DQ11

BASIC LOGIC TEST PART 2

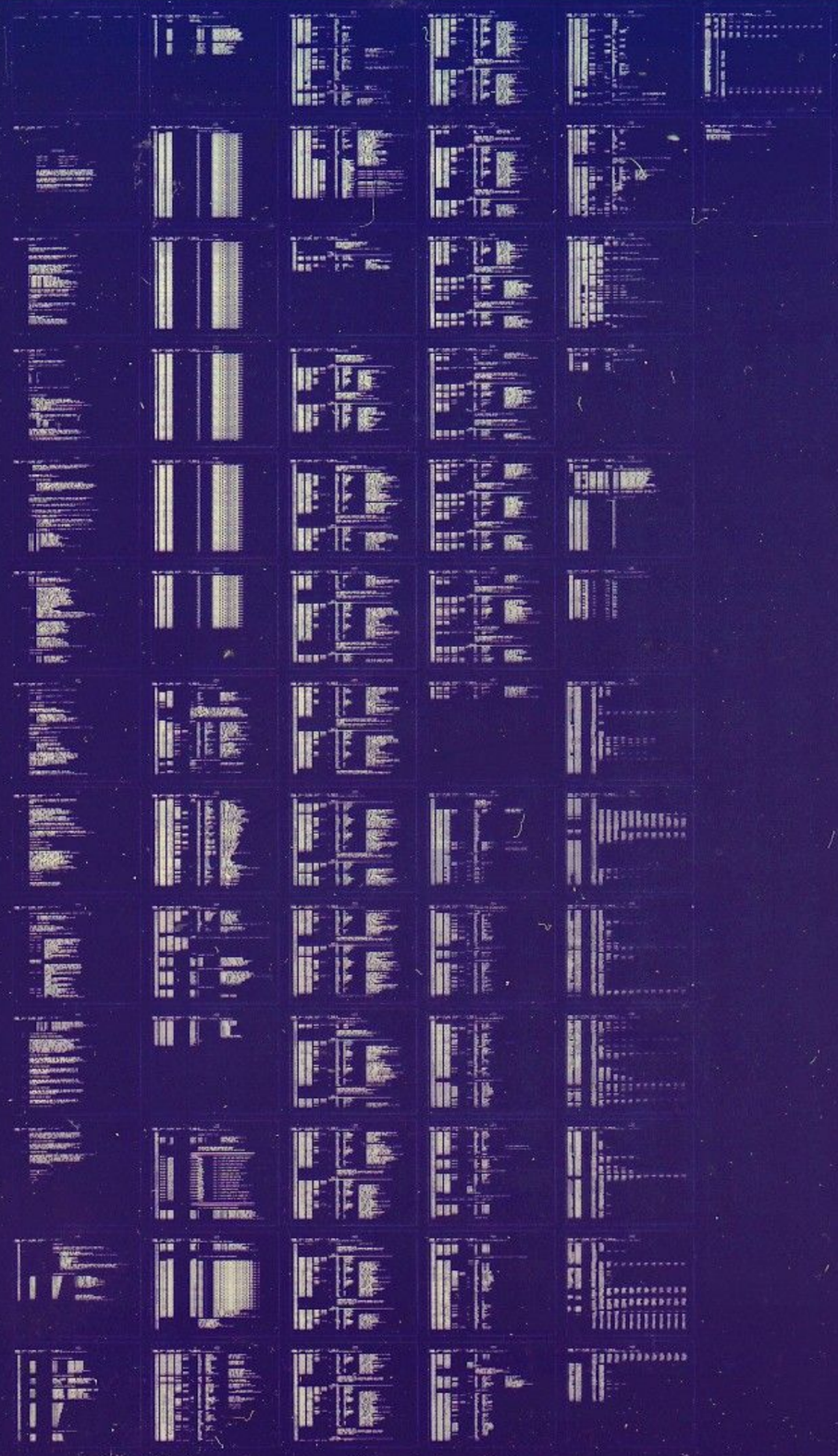
MD-11-DZDQB-C

EP-DZDQB-C-DL-B APR 1977

FICHE 1 OF 1

digital

MADE IN USA



B01

EOF1DZDQACSEQ	00010000	770325	PDP10 411	LEHDR1DZDQBCSEQ	00010000	770325

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDQB-C-D

PRODUCT NAME: BASIC LOGIC TEST PART 2

DATE: MARCH 1977

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

THE FUNCTION OF THE DQ11 DIAGNOSTICS ARE TO
VERIFY THAT THE OPTION OPERATES ACCORDING
TO SPECIFICATIONS.

THIS TEST CHECKS MASTER CLEAR,READ WRITE OF SECONDARY
REGISTERS, AND THE WRITE ENABLE FUNCTION. THIS IS
PART TWO OF THE READ WRITE TESTS.

CURRENTLY THERE ARE SEVEN OFF LINE DIAGNOSTICS THAT ARE
TO BE RUN IN SEQUENCE TO INSURE THAT IF AN
ERROR SHOULD OCCUR IT WILL BE DETECTED
AT AN EARLY STAGE AND INSURING THAT
DIAGNOSIS OF ERROR WILL BE IMMEDIATE TO PROBLEM
NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE SEVEN DIAGNOSTICS ARE:
1. DZDQA [REV] BASIS R/W TEST #1
2. DZDQB [REV] BASIC R/W TEST #2
3. DZDQC [REV] BASIC NPR AND INTERUPT TEST
4. DZDQD [REV] RECEIVER TRANSMITTER EXERCISER TEST
5. DZDQE [REV] MISC. RX AND TX TESTS, PLUS BCC TESTS.
6. DZDQF [REV] CHARACTER DETECT TESTS.
7. DZDQH [REV] CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSED LATER.
1. DZDQO [REV] ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE
1. DZDQG [REV] DQ11 TRIAL PROGRAM (PARAMETER INPUT)

2. REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 4K MEMORY)-WITH
 OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570)
ASR 33 (OR EQUILIVALENT)
DQ11
SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

2.2 STORAGE

PROGRAM WILL LOAD AND RUN
IN 4K OF MEMORY.
LOCATION 1400 THRU 1600 ARE ESPECIALLY TO
BE NOTED AND TO BE UNTOUCHED BY OPERATOR
AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED.
OR AFTER THE "AUTO SIZING" HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND
ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS *500

MEMORY *
SIZE

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

3.1.1 LOAD THE ADDRESS OF ABS. LOADER (LOC.XXX500)

3.1.2 THEN START

4. STARTING PROCEEDURE

A. LOAD LOC. 200
B. SET SWR TO ZERO FOR "AUTO SIZING" OR LEAVE
   LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP
   BY DQ11 TRIAL PROGRAM OR A PREVIOUSLY RUN DQ11 DIAGNOSTIC
   THAT USED THE "AUTO SIZING".
    ****REFER TO SECTION 4.1 FOR SOFTWARE SWITCH REGISTER OPERATION
        AND OPTIONS.****
   NOTE:THE SOFTWARE SWITCH REGISTER IS LOCATED AT LOC.176
   SOFTWARE DISPLAY REGISTER IS LOCATED AT LOC.174

C.THEN START
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 DQ11 STATUS"
        1400   160010
        1402   152300
        1404   160020
        1406   150310

THE ABOVE IS ONLY AN EXAMPLE!
THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD.
1400 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE
USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS
TABLE SEE SECTION 8.4 FOR HELP.

****IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
    WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:

SWR=XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR'S OPTION)****
NOTE:IF USING THE SOFTWARE SWITCH REGISTER WHEN A HARDWARE
SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL NOT
TYPE OUT THE TITLE.

