(R1)     ;IF CRAM SET RUN
3381  022166  105227  000000                        INCB    #0              ;DELAY
3382  022172  001375                                BNE     .-4             ;BR IF NOT DONE DELAY
3383  022174  005711                        1$:     TST     (R1)            ;IS RUN SET?
3384  022176  100376                                BPL     1$              ;BR IF NO
3385  022200  052711  004000                        BIS     #BIT11,(R1)     ;SET LU LOOP
3386  022204  152711  000043                        BISB    #43,(R1)        ;BASE REQUEST
3387  022210  105711                        2$:     TSTB    (R1)            ;RDY I SET?
3388  022212  100376                                BPL     2$              ;BR IF NO
3389  022214  012761  021430  000004                MOV     #BASE,4(R1)     ;LOAD BASE ADDRESS
3390  022222  005061  000006                        CLR     6(R1)           ;CLEAR CC
3391  022226  142711  000040                        BICB    #40,(R1)        ;CLEAR RQI
3392  022232  105711                        3$:     TSTB    (R1)            ;RDY I CLEAR?
3393  022234  100776                                BMI     3$              ;BR IF NO
3394  022236  152711  000041                BISB    #41,(R1)        ;ASK FOR CNTL I
3395  022242  105711                64$:    TSTB    (R1)            ;WAIT FOR RDI
3396  022244  100376                        BPL     64$             ;BR IF NOT SETY
3397  022246  012761  002000  000006        MOV     #BIT10,6(R1)    ;SET HALF DUPLEX
3398  022254  142711  000040                BICB    #40,(R1)        ;CLEAR RQI
3399  022260  105711                65$:    TSTB    (R1)            ;RDI UP?
3400  022262  100776                        BMI     65$             ;BR IF YES
3401  022264  000207                        RTS     PC              ;RETURN
3402
3403  022266                        RESUM:
3404                                        ;THIS SUBROUTINE LOADS THE DMC WITH A BASE ADDRESS
3405                                        ;WITH RESUME BIT SET AND PUTS DMC INTO FULL-DUPLEX MODE
3406
3407  022266  012711  040000                        MOV     #BIT14,(R1)     ;MASTER CLEAR
3408  022272  032737  100000  001366                BIT     #BIT15,STAT1    ;CRAM?
3409  022300  001402                                BEQ     .+6             ;BR IF NO
3410  022302  012711  100000                        MOV     #BIT15,(R1)     ;IF CRAM SET RUN
3411  022306  105227  000000                        INCB    #0              ;DELAY
3412  022312  001375                                BNE     .-4             ;BR IF NOT DONE DELAY
3413  022314  005711                        1$:     TST     (R1)            ;IS RUN SET?
3414  022316  100376                                BPL     1$              ;BR IF NO
3415  022320  052711  004000                        BIS     #BIT11,(R1)     ;SET LU LOOP
3416  022324  152711  000043                        BISB    #43,(R1)        ;BASE REQUEST
3417  022330  105711                        2$:     TSTB    (R1)            ;RDY I SET?
3418  022332  100376                                BPL     2$              ;BR IF NO
3419  022334  012761  021430  000004                MOV     #BASE,4(R1)     ;LOAD BASE ADDRESS
3420  022342  012761  010000  000006                MOV     #BIT12,6(R1)    ;SET RESUME BIT
3421  022350  142711  000040                        BICB    #40,(R1)        ;CLEAR RQI
3422  022354  105711                        3$:     TSTB    (R1)            ;RDY I CLEAR?
3423  022356  100776                                BMI     3$              ;BR IF NO
3424  022360  152711  000041                BISB    #41,(R1)        ;ASK FOR CNTL I
3425  022364  105711                64$:    TSTB    (R1)            ;WAIT FOR RDI
3426  022366  100376                        BPL     64$             ;BR IF NOT SETY
3427  022370  005061  000006                CLR     6(R1)           ;SET FULL DUPLEX
3428  022374  142711  000040                BICB    #40,(R1)        ;CLEAR RQI
3429  022400  105711                65$:    TSTB    (R1)            ;RDI UP?
3430  022402  100776                        BMI     65$             ;BR IF YES
3431  022404  000207                        RTS     PC              ;RETURN
```