THE PROGRAM WILL TYPE "R"
AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G); THIS WILL ALLOW THE TTY TO ENTER DATA INTO
   LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS
   OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
   OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A (CR).
      (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
      WILL BE ALLOWED)
      IF A (CR) IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
      REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U (↑U) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
      BACK TO STEP 2.

SW 15 SET: HALT ON ERROR
SW 14 SET: LOOP ON CURRENT TEST
SW 13 SET: INHIBIT ERROR PRINT OUT
SW 12 SET: INHIBIT TYPE OUT/BELL ON ERROR.
SW 11 SET: INHIBIT ITERATIONS
SW 10 SET: ESCAPE TO NEXT TEST
SW 09 SET: LOOP WITH CURRENT DATA
SW 08 SET: CATCH ERROR AND LOOP ON IT
SW 07 SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.
SW 06 SET:
SW 05 SET:
SW 04 SET:

SW 03 SET:
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: RESELECT DQ11'S DESIRED ACTIVE.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DQ11'S DESIRED ACTIVE.
PLEASE NOTE THAT A MESSAGE IS TYPED
OUT FOR SWITCH REGISTER BEING EQUAL TO DQ11'S
ACTIVE. THIS MEANS IF THE SYSTEM HAS
FOUR DQ11S; BITS 00,01,02,03 WILL
BE SET IN LOC "DQACTV". USING THIS
SWITCH ALTERS THAT LOCATION;THEREFORE
IF FOUR DQ11S 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
DQ11S THAN HAS BEEN GIVEN INFORMATION
ABOUT IN TRIAL PROGRAM.

METHOD:
A: LOAD ADDRESS 200
B: START WITH SW 00=1
C: PROGRAM WILL TYPE MESSAGE
D: CONTINUE THE BINARY NUMBER OF DQ11S DESIRED ACTIVE
EXAMPLE: 1=1 DQ11; 3=2 DQ11; 7=3 DQ11; 17=4 DQ11 37=5 DQ11 ETC.
E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05, 11/04, 11/34)
F: CONTINUE WITH ANY OTHER SWITCH SETTINGS DESIRED.

SW 01 IT IS STRONGLY SUGGESTED THAT
AT LEAST ONE PASS HAS BEEN MADE
BEFORE TRYING TO SELECT A TEST
THAT IS NOT IN THE ORDER OF SEQUENCE
THE REASON BEING IS THAT THE
PROGRAM HAS TO CLEAR AREAS AND SET
UP PARAMETERS. ALSO WHEN A TEST IS
SELECTED ALWAYS START AT THE VERY
BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA:
THIS SWITCH WILL ONLY WORK IF
CALL "SCOP1" IS IN THAT TEST.
THE REASON BEING THAT MOST TESTS
DEAL WITH BLOCKS OF DIFFERENT DATA
TO BE SENT OR RECEIVED ALL AT ONCE
THUS IN BLOCK DATA; ONE PATTERN CANN'T BE SINGLED OUT.

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

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

****HLT (ERROR) ROUTINE SUPPORTS <↑G> OPERATION****

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY "SCOP1")
2. SW 14
3. SW 11

****SCOPE ROUTINE WILL SUPPORT <↑G> OPERATION****

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200
THERE ARE NO OTHER STARTING ADDRESSES
FOR THE DQ11 DIAGNOSTICS PREVIOUSLY MENTIONED

NOTE: IF ADDRESS 000042 IS NON-ZERO
THE PROGRAM ASSUMES IT IS UNDER
ACT11 OR DDP CONTROL AND WILL ACT ACCORDINGLY
AFTER *ALL* AVAILABLE DQ11'S ARE TESTED
THE PROGRAM WILL RETURN TO "DDP2" OR "ACT-11".

5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION
FOUR WILL BE PRINTED.

AND PROGRAM WILL BEGIN RUNNING THE
DIAGNOSTIC

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

THE TEST NUMBER AND PC WILL BE TYPED OUT AND
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 FUNCTIONING OF THE TEST
CAN BE INTERPEDITED

6. ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAY 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 THE THE ERROR MESSAGE
WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE
ERROR.

6.2 ERROR RECOVERY

IF FOR SOME REASON THE DQ11 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 DQ11 WAS DOING AT THE TIME OF THE ERROR.

6.3 ****HALT RECOVERY WHEN USING SOFTWARE SWITCH REGISTER****

IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT
THE THE OPERATOR IS REQUIRED TO TYPE A ⟨↑G⟩ BEFORE DEPRESSING CONTINUE.
THE FOLLOWING WILL BE TYPED:
SWR=XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR OPTION)

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

DQ11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE
FIRST AND ONLY THE FIRST RUNNING OF ANY DQ11 DIAGNOSTIC
NOTE: IF NO PROGRAM OTHER THAN A
DQ11 DIAGNOSTIC WAS LOADED AFTER DQ11 TRIAL OR
IF CORE MEMORY HAS NOT BEEN CHANGED;OR IF THERE
IS NO DQ11 CONFIGURATION CHANGES; THE
DQ11 TRIAL PROGRAM NEED NEVER BE RUN AGAIN.
HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED
THE DQ11 TRIAL PROGRAM MUST BE RUN AGAIN
BEFORE RUNNING THE DIAGNOSTICS
NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING
THE "AUTO SIZING" WHEN PROGRAM IS INITIALLY STARTED
WITH SW07=0.

8. MISCELLANEOUS

8.1 EXECUTION TIME

8.2 PASS COMPLETE

WHEN THE DIAGNOSTIC HAS COMPLETED
A PASS THE FOLLOWING IS AN EXAMPLE
OF THE PRINT OUT TO BE EXPECTED.

END PASS DZDQB-C CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE
NOT NECESSARILY THE VALUES FOR THE DEVICE
THEY ARE ONLY FOR THIS EXAMPLE.

8.3 TST1 (MINI MONITOR)

THE VERY FIRST "TEST" (TST1)
IS *NOT* A TEST OF THE DQ11 HARDWARE
IT IS A MINI-MONITOR USED TO CYCLE DQ11 IN THE
SYSTEM THROUGH THE DIAGNOSTIC.

REMEMBER: TST1 IS NOT A TEST OF DQ11 HARDWARE!!!!!!!

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	(1304)	THE BIT IN "RUN" ALWAYS POINTS ONE PAST THE DQ11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) 1304/0000000001000000 MEANS THAT DQ11 NO.05 IS THE DQ11 NOW RUNNING.
DQCR00-DQCR17 DQST00-DQST17 (1400)-(1476)		THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S SEQUENTIALY. THEY CONTAIN THE CSR,VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DQ11.
DQACTV	(1500)	EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DQ11 WILL BE TESTED IN TURN. EXAMPLE: (DQACTV) 1500/0000000000011111 MEANS THAT DQ11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DQACTV) 1500/0000000000010001 MEANS THAT DQ11 NO. 00,04 WILL BE TESTED.
DQCSR	(1506)	CONTAINS THE RECEIVER CSR OF THE CURRENT DQ11 UNDER TEST.
DQSTAT	(1510)	CONTAINS THE STATUS OF THE CURRENT DQ11 UNDER TEST.
BIT 15		SET: TWO SYNC CHARS/ONE SYNC CHAR
BIT 14		SET: TEST JUMPER INSTALLED/NOT INSTALLED

BIT 13 SET: BB OPTION INSTALLED/NOT INSTALLED
BIT 12 SET: BA OPTION INSTALLED/NOT INSTALLED
BIT 11 SET: ACTIVE ON FIRST NON-SYNC/ACTIVE AFTER NO. OF SYNC
BIT 10 SET: AB OPTION INSTALLED/NOT INSTALLED
BIT 09 SET: ODD VRC/EVEN VRC
BIT 00-08 VECTOR "A" OF DEVICE

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

WHEN LOOKING FOR THE CSR IT IS NECESSARY TO TAKE CARE
THAT WHEN A CSR IS FOUND THAT IT IS INDEED A DQ11. THAT
IS THE METHOD OF MY MADNESS FOR THIS ROUTINE.
AN ATTEMPT TO CLEAR THE MISC. REGISTER IS TRIED
IF A TIME-OUT TRAP OCCURES POINTERS ARE UPDATED
AND ATTEMPTED AGAIN. IF NO TIME-OUT; THE RECEIVER "ACTIVE BIT" (BIT 12)
IS SET AND A *COMPARE* FOR BOTH SYNC1 AND SYNC 2 IS DONE
AT THE MISC. REGISTER. IF THEY ARE THERE THIS IS
A DQ11. THE INFORMATION IS STORED AWAY.