```
3432
3433  022406                                  RFRELD:
3434                                                  ;THIS SUBROUTINE LOADS THE DMC WITH A RECEIVE BA/CC
3435
3436  022406  152711  000044                        BISB    #44,(R1)        ;REC BA/CC REQUEST
3437  022412  105711                          1$:     TSTB    (R1)            ;RDY I SET?
3438  022414  100376                                  BPL     1$              ;BR IF NO
3439  022416  012561  000004                        MOV     (R5)+,4(R1)     ;LOAD REC BA
3440  022422  012561  000006                        MOV     (R5)+,6(R1)     ;LOAD REC CC
3441  022426  142711  000040                        BICB    #40,(R1)        ;CLEAR RQI
3442  022432  105711                          2$:     TSTB    (R1)            ;IS RDY I CLEAR
3443  022434  100776                                  BMI     2$              ;BR IF NO
3444  022436  000205                                  RTS     R5              ;RETURN
3445
3446  022440                                  XFRELD:
3447                                                  ;THIS SUBROUTINE LOADS THE DMC WITH A TRANSMIT BA/CC
3448
3449  022440  152711  000040                        BISB    #40,(R1)        ;XMIT BA/CC REQUEST
3450  022444  105711                          1$:     TSTB    (R1)            ;RDY I SET?
3451  022446  100376                                  BPL     1$              ;BR IF NO
3452  022450  012561  000004                        MOV     (R5)+,4(R1)     ;LOAD XMIT BA
3453  022454  012561  000006                        MOV     (R5)+,6(R1)     ;LOAD XMIT CC
3454  022460  142711  000040                        BICB    #40,(R1)        ;CLEAR RQI
3455  022464  105711                          2$:     TSTB    (R1)            ;IS RDY I CLEAR
3456  022466  100776                                  BMI     2$              ;BR IF NO
3457  022470  000205                                  RTS     R5              ;RETURN
3458
3459
3460  022472                                  SHUTDOWN:
3461                                                  ;THIS SUBROUTINE FORCES THE DMC TO UPDATE THE BASE TABLE
3462
3463  022472  042761  000207  000002                BIC     #207,2(R1)      ;CLEAR ANY OUTPUT DONES
3464  022500  152711  000046                        BISB    #46,(R1)        ;ASK FOR ILLEGAL REQUEST
3465  022504  105711                          1$:     TSTB    (R1)            ;RDI SET?
3466  022506  100376                                  BPL     1$              ;BR IF NO
3467  022510  142711  000040                        BICB    #40,(R1)        ;CLEAR RQ;
3468  022514  105761  000002                2$:     TSTB    2(R1)           ;OUTPUT DONE SET?
3469  022520  100375                                  BPL     2$              ;BR IF NOT
3470  022522  000207                                  RTS     PC              ;RETURN
3471
3472
      022524  052377  040522  051516          EM2:    .ASCIZ  <377>/TRANSMIT BA ERROR/
      022547     377  051124  047101          EM3:    .ASCIZ  <377>/TRANSMIT COUNT ERROR/
      022575     377  042522  042503          EM4:    .ASCIZ  <377>/RECEIVE BA ERROR/
      022617     377  042522  042503          EM5:    .ASCIZ  <377>/RECEIVE COUNT ERROR/
      022644  051377  041505  044505          EM11:   .ASCIZ  <377>/RECEIVE DATA ERROR/
```

```
        022670  043377  042522  020105          EM12:   .ASCIZ  <377>/FREE RUNNING ERROR/
```

```
022714  041777  047117  051124          EM13:   .ASCIZ  <377>/CONTROL OUT ERROR/
022737     377  047111  042524          EM14:   .ASCIZ  <377>/INTERNAL DDCMP ERROR COUNTS NON ZERO/

023005     377  054105  042520          DH1:    .ASCIZ  <377>/EXPECTED  FOUND  ADDRESS/
023037     377  054105  042520          DH2:    .ASCIZ  <377>/EXPECTED  FOUND/
023060  020377  042523  032114          DH3:    .ASCIZ  <377>/ SEL4      SEL6/
023101     377  040502  042523          DH4:    .ASCIZ  <377>/BASE+3 THRU BASE+12 /
023127     377  046504  030503          DH5:    .ASCIZ  <377>/DMC11 IS HUNG/
                                         .EVEN

023146  000003                          DT1:    3
023150     006     004                          .BYTE   6,4
023152  001264                                  SAVR2
023154     006     004                          .BYTE   6,4
023156  001270                                  SAVR4
023160     004     002                          .BYTE   4,2
023162  001260                                  SAVR0
023164  000003                          DT2:    3
023166     006     004                          .BYTE   6,4
023170  001272                                  SAVR5
023172     006     004                          .BYTE   6,4
023174  001270                                  SAVR4
023176     004     002                          .BYTE   4,2
023200  001264                                  SAVR2
023202  000003                          DT3:    3
023204     006     004                          .BYTE   6,4
023206  001272                                  SAVR5
023210     006     004                          .BYTE   6,4
023212  001270                                  SAVR4
023214     004     002                          .BYTE   4,2
023216  001252                                  TEMP3
023220  000002                          DT4:    2
023222     003     007                          .BYTE   3,7
```

```
023224  001272                          SAVR5
023226     003      002                 .BYTE   3,2
023230  001270                          SAVR4
023232  000002                  DT5:    2
023234     006      004                 .BYTE   6,4
023236  001272                          SAVR5
023240     006      002                 .BYTE   6,2
023242  001270                          SAVR4
023244  000003                  DT6:    3
023246     003      010                 .BYTE   3,10
023250  001272                          SAVR5
023252     003      004                 .BYTE   3,4
023254  001270                          SAVR4
023256     004      002                 .BYTE   4,2
023260  021304                          FLAG
023262  000003                  DT7:    3
023264     003      010                 .BYTE   3,10
023266  001272                          SAVR5
023270     003      004                 .BYTE   3,4
023272  001270                          SAVR4
023274     004      002                 .BYTE   4,2
023276  001264                          SAVR2
023300  000003                  DT10:   3
023302     003      007                 .BYTE   3,7
023304  001272                          SAVR5
023306     003      004                 .BYTE   3,4
023310  001270                          SAVR4
023312     006      002                 .BYTE   6,2
023314  001252                          TEMP3
023316  000002                  DT11:   2
023320     006      004                 .BYTE   6,4
023322  001252                          TEMP3
023324     006      002                 .BYTE   6,2
023326  001254                          TEMP4
023330  000010                  DT12:   10
023332     003      002                 .BYTE   3,2
023334  001250                          TEMP2
023336     003      002                 .BYTE   3,2
023340  021434                          BASE+4
023342     003      002                 .BYTE   3,2
023344  001252                          TEMP3
023346     003      002                 .BYTE   3,2
023350  021436                          BASE+6
023352     003      002                 .BYTE   3,2
023354  001254                          TEMP4
023356     003      002                 .BYTE   3,2
023360  021440                          BASE+10
023362     003      002                 .BYTE   3,2
023364  001256                          TEMP5
023366     003      002                 .BYTE   3,2
023370  021442                          BASE+12
023372  000002                  DT13:   2
023374     006      004                 .BYTE   6,4
023376  001272                          SAVR5
023400     006      002                 .BYTE   6,2
023402  001270                          SAVR4
```

```
023404                                  .ERRTAB:
023404  000000                                0
023406  000000                                0
023410  000000                                0
023412  022670                          EM12
023414  023127                          DH5     ;HLT    1
023416  000000                          0
023420  022524                          EM2
023422  023037                          DH2     ;HLT    2
023424  023372                          DT13
023426  022547                          EM3
023430  000000                          0       ;HLT    3
023432  000000                          0
023434  022575                          EM4
023436  000000                          0       ;HLT    4
023440  000000                          0
023442  022617                          EM5
023444  000000                          0
023446  000000                          0
023450  022575                          EM4
023452  023037                          DH2     ;HLT    6
023454  023232                          DT5
023456  022617                          EM5
023460  023037                          DH2     ;HLT    7
023462  023220                          DT4
023464  000000                          0
023466  023005                          DH1     ;HLT    10
023470  023244                          DT6
023472  000000                          0
023474  023005                          DH1     ;HLT    11
023476  023262                          DT7
023500  000000                          0
023502  023037                          DH2     ;HLT    12
023504  023220                          DT4
023506  022644                          EM11
023510  023005                          DH1     ;HLT    13
023512  023300                          DT10
023514  022670                          EM12
023516  000000                          0       ;HLT    14
023520  000000                          0
023522  022670                          EM12
023524  023037                          DH2     ;HLT    15
023526  023232                          DT5
023530  022714                          EM13
023532  023060                          DH3     ;HLT    16
023534  023316                          DT11
023536  022737                          EM14
023540  023101                          DH4     ;HLT    17
023542  023330                          DT12


023544                                  CORMAX:
        000001                          .END
```

CZDMH MACY11 30A(1052) 08-JUL-80 08:21 PAGE 75
CZDMH.P11 08-JUL-80 08:21 SYMBOL TABLE SEQ 0073