8.5.2 ONE SYNC BIT OR TWO?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE THE
PRESENTS OF ONE SYNC OR TWO. THE PROGRAM ASSUMES TWO SYNC
CHARS. NOTE: THIS ASSUMPTION MAY BE ALTERED AFTER AUTO SIZING
BY ALTERING BIT IS IN APPRIOATE DQSTXX: LOCATION.

8.5.3 "BB" OPTION INSTALLED?

TO SENSE FOR THE "BB" OPTION THE PROGRAM SELECTS THE
CHARACTER DET. REGISTER AND THE LOADS IN ALL 1'S. IF
ANY ONE OR COMBINATION OF BITS ARE SET THE BB OPTION
IS ASSUMED TO EXIST.

8.5.4 "AB" OPTION INSTALLED?

TO SENSE FOR THE "AB" OPTION THE PROGRAM SELECTS THE
POLYNOMIAL REGISTER AND WRITES ALL 1'S INTO IT; IF ANY
ONE OR COMBINATION OF BITS ARE SET THE AB OPTION IS ASSUMED
TO EXIST.

8.5.5 "BA" OPTION INSTALLED?

TO SENSE FOR "BA" OPTION REQUEST TO SEND AND DATA TERMINAL
READY ARE SET; IF EITHER ONE OR BOTH ARE SET THE PROGRAM
ASSUMES THE BA OPTION EXISTES

8.5.6 JUMPER ON END OF CABLE?

THE PROGRAM CHECKS TO SEE IF EITHER OR BOTH CLEAR TO
SEND AND CARRIER ARE SET; IF SO THE PROGRAM ASSUMES THE
TEST JUMPER IS ON THE END OF THE CABLE.

8.5.7 ACTIVE ON FIRST NON-SYNC?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE FOR
WHEN THE DQ11 GOES ACTIVE THE PROGRAM ASSUMES "ACTIVE
ON FIRST NON-SYNC". NOTE: THIS CAN BE CHANGED BY ALTERING
BIT 11 IN THE APPRIOATE DQSTXX: AFTER AUTO SIZING

8.5.8 SET FOR ODD OR EVEN PARITY?

AS ABOVE TOO MUCH HARDWARE IS NEED TO SENSE WHICH PARITY
WAS SELECTED.SO THE PROGRAM ASSEMES ODD PARITY.
NOTE: THIS CAN BE CHANGED BY ALTERING BIT 9 IN APPRIO-
ATE DQSTXX: LOCATION. AFTER AUTO SIZING

8.5.9 FINDING THE VECTOR.

THE PROGRAM SETS "PRIMARY DONE","SECONDAY DONE", AND "INTERUPT ENABLE"
AND LOOKS FOR AN INTERUPT. IF IT INTERUPTS IT IS PICKED
UP AND STORED AWAY. IF NO INTERUPT OCCURES THE PROGRAM
ASSUMES VECTOR =300. THIS PROBLEM WILL BE FIXED IN ONE
OF THE DIAGNOSTICS AND *AUTO SIZING* SHOULD BE REDONE TO
GET THE CORRECT VECTOR.

9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING

10. LISTING

FOLLOWING

```
                                .ENABLE AMA

                                ;MAINDEC-11-DZDQB-C/(377)/DQ11 STATIC LOGIC TEST-PART 2
                                ;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                ;REVISED 16-DEC-76 BY R. BLACK
                                ;       A)SUPPORTS SOFTWARE SWITCH REGISTER
                                ;       B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
                                ;         BY (↑G).
                                ;       ;STARTING PROCEDURE
                                ;       ;LOAD PROGRAM
                                ;       ;LOAD ADDRESS 000200
                                ;       ;PRESS START
                                ;       ;PROGRAM WILL TYPE "MAINDEC-11-DZDQB-C/(377)/DQ11 STATIC LOGIC TEST-PART 2"
                                ;       ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                ;       ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                ;       ;AND THEN RESUME TESTING


                                ;SWITCH REGISTER OPTIONS

100000                          SW15=100000             ;=1,HALT ON ERROR
040000                          SW14=40000              ;=1,LOOP ON CURRENT TEST
020000                          SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
010000                          SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR.
004000                          SW11=4000               ;=1,INHIBIT ITERATIONS
002000                          SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
001000                          SW09=1000               ;=1,LOOP WITH CURRENT DATA
000400                          SW08=400                ;=1,LOOP ON ERROR
000100                          SW06=100
000040                          SW05=40
000020                          SW04=20
000010                          SW03=10
000004                          SW02=4                  ;LOCK ON TEST SELECT
000002                          SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
000001                          SW00=1                  ;RESELECT DQ11 DESIRED ACTIVE
                                                        ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 13
DZDQBC.P11    16-DEC-76 13:23          GENERAL DEFINATIONS AND EQUIVALENCIES

                                ;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

177570                          DSWR=   177570  ;HARDWARE SWITCH REGISTER LOC.
177570                          DLIGHTS=177570  ;HARDWARE DISPLAY REGISTER LOC.
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


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


                                ;DQ11 OPTIONAL DEFINITIONS

002000                          ABBIT=2000
004000                          ACTBIT=4000
010000                          B?BIT=10000
020000                          B_3IT=20000
040000                          JUMBIT=40000
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 14
DZDQBC.P11    16-DEC-76 13:23          GENERAL DEFINATIONS AND EQUIVALENCIES

   618          001000                  ODDBIT=1000
   619          100000                  SYNBIT=100000
   620
   621
   622                                          ;DQ11 SECONDARY REGISTER DEFINATIONS
   623
   624          000000                  RXBA.P=0        ;RECEIVER BUS ADDRESS PRIMARY.
   625          000001                  RXWC.P=1        ;RECEIVER WORD COUNT PRIMARY.
   626          000002                  TXBA.P=2        ;TRANSMITTER BUS ADDRESS PRIMARY.
   627          000003                  TXWC.P=3        ;TRANSMITTER BUS ADDRESS PRIMARY.
   628          000004                  RXBA.S=4        ;RECEIVER BUS ADDRESS SECONDARY.
   629          000005                  RXWC.S=5        ;RECEIVER WORD COUNT SECONDARY.
   630          000006                  TXBA.S=6        ;TRANSMITTER BUS ADDRESS SECONDARY.
   631          000007                  TXWC.S=7        ;TRANSMITTER WORD COUNT SECONDARY.
   632
   633          000010                  CHARDT=10       ;CHARACTER DETECT REGISTER.
   634          000011                  SYNC.=11        ;SYNC REGISTER.
   635          000012                  MISC.=12        ;MISCELLANEOUS REGISTER.
   636          000013                  TX.MUX=13       ;TRANSMITTER MUX REGISTER.
   637          000014                  SEQ.=14         ;SEQUENCE REGISTER.
   638          000015                  RX.BCC=15       ;RECEIVER BCC REGISTER.
   639          000016                  TX.BCC=16       ;TRANSMITTER BCC REGISTER.
   640          000017                  POLY.=17        ;POLYNOMIAL REGISTER.
   641
   642
```

```
                                        ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                        000000          .=0
000000  000002          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000002  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000004  000006          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000006  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000010  000012          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000012  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000014  000016          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000016  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000020  000022          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000022  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000024  000026          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000026  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000030  000032          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000032  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000034  000036          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000036  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000040  000042          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000042  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000044  000046          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000046  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000050  000052          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000052  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000054  000056          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000056  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000060  000062          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000062  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000064  000066          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000066  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000070  000072          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000072  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000074  000076          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000076  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000100  000102          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000102  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000104  000106          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000106  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000110  000112          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000112  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000114  000116          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000116  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000120  000122          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000122  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000124  000126          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000126  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000130  000132          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000132  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000134  000136          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000136  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000140  000142          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000142  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000144  000146          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000146  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000150  000152          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000152  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
699  000154  000156                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
700  000156  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
701  000160  000162                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
702  000162  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
703  000164  000166                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
704  000166  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
705  000170  000172                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
706  000172  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
707  000174  000176                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
708  000176  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
709  000200  000202                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
710  000202  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
711  000204  000206                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
712  000206  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
713  000210  000212                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
714  000212  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
715  000214  000216                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
716  000216  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
717  000220  000222                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
718  000222  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
719  000224  000226                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
720  000226  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
721  000230  000232                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
722  000232  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
723  000234  000236                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
724  000236  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
725  000240  000242                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
726  000242  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
727  000244  000246                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
728  000246  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
729  000250  000252                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
730  000252  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
731  000254  000256                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
732  000256  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
733  000260  000262                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
734  000262  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
735  000264  000266                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
736  000266  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
737  000270  000272                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
738  000272  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
739  000274  000276                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
740  000276  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
741  000300  000302                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
742  000302  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
743  000304  000306                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
744  000306  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
745  000310  000312                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