```
ADRCNT= 004373
AUDONE  003024
AUSTRT  002446
AUTO.S  010512
BASE    021430
BASELD  022030
BASELH  022146
BINWRD  004714
BIT0  = 000001
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BM      007054
BRLVL   012252
BRW     003730
BRX     003732
CHRCNT  004712
CKSWR   007606
CKSWR1  007666
CKSWR2  007700
CKSWR3  007704
CKSWR4  007710
CKSWR5  010014
CLKX    001242
CLRTAB  016444
CNERR   007277
CNT.MA  001702
CNVRT = 104411
CONERR  007223
CONN    007114
CONTAB  002776
CONVRT= 104410
CORMAX  023544
CRAM    006606
CREAM   001320
CSR     006510
CSRMAP  010514
CYCLE   010060
DATABP  005216
DATACL= 104415
DATAHD  005204
DELAY = 104413
DEVADR  004370
DEVTAB  003010
DH1     023005
DH2     023037
DH3     023060
DH4     023101
DH5     023127
DISPLA  001200
DISPRE  000174
DMACTV  001306
DMCM    007320
DMCR00  001500
DMCR01  001510
DMCR02  001520
DMCR03  001530
DMCR04  001540
DMCR05  001550
DMCR06  001560
DMCR07  001570
DMCR10  001600
DMCR11  001610
DMCR12  001620
DMCR13  001630
DMCR14  001640
DMCR15  001650
DMCR16  001660
DMCR17  001670
DMCSR   001404
DMCSRH  001406
DMCTL   001410
DMNUM   001310
DMP04   001412
DMP06   001414
DMRLVL  001376
DMRVEC  001374
DMS100  001502
DMS101  001512
DMS102  001522
DMS103  001532
DMS104  001542
DMS105  001552
DMS106  001562
DMS107  001572
DMS110  001602
DMS111  001612
DMS112  001622
DMS113  001632
DMS114  001642
DMS115  001652
DMS116  001662
DMS117  001672
DMS200  001504
DMS201  001514
DMS202  001524
DMS203  001534
DMS204  001544
DMS205  001554
DMS206  001564
DMS207  001574
DMS210  001604
DMS211  001614
DMS212  001624
DMS213  001634
DMS214  001644
DMS215  001654
DMS216  001664
DMS217  001674
DMS300  001506
DMS301  001516
DMS302  001526
DMS303  001536
DMS304  001546
DMS305  001556
DMS306  001566
DMS307  001576
DMS310  001606
DMS311  001616
DMS312  001626
DMS313  001636
DMS314  001646
DMS315  001656
DMS316  001666
DMS317  001676
DMTLVL  001402
DMTVEC  001400
DM.END  001700
DM.MAP  001500
DONE    003734
DT1     023146
DT10    023300
DT11    023316
DT12    023330
DT13    023372
DT2     023164
DT3     023202
DT4     023220
DT5     023232
DT6     023244
DT7     023262
EM11    022644
EM12    022670
EM13    022714
EM14    022737
EM2     022524
EM3     022547
EM4     022575
EM5     022617
ENDEX   016740
ENDEX1  017034
ENDEX2  017036
ERCT00  001704
ERCT01  001710
ERCT02  001714
ERCT03  001720
ERCT04  001724
ERCT05  001730
ERCT06  001734
ERCT07  001740
ERCT10  001744
ERCT11  001750
ERCT12  001754
ERCT13  001760
ERCT14  001764
ERCT15  001770
ERCT16  001774
ERCT17  002000
ERR     002700
ERRCNT  001232
ERRFLG  001325
ERRMSG  005172
ERRPC   002770
ERTABO  005322
EXIT  = 000205
EXITER  005252
FLAG    021304
FLOAT   002536
FY      002566
HALTS   005222
HILIM   004366
ICOUNT  001222
IISR    017122
INBUF   007502
INCHAR  010020
INIFLG  001324
INSTER= 104404
INSTR = 104403
INSTR2  004166
INTTY   012266
KMCM    007330
LIMITS  004314
LINE    007016
LOBITS  004372
LOCK    001220
LOKFLG  001326
LOLIM   004364
LPCNT   001224
LSTERR  001234
MASKX   001244
MASTEK  006142
MCRLF   005672
MCSRX   006072
MDATA   007544
MEMLIM  001304
MEPASS  005733
MERRPC  006217
MERRX   006117
MERR2   005760
MERR3   006005
MILK    001322
MLOCK   006043
MNEW    006144
MODU    006704
MPASSX  006106
MPFAIL  005675
MQM     005666
MR      005755
MRESET= 004000
MSTCLR= 104412
MTITLE  001000
MTSTN   006130
MTSTPC  006031
MVECX   006100
NEXT    001216
NOACT   007154
NODEV   002674
NUM     006450
OISR    017412
OK      002646
ONE     001302
PACT00  001702
PACT01  001706
PACT02  001712
PACT03  001716
PACT04  001722
PACT05  001726
PACT06  001732
PACT07  001736
PACT10  001742
PACT11  001746
PACT12  001752
PACT13  001756
PACT14  001762
PACT15  001766
PACT16  001772
PACT17  001776
PARAM = 104405
PARAM1  004234
PARBIT= 040000
PARERR  004310
PASCNT  001230
PERFOR= 004537
```