746  000312  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
747  000314  000316                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
748  000316  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
749  000320  000322                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
750  000322  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
751  000324  000326                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
752  000326  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
753  000330  000332                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
754  000332  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
000334  000336         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000336  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000340  000342         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000342  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000344  000346         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000346  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000350  000352         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000352  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000354  000356         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000356  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000360  000362         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000362  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000364  000366         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000366  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000370  000372         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000372  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000374  000376         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000376  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000400  000402         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000402  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000404  000406         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000406  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000410  000412         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000412  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000414  000416         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000416  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000420  000422         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000422  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000424  000426         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000426  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000430  000432         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000432  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000434  000436         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000436  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000440  000442         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000442  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000444  000446         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000446  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000450  000452         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000452  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000454  000456         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000456  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000460  000462         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000462  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000464  000466         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000466  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000470  000472         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000472  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000474  000476         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000476  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000500  000502         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000502  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000504  000506         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000506  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
000510  000512         .+2          ;UNEXPECTED TRAP TO THIS LOCATION
000512  000000         HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
  811  000514  000516                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  812  000516  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  813  000520  000522                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  814  000522  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  815  000524  000526                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  816  000526  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  817  000530  000532                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  818  000532  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  819  000534  000536                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  820  000536  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  821  000540  000542                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  822  000542  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  823  000544  000546                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  824  000546  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  825  000550  000552                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  826  000552  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  827  000554  000556                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  828  000556  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  829  000560  000562                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  830  000562  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  831  000564  000566                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  832  000566  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  833  000570  000572                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  834  000572  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  835  000574  000576                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  836  000576  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  837  000600  000602                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  838  000602  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  839  000604  000606                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  840  000606  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  841  000610  000612                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  842  000612  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  843  000614  000616                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  844  000616  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  845  000620  000622                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  846  000622  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  847  000624  000626                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  848  000626  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  849  000630  000632                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  850  000632  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  851  000634  000636                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  852  000636  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  853  000640  000642                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  854  000642  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  855  000644  000646                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  856  000646  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  857  000650  000652                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  858  000652  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  859  000654  000656                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  860  000656  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  861  000660  000662                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  862  000662  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  863  000664  000666                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  864  000666  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
  865  000670  000672                .+2          ;UNEXPECTED TRAP TO THIS LOCATION
  866  000672  000000                HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
  867  000674  000676                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  868  000676  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  869  000700  000702                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  870  000702  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  871  000704  000706                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  872  000706  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  873  000710  000712                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  874  000712  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  875  000714  000716                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  876  000716  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  877  000720  000722                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  878  000722  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  879  000724  000726                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  880  000726  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  881  000730  000732                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  882  000732  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  883  000734  000736                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  884  000736  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  885  000740  000742                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  886  000742  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  887  000744  000746                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  888  000746  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  889  000750  000752                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  890  000752  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  891  000754  000756                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  892  000756  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  893  000760  000762                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  894  000762  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  895  000764  000766                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  896  000766  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  897  000770  000772                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  898  000772  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
  899  000774  000776                .+2             ;UNEXPECTED TRAP TO THIS LOCATION
  900  000776  000000                HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
                                        ;STANDARD INTERRUPT VECTORS

                000024                  .=24
000024  011056                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  010526                          .HLT                    ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  010474                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
                000046                  .=46
000046  007254                          LOGICAL                 ;ACT HOOKS
                000052                  .=52
000052  000000                          .WORD 0
                                        ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERUPT
                                        ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
                                        ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
                                        ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
                                        ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
                                        ;TO TAKE THE PC FROM THE STACK AND US IT AS THE VECTOR ADDRESS
                000056                  .=56

000056                                  VECMAP:
000056  010120                  1$:     MOV     R1,(R0)+        ;START FILLING THE VECTOR AREA
000060  012721  000004                  MOV     #4,(R1)+        ;WITH .+2; IOT (4)
000064  022021                          CMP     (R0)+,(R1)+     ;UPDATE THE POINTERS
000066  020127  001000                  CMP     R1,#1000        ;IS ALL FLOATING VECTOR AREA DONE
000072  101771                          BLOS    1$              ;BR IF NOT ALL DONE
000074  012737  000146  000020          MOV     #4$,@#20        ;SET FOR IOT TRAP BY DQ11
000102  013737  001500  001244          MOV     DQACTV,TEMP1    ;GET THE ACTIVE DQ11 S
000110  006037  001244          2$:     ROR     TEMP1           ;ARE YOU ACTIVE.. DQ11
000114  103023                          BCC     5$              ;IF CARRY CLEAR.. NO MORE DQ11S
000116  005037  177776                  CLR     PS              ;CLEAR PS
000122  005722                          TST     (R2)+           ;PUT POINTER TO STATUS TABLE
000124  012772  000340  177776          MOV     #340,@-2(R2)    ;TRY AND SET PRI/SEC DONE AND IE
000132  105200                          INCB    R0              ;DELAY.....
000134  001376                          BNE     .-2             ;.....DELAY
000136  112712  000300                  MOVB    #300,(R2)       ;NO INTERUPT ASSUME 300 FIX IN TEST C
000142  005722                  3$:     TST     (R2)+           ;UPDATE POINTERS
000144  000761                          BR      2$              ;GO DO IT AGAIN
000146  051612                  4$:     BIS     (SP),(R2)       ;ENTERD  BY IOT TRAP BY DQ11
000150  042712  000007                  BIC     #7,(R2)         ;CLEAR UNWANTED BITS
000154  022626                          CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
000156  012716  000142                  MOV     #3$,(SP)        ;SET RETURN PC ON STACK
000162  000002                          RTI                     ;GO HOME.
000164  000207                  5$:     RTS     PC              ;ALL SIZING IS DONE

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

                                        ;PROGRAM START

                000200                  .=200
000200  000137  001512                  JMP     .START          ;GO TO START OF PROGRAM
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 21
DZDQBC.P11    15-DEC-76 13:23          ROUTINES USED FOR AUTO SIZING.

                000220                  .=220
000220  012702  001400          CSRMAP: MOV     #1400,R2        ;CLEAR ALL STATUS TABLE
000224  005022                          CLR     (R2)+           ;DO CLEAR
000226  020227  001512                  CMP     #1512,R2        ;ALL TABLE DONE
000232  001374                          BNE     .-6             ;BR IF MORE TO GO
000234  005037  001504                  CLR     DQNUM           ;SET NUMBER OF DQ11S TO 0
000240  012702  001400                  MOV     #1400,R2        ;SET TABLE POINTER
000244  012701  160000                  MOV     #160000,R1      ;GET FIRST FLOATING ADDRESS
000250  012737  000614  000004          MOV     #5$,@#4         ;SET FOR TIME OUT TRAP--NO DEVICE--
000256  112761  000012  000005  1$:     MOVB    #12,5(R1)       ;TRY AND SEL MISC REGISTER
000264  005061  000006                  CLR     6(R1)           ;TRY AND CLEAR MISC REG
000270  012711  010000                  MOV     #10000,(R1)     ;TRY AND SET RX ACTIVE
000274  022761  030000  000006          CMP     #30000,6(R1)    ;LOOK FOR SYNC 1 AND SYNC 2
000302  001071                          BNE     2$              ;THIS IS NOT A DQ11 IF I BRANCH
000304  010122                          MOV     R1,(R2)+        ;NOW THIS IS A DQ11 --STORE CSR
000306  052712  100000                  BIS     #SYNBIT,(R2)    ;SET FOR TWO SYNC CHARS
000312  005011                          CLR     (R1)            ;CLEAR DQ ACTIVE BIT
000314  112761  000010  000005          MOVB    #10,5(R1)       ;SEL CHAR DET REGISTER
000322  012761  177777  000006          MOV     #-1,6(R1)       ;WRITE INTO CHAR DET REG
000330  005761  000006                  TST     6(R1)           ;WAS THE REGISTER WRITTEN?
000334  001402                          BEQ     .+6             ;APPERENTLY NO BB OPTION.
000336  052712  020000                  BIS     #BBBIT,(R2)     ;SET FOR BB OPTION
000342  112761  000017  000005          MOVB    #17,5(R1)       ;SEL POLYNO. REGISTER
000350  012761  177777  000006          MOV     #-1,6(R1)       ;WRITE POLYNO.REGISTER
000356  005761  000006                  TST     6(R1)           ;WAS REG WRITTEN??
000362  001402                          BEQ     .+6             ;BR IF NO AB OPTION
000364  052712  002000                  BIS     #ABBIT,(R2)     ;SET FOR AB OPTION
000370  012761  001400  000002          MOV     #1400,2(R1)     ;TRY TO SET ,DTR. .RS.
000376  032761  001400  000002          BIT     #1400,2(R1)     ;DID ANY OF THEM SET
000404  001402                          BEQ     .+6             ;BR IF NO BA OPTION
000406  052712  010000                  BIS     #BABIT,(R2)     ;SET FOR BA OPTION
000412  032761  030000  000002          BIT     #30000,2(R1)    ;DID .CS. .CO. SET
000420  001402                          BEQ     .+6             ;BR IF NO JUMPER
000422  052712  040000                  BIS     #JUMBIT,(R2)    ;SET FOR JUMPER
000426  052712  004000                  BIS     #ACTBIT,(R2)    ;SET FOR ACTIVE ON FIRST NON-SYNC
000432  052712  001000                  BIS     #ODDBIT,(R2)    ;SET FOR ODD VRC.......
000436  005722                          TST     (R2)+           ;POP POINTER
000440  005011                          CLR     (R1)            ;CLEAR RCSR
000442  005061  000002                  CLR     2(R1)           ;CLEAR TCSR
000446  005061  000002                  CLR     2(R1)           ;CLEAR AGAIN
000452  005061  000004                  CLR     4(R1)           ;CLEAR ERROR REG
000456  005061  000006                  CLR     6(R1)           ;CLEAR SEC REG
000462  005237  001504                  INC     DQNUM           ;UPDATE NUMBER OF DQ11S
000466  062701  000010          2$:     ADD     #10,R1          ;UPDATE CSR POINTER BY 10 (8)
000472  022701  164000                  CMP     #164000,R1      ;HAVE ALL FLOATING ADDRESSES BEEN CHECKED??
000476  001267                          BNE     1$              ;BR IF NOT ALL DONE
000500  005037  001500                  CLR     DQACTV          ;ZERO ACTIVE DQ11S
000504  005737  001504                  TST     DQNUM           ;WERE ANY DQ11S FOUND
000510  001434                          BEQ     4$              ;HEY BUDDY. NO DQ11S FOUND IN SYSTEM
000512  013701  001504                  MOV     DQNUM,R1        ;SAVE NUMBER OF DQ11S
000516  010137  001276                  MOV     R1,SAVNUM       ;SAVE NUMBER FOR ACT11
000522  000241          3$:     CLC                             ;CLEAR CARRY
000524  006137  001500                  ROL     DQACTV          ;++++ ACTIVE ADDRESS
000530  005237  001500                  INC     DQACTV          ;SET BIT 0
000534  005301                          DEC     R1              ;DEC NUMBER OF DQ11S
000536  001371                          BNE     3$              ;BR IF MORE TO GO
```

```
1013  000540  012737  000006  000004          MOV     #6,@#4          ;RESET TIME OUT VECTOR
1014  000546  013737  001500  001502          MOV     DQACTV,SAVACT   ;SAVE ACTIVE
1015  000554  012737  000340  000022          MOV     #340,@#22       ;SET IOT TRAP PRIO: TO 7
1016  000562  012702  001400                  MOV     #1400,R2        ;SET TABLE POINTER
1017  000566  012700  000300                  MOV     #300,R0         ;SET VECTOR START
1018  000572  012701  000302                  MOV     #302,R1         ;SET VECTOR+2 START
1019  000576  000137  000056                  JMP     VECMAP          ;GO FIND THE VECTORS
1020  000602  104402                  4$.     TYPE                    ;TYPE MESSAGE
1021  000604  011417                          MERR2                   ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
1022  000606  005000                          CLR     R0              ;
1023  000610  000000                          HALT                    ;HOW CAN I TEST NO DQ11S
1024  000612  000776                          BR      .-2             ;DON'T LET OPR HIT CONT. SW
1025  000614  012716  000466          5$:     MOV     #2$,(SP)        ;ENTERED BY TIME OUT TRAP
1026  000620  000002                          RTI                     ;GO HOME.
1027
1028
1029          001000                  .=1000
1030  001000  005377  040515  047111  MTITLE: .ASCIZ  <377><12>/MAINDEC-11-DZDQB-C/<377>/DQ11 STATIC LOGIC TEST-PART 2/<377>
1031  001006  042504  026503  030461
1032  001014  042055  042132  041121
1033  001022  041455  042377  030521
1034  001030  020061  052123  052101
1035  001036  041511  046040  043517
1036  001044  041511  052040  051505
1037  001052  026524  040520  052122
1038  001060  031040  000377
1039
1040          001200                  .=1200
1041                                  ;INDIRECT POINTERS
1042
1043  001200  177570          SWR:    177570                  ;SWITCH REGISTER POINTER
1044  001202  177570          LIGHTS: 177570                  ;DISPLAY REGISTER POINTER
1045  001204  177560          TKCSR:  177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
1046  001206  177562          TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
1047  001210  177564          TPCSR:  177564                  ;TELEPRINTER CONTROL REGISTER
1048  001212  177566          TPDBR:  177566                  ;TELEPRINTER DATA BUFFER
1049
1050                                  ;PROGRAM CONTROL PARAMETERS
1051
1052  001214  000000          RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
1053  001216  000000          NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
1054  001220  000000          LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
1055  001222  000003          ICOUNT: 3                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
1056  001224  000000          LPCNT:  0                       ;NUMBER OF ITEREATIONS COMPLETED
1057  001226  000000          TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
1058  001230  000000          PASCNT: 0                       ;NUMBER OF PASSES COMPLETED
1059  001232  000000          ERRCNT: 0                       ;TOTAL NUMBER OF ERRORS
1060  001234  000000          LSTERR: 0                       ;PC OF LAST ERROR CALL
1061
1062                                  ;PROGRAM VARIABLES
1063
1064  001236  000000          CHAR1:  0
1065  001240  000000          CHAR2:  0
1066  001242  000000          CHAR3:  0
1067  001244  000000          TEMP1:  0                       ;TEMPORARY STORAGE
1068  001246  000000          TEMP2:  0                       ;TEMPORARY STORAGE
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
1069  001250  000000                TEMP3:  0                       ;TEMPORARY STORAGE
1070  001252  000000                TEMP4:  0                       ;TEMPORARY STORAGE
1071  001254  000000                TEMP5:  0                       ;TEMPORARY STORAGE
1072  001256  000000                SAVR0:  0                       ;R0 STORAGE
1073  001260  000000                SAVR1:  0                       ;R1 STORAGE
1074  001262  000000                SAVR2:  0                       ;R2 STORAGE
1075  001264  000000                SAVR3:  0                       ;R3 STORAGE
1076  001266  000000                SAVR4:  0                       ;R4 STORAGE
1077  001270  000000                SAVR5:  0                       ;R5 STORAGE
1078  001272  000000                SAVSP:  0                       ;STACK POINTER STORAGE
1079  001274  000000                SAVPC:  0                       ;PROGRAM COUNTER STORAGE