```
CZDMH   MACY11 30A(1052)  08-JUL-80  08:21  PAGE 76
CZDMH.P11    08-JUL-80 08:21              SYMBOL TABLE                                                    SEQ 0074

PFTAB   005430          SAVACT  001312          SW02  = 000004          TST12   016204          X5    = 000105
POPRO = 012600          SAVNUM  001314          SW03  = 000010          TST2    013404          X6    = 000106
POP1SP= 005726          SAVPC   001276          SW04  = 000020          TST3    013744          X7    = 000107
POP2SP= 022626          SAVR0   001260          SW05  = 000040          TST4    014226          ZERO    001300
PRIO    006547          SAVR1   001262          SW06  = 000100          TST5    014500          $CRAP = 177777
PS    = 177776          SAVR2   001264          SW07  = 000200          TST6    014762          $ENDAD  003522
PUSHR0= 010046          SAVR3   001266          SW08  = 000400          TST7    015224          $N    = 000012
PUSH1S= 005746          SAVR4   001270          SW09  = 001000          TTST    003612          $S    = 000014
PUSH2S= 024646          SAVR5   001272          SW10  = 002000          TWOSYN= 010000          $Y    = 000017
QV.FLG  001327          SAVSP   001274          SW11  = 004000          TYPDAT  005206          .     = 023544
RBUF    021362          SAV05 = 104406          SW12  = 010000          TYPE  = 104402          .BEGIN  003152
RBUFF   020346          SCAN    016574          SW13  = 020000          TYPMSG  005106          .CNVRT  004472
RBUFFE  021302          SCAN1   017072          SW14  = 040000          VEC     006526          .CONVR  004466
RBUFF1  020346          SCAN2   016606          SW15  = 100000          VECMAP  012010          .DATAC  005552
RBUFF2  020452          SCOPE = 104400          TBUF    021314          WHICH   012002          .DELAY  005436
RBUFF3  020556          SCOP1 = 104401          TBUFF   020110          WRDCNT  004710          .EOP    003364
RBUFF4  020662          SHUTDO  022472          TCOUNT  021312          WRKO.F  005174          .ERRTA  023404
RBUFF5  020766          SKIP    002632          TEMP    001416          XBX     005000          .HLT    004750
RBUFF6  021072          SOFTSW  010052          TEMP1   001246          XCNTAB  020240          .INSTE  004154
RBUFF7  021176          SPACNT= 004713          TEMP2   001250          XCSR    003546          .INSTR  004050
RCNTAB  020220          SPEED   007340          TEMP3   001252          XDNTAB  020312          .INST1  004070
RCOUNT  021360          STACK = 001200          TEMP4   001254          XERR    003570          .MSG    004072
RDNTAB  020256          STAT    001240          TEMP5   001256          XFRELD  022440          .MSTCL  005466
RECBA   020052          STAT1   001366          TFLAG   021306          XHEAD   006224          .PARAM  004174
RESREG  005220          STAT2   001370          TIMER = 104416          XLOC    003022          .PFAIL  005336
RESTAR  005350          STAT3   001372          TKCSR   001204          XMITBA  020052          .RES05  004434
RESTRT  003534          STRTSW  001236          TKDBR   001206          XPASS   003562          .ROMCL  005504
RESUM   022266          SV05    004402          TLAST = 016204          XSTATQ  007454          .SAV05  004374
RESUME  020050          SWFLG   010016          TPCSR   001210          XTSTN   005330          .SCOPE  003576
RES05 = 104407          SWMES   007205          TPDBR   001212          XVEC    003554          .SCOP1  003736
RETURN  001214          SWMES1  007215          TRPOK   004730          X0    = 000110          .START  002002
RFLAG   021310          SWR     001202          TSTNO   001226          X1    = 000101          .TIMER  005616
RFRELD  022406          SWREG   000176          TST1    012320          X2    = 000102          .TRPSR  004716
ROMCLK= 104414          SW00  = 000001          TST10   015466          X3    = 000103          .TRPTA  001330
RUN     001316          SW01  = 000002          TST11   015622          X4    = 000104          .TYPE   003766


. ABS.  023544      000


 ERRORS DETECTED:  0

 CZDMH,CZDMH/NL:TOC/SOL=CZDMH.MAC,CZDMH.P11
 RUN-TIME: 7 11 .3 SECONDS
 RUN-TIME RATIO: 28/19=1.4
 CORE USED:  25K  (49 PAGES)
```