1080  001276  000000                SAVNUM: 0
1081  001300  000001                CREAM:  .BLKW   1
1082  001302  000000                RUNFLG: 0
1083  001304  000000                RUN:    0
1084  001306  000000                RUNCNT: 0
```

```
1085
1086                                    ;PROGRAM CONTROL FLAGS
1087
1088  001310    000           INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
1089  001311    000           STFLG:  .BYTE   0               ;TEST START FLAG
1090  001312    000           ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
1091  001313    000           LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1092            000000          $Y=0
1093
1094                                    ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
1095                                    ;POINTERS TO SUBROUTINES CAN BE FOUND
1096                                    ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
1097
1098                                ;*****************************************************************
1099                                ;*****************************************************************
1100  001314                        .TRPTAB:
1101          104400                SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
1102  001314  007330                        .SCOPE
1103          104401                SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
1104  001316  007442                        .SCOP1
1105          104402                TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
1106  001320  007462                        .TYPE
1107          104403                INSTR=TRAP+3            ;CALL TO ASCII STRING INPUT ROUTINE
1108  001322  007570                        .INSTR
1109          104404                INSTER=TRAP+4           ;CALL TO INPUT ERROR HANDLER
1110  001324  007706                        .INSTER
1111          104405                PARAM=TRAP+5            ;CALL TO NUMERICAL DATA INPUT ROUTINE
1112  001326  007740                        .PARAM
1113          104406                SAVOS=TRAP+6            ;CALL TO REGISTER SAVE ROUTINE
1114  001330  010154                        .SAVOS
1115          104407                RESOS=TRAP+7            ;CALL TO REGISTER RESTORE ROUTINE
1116  001332  010214                        .RESOS
1117          104410                CONVRT=TRAP+10          ;CALL TO DATA OUTPUT ROUTINE
1118  001334  010246                        .CONVRT
1119          104411                CNVRT=TRAP+11           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
1120  001336  010252                        .CNVRT
1121          104412                MSTCLR=TRAP+12          ;CALL TO ISSUE MASTER CLEAR
1122  001340  012114                        .MSTCLR
1123          104413                MEMCLR=TRAP+13          ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
1124  001342  012112                        .MEMCLR
1125          104414                CKSWR=TRAP+14           ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
1126  001344  011154                        .CKSWR
1127          104415                CNTLU=TRAP+15           ;CALL TO ALLOW LOADING OF SWREG FROM TTY
1128  001346  011230                        .CNTLU
1129
1130                                ;*****************************************************************
1131                                ;*****************************************************************
1132
1133                                    ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
1134
1135  001350  000000    DQRVEC: 0               ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
1136  001352  000000    DQRLVL: 0               ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
1137  001354  000000    DQTVEC: 0               ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
1138  001356  000000    DQTLVL: 0               ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
1139  001360  000000    DQRCSR: 0               ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
1140  001362  000000    DQRCSH: 0               ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER
```

```
DZDQ8   MACY11 27(1006)  22-DEC-76  11:05  PAGE 25
DZDQ8C.P11    16-DEC-76 13:23          PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

  1141  001364  000000                DQTCSR: 0               ;POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
  1142  001366  000000                DQERR:  0               ;POINTER TO DQ11 ERROR REGISTER
  1143  001370  000000                DQREG:  0               ;POINTER TO HIGH BYTE OF ERROR REGISTER
  1144  001372  000000                DQSEC:  0               ;POINTER TO DQ11 SECONDARY REGISTER
  1145  001374  000000                DQSECH: 0               ;POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
  1146
  1147
  1148
  1149                                        ;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
  1150
  1151          001400                .=1400
  1152  001400  000001                DQCR00: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 00
  1153  001402  000001                DQST00: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
  1154  001404  000001                DQCR01: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 01
  1155  001406  000001                DQST01: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
  1156  001410  000001                DQCR02: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 02
  1157  001412  000001                DQST02: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
  1158  001414  000001                DQCR03: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 03
  1159  001416  000001                DQST03: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
  1160  001420  000001                DQCR04: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 04
  1161  001422  000001                DQST04: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
  1162  001424  000001                DQCR05: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 05
  1163  001426  000001                DQST05: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
  1164  001430  000001                DQCR06: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 06
  1165  001432  000001                DQST06: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
  1166  001434  000001                DQCR07: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 07
  1167  001436  000001                DQST07: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
  1168  001440  000001                DQCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 10
  1169  001442  000001                DQST10: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
  1170  001444  000001                DQCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 11
  1171  001446  000001                DQST11: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
  1172  001450  000001                DQCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 12
  1173  001452  000001                DQST12: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
  1174  001454  000001                DQCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 13
  1175  001456  000001                DQST13: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
  1176  001460  000001                DQCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 14
  1177  001462  000001                DQST14: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
  1178  001464  000001                DQCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 15
  1179  001466  000001                DQST15: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
  1180  001470  000001                DQCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 16
  1181  001472  000001                DQST16: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
  1182  001474  000001                DQCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR DEVICE NO: 17
  1183  001476  000001                DQST17: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
  1184  001500  000001                DQACTV: .BLKW   1       ;HOLD ACTIVE BITS FOR TESTING
  1185  001502  000001                SAVACT: .BLKW   1       ;SAVE NUMBER OF ACTIVE DQ11S
  1186  001504  000001                DQNUM:  .BLKW   1       ;OCTAL NUMBER OF TOTAL NUMBER OD DQ11S
  1187  001506  000001                DQCSR:  .BLKW   1       ;CSR OF DQ11 UNDER TEST
  1188  001510  000001                DQSTAT: .BLKW   1       ;VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
  1189
  1190                                        ;PROGRAM INITIALIZATION
  1191                                        ;LOCK OUT INTERRUPTS
  1192                                        ;SET UP PROCESSOR STACK
  1193                                        ;SET UP POWER FAIL VECTOR
  1194                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
  1195                                        ;TYPE TITLE MESSAGE
  1196
```

```
1197  001512  012737  000340  177776  .START: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
1198  001520  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
1199  001524  012737  011056  000024          MOV     #,PFAIL,@#24            ;SET UP POWER FAIL VECTOR
1200  001532  013737  001504  001276          MOV     DQNUM,SAVNUM
1201  001540  105037  001311                  CLRB    STFLG                   ;CLEAR START FLAG
1202  001544  005037  001230                  CLR     PASCNT                  ;CLEAR PASS COUNT
1203  001550  105037  001312                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
1204  001554  005037  001302                  CLR     RUNFLG
1205  001560  012737  001400  001300          MOV     #1400,CREAM
1206  001566  005037  001232                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
1207  001572  005037  001234                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
1208  001576  012737  000001  001226          MOV     #1,TSTNO                ;SET UP FOR TEST 1
1209  001604  012737  001512  001214          MOV     #.START,RETURN          ;SET UP FOR POWER FAIL BEFORE
1210                                                                          ;TESTING STARTS
1211  001612  105737  001310                  TSTB    INIFLG                  ;HAS INITIALIZATION BEEN PERFORMED
1212  001616  001075                          BNE     12$
1213  001620  104402  001000                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
1214  001624  105137  001310                  COMB    INIFLG                  ;IF NOT SET FLAG AND DO
1215
1216  001630  012737  177570  001200          MOV     #DSWR,SWR               ;MOV HARDWARE SWR TO SWR
1217  001636  012737  177570  001202          MOV     #DLIGHTS,LIGHTS         ;MOV DISPLAY LIGHTS TO LIGHTS
1218  001644  013746  000006                  MOV     @#6,-(SP)               ;SAVE VECTORS
1219  001650  013746  000004                  MOV     @#4,-(SP)
1220  001654  012737  001674  000004          MOV     #64$,@#4                        ;SET UP FOR TIMEOUT
1221  001662  022777  177777  177310          CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
1222  001670  001402                          BEQ     65$
1223  001672  000407                          BR      66$
1224  001674  022626                  64$:    CMP     (SP)+,(SP)+             ;ADJUST STACK
1225  001676  012737  000176  001200  65$:    MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
1226  001704  012737  000174  001202          MOV     #DISPREG,LIGHTS         ;POINT TO SOFT DISPLAY REG
1227  001712  012637  000004          66$:    MOV     (SP)+,@#4               ;RESTORE VECTORS
1228  001716  012637  000006                  MOV     (SP)+,@#6
1229  001722  005737  000042                  TST     @#42                    ;UNDER MONITOR
1230  001726  001005                          BNE     67$
1231  001730  012737  000176  001200          CMP     #SWREG,SWR              ;IS SWREG USED
1232  001736  001001                                  BNE     67$
1233  001740  104415                          CNTLU
1234  001742  105777  177232          67$:    TSTB    @SWR
1235  001746  100402                          BMI     .+6
1236  001750  004737  000220                  JSR     PC,CSRMAP
1237  001754  104402  011704                  TYPE    ,XHEAD
1238  001760  012737  001400  001244          MOV     #1400,TEMP1
1239  001766  017737  177252  001246          MOV     @TEMP1,TEMP2
1240  001774  001406                          BEQ     .+16
1241  001776  104410                          CONVRT
1242  002000  011732                          XSTATQ
1243  002002  062737  000002  001244          ADD     #2,TEMP1
1244  002010  000766                          BR      .-22
1245  002012  032777  000001  177160  12$:    BIT     #SW00,@SWR
1246  002020  001424                          BEQ     1$
1247  002022  104402                          TYPE
1248  002024  011625                          MNEW
1249  002026  005000                          CLR     R0
1250  002030  000000                          HALT
1251  002032  104414                          CKSWR
1252  002034  027737  177140  001502          CMP     @SWR,SAVACT
```

```
 1253  002042  101404                          BLOS    11$
 1254  002044  104402                          TYPE
 1255  002046  011466                          MERR3
 1256  002050  000000                          HALT
 1257  002052  000776                          BR      .-2
 1258  002054  017737  177120  001500  11$:    MOV     @SWR,DQACTV
 1259  002062  013700  001500                  MOV     DQACTV,R0
 1260  002066  000000                          HALT
 1261  002070  104414                          CKSWR
 1262  002072  012700  000300          1$:     MOV     #300,R0
 1263  002076  012701  000302                  MOV     #302,R1
 1264  002102  010120                  2$:     MOV     R1,(R0)+
 1265  002104  005021                          CLR     (R1)+
 1266  002106  022021                          CMP     (R0)+,(R1)+
 1267  002110  022700  001000                  CMP     #1000,R0
 1268  002114  001372                          BNE     2$
 1269
 1270                                          ;TEST START AND RESTART
 1271
 1272  002116  012737  000340  177776  .BEGIN: MOV     #340,PS                 ;LOCK OUT INTERRUPTS
 1273  002124  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
 1274  002130  005737  000042                  TST     @#42                    ;IS PROGRAM UNDER MONITOR CONTROL
 1275  002134  001040                          BNE     3$
 1276  002136  104414                          CKSWR                           ;CHECK FOR (↑G)
 1277  002140  032777  000004  177032          BIT     #BIT2,@SWR              ;CHECK FOR LOCK ON TEST
 1278  002146  001411                          BEQ     1$
 1279  002150  104402  011524                  TYPE    ,MLOCK
 1280  002154  012737  000240  007340          MOV     #NOP,TTST
 1281  002162  012737  000240  007342          MOV     #NOP,TTST+2             ;SET UP TO LOCK
 1282  002170  000406                          BR      2$
 1283  002172  013737  007436  007340  1$:     MOV     BRW,TTST
 1284  002200  013737  007440  007342          MOV     BRX,TTST+2              ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
 1285  002206  032777  000002  176764  2$:     BIT     #SW01,@SWR              ;IF SW01=1, GET STARTING PC
 1286  002214  001410                          BEQ     3$
 1287  002216  104403                          INSTR
 1288  002220  011512                          MTSTPC
 1289  002222  104405                          PARAM
 1290  002224  002254                          TST1
 1291  002226  007046                          TLAST
 1292  002230  001214                          #RETURN
 1293  002232     001                  .BYTE   1
 1294  002233     001                  .BYTE   1
 1295  002234  000403                          BR      4$
 1296  002236  012737  002254  001214  3$:     MOV     #TST1,RETURN            ;START AT TEST 1
 1297  002244  104402  011414          4$:     TYPE    ,MR                     ;TYPE R
 1298  002250  000177  176740                  JMP     @RETURN                 ;START TESTING
 1299                                  ;  TEST 1
 1300                                  ;*****************************************************************************
 1301  002254  012737  000001  001226  TST1:   MOV     #1,TSTNO
 1302  002262  012737  002644  001214          MOV     #TST2,RETURN
 1303  002270  012737  002644  001216          MOV     #TST2,NEXT
 1304  002276  105737  001302                  TSTB    RUNFLG          ;IS THIS MY FIRST TIME HERE?
 1305  002302  001010                          BNE     1$              ;BR IF FLAG IS SET
 1306  002304  012737  000001  001304          MOV     #BIT0,RUN       ;SET RUN POINTER.
 1307  002312  012737  000020  001306          MOV     #16.,RUNCNT     ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
 1308  002320  105137  001302                  COMB    RUNFLG          ;SET RUN FLAG
```

```
 1309  002324  033737  001304  001500  1$:     BIT     RUN,DQACTV      ;FIND AN ACTIVE DQ11 TO TEST.
 1310  002332  001032                          BNE     3$              ;BR IF I FOUND ONE TO TEST.
 1311  002334  005737  001500                  TST     DQACTV          ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
 1312  002340  001423                          BEQ     2$              ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
 1313  002342  000257                          CCC                     ;CLEAR ALL THE CONDITION CODES OF CPU
 1314  002344  006137  001304                  ROL     RUN             ;UPDATE RUN POINTER
 1315  002350  062737  000004  001300          ADD     #4,CREAM        ;UPDATE ADDRESS POINTER.
 1316  002356  005337  001306                  DEC     RUNCNT          ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
 1317  002362  001360                          BNE     1$              ;BR AND KEEP LOOKING.
 1318  002364  012737  000020  001306          MOV     #16.,RUNCNT     ;START RESTORING MY POINTERS.
 1319  002372  012737  001400  001300          MOV     #1400,CREAM     ;RESTORE  ADDRESS POINTER
 1320  002400  012737  000001  001304          MOV     #1,RUN          ;RESTORE RUN POINTER.
 1321  002406  000746                          BR      1$              ;KEEP ON TESTING.
 1322  002410  104402                  2$:     TYPE                    ;ALLERT OPERATOR OF FATAL ERROR
 1323  002412  011417                          MERR2                   ;NO DQ11 ACTIVE. WHY AM I HERE???
 1324  002414  000000                          HALT                    ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
 1325  002416  000776                          BR      .-2             ;STICK HERE ON CONT.
 1326  002420  000257                  3$:     CCC                     ;CLEAR CPU COND. CODES
 1327  002422  006137  001304                  ROL     RUN             ;UPDATE RUN. ACTIVE DQ11 FOUND.
 1328  002426  017737  176646  001506          MOV     @CREAM,DQCSR    ;PLACE ADDRESSS OF DQ11 AT DQCSR
 1329  002434  062737  000002  001300          ADD     #2,CREAM        ;UPDATE ADDRESS POINTER
 1330  002442  017737  176632  001510          MOV     @CREAM,DQSTAT   ;PLACE STATUS OF DQ11 AT DQSTAT
 1331  002450  062737  000002  001300          ADD     #2,CREAM        ;UPDATE ADDRESS POINTER
 1332  002456  013737  001506  001360          MOV     DQCSR,DQRCSR
 1333  002464  013737  001510  001350          MOV     DQSTAT,DQRVEC
 1334  002472  042737  177007  001350          BIC     #177007,DQRVEC
 1335  002500  013737  001350  001352          MOV     DQRVEC,DQRLVL   ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
 1336  002506  062737  000002  001352          ADD     #2,DQRLVL
 1337  002514  013737  001352  001354          MOV     DQRLVL,DQTVEC   ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
 1338  002522  062737  000002  001354          ADD     #2,DQTVEC
 1339  002530  013737  001354  001356          MOV     DQTVEC,DQTLVL   ;GENERATE ADDRESS  OF TRANSMITTER INTERRUPT SERVICE PS
 1340  002536  062737  000002  001356          ADD     #2,DQTLVL
 1341  002544  013737  001360  001362          MOV     DQRCSR,DQRCSH
 1342  002552  005237  001362                  INC     DQRCSH          ;GENERATE ADDRESS OF HIGH BYTE
 1343  002556  013737  001360  001364          MOV     DQRCSR,DQTCSR   ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
 1344  002564  062737  000002  001364          ADD     #2,DQTCSR
 1345  002572  013737  001364  001366          MOV     DQTCSR,DQERR    ;GENERATE ADDRESS OF ERROR REGISTER
 1346  002600  062737  000002  001366          ADD     #2,DQERR
 1347  002606  013737  001366  001370          MOV     DQERR,DQREG     ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
 1348  002614  005237  001370                  INC     DQREG
 1349  002620  013737  001370  001372          MOV     DQREG,DQSEC     ;GENERATE ADDRESS OF SECONDARY REGISTER
 1350  002626  005237  001372                  INC     DQSEC
 1351  002632  013737  001372  001374          MOV     DQSEC,DQSECH    ;GENERATE ADDRESS OF HIGH BYTE
 1352  002640  005237  001374                  INC     DQSECH
```

```
                                        ;MEMORY EXTENSION WRITE ENABLE TEST
                                        ;SET MEMORY EXTENSION WRITE ENABLE
                                        ;ADDRESS BUS ADDRESS MEMORY
                                        ;EXPECTED RESULTS
                                        ;       MEMORY EXTENSION WRITE ENABLE IS CLEARED

002644                          OPT2X:
                                ;  TEST 2
                                ;*************************************************************************
002644  012737  000002  001226  TST2:   MOV     #2,TSTNO
002652  012737  002706  001216          MOV     #TST3,NEXT
002660  112777  000020  176502          MOVB    #BIT4,@DQREG            ;SET WRITE ENABLE,
                                                                        ;SELECT RECEVIER BUS ADDRESS
                                                                        ;MEMORY
002666  105077  176500                  CLRB    @DQSEC                  ;CLEAR MEMORY
002672  132777  000020  176470          BITB    #BIT4,@DQREG            ;WAS WRITE ENABLE CLEARED
002700  001401                          BEQ     1$                      ;BR IF WRITE ENABLE CLEARED
002702  104012                          HLT     12                      ;WRITE ENABLE NOT CLEARED
002704  104400                  1$:     SCOPE
```

```
1372
1373                                            ;MEMORY EXTENTION BITS READ/WRITE TEST
1374                                            ;READ MEMORY EXTENSION BITS
1375                                            ;WITH WRITE ENABLE BITS CLEARED, ATTEMPT
1376                                            ;TO CHANGE MEMORY EXTENSION BITS
1377                                            ;VERITY THAT NO CHANGE OCCURS
1378
1379                                    ;  TEST 3
1380                                    ;*************************************************************************
1381  002706  012737  000003  001226  TST3:    MOV     #3,TSTNO
1382  002714  012737  002774  001216          MOV     #TST4,NEXT
1383  002722  013703  001370                  MOV     @DQREG,R3       ;SET FOR ERROR PRINTOUT
1384  002726  142777  000037  176434          BICB    #37,@DQREG      ;CLEAR MEM EXT AND WRITE ENABLE
1385  002734  117705  176430                  MOVB    @DQREG,R5       ;SAVE REGISTER
1386  002740  010500                          MOV     R5,R0           ;STORE REG
1387  002742  105100                          COMB    R0              ;CHANGE CONTENTS
1388  002744  142700  000237                  BICB    #237,R0         ;CLEAR UNWANTED BITS
1389  002750  110077  176414                  MOVB    R0,@DQREG       ;WRITE REGISTER
1390  002754  117704  176410                  MOVB    @DQREG,R4       ;READ REGISTER
1391  002760  142704  000037                  BICB    #37,R4          ;CLEAR UNWANTED BITS
1392  002764  020504                          CMP     R5,R4           ;DID IT CHANGE
1393  002766  001401                          BEQ     1$              ;BR IF NO CHANGE OCCURED
1394  002770  104000                          HLT     0               ;REPORT ERROR
1395  002772  104400                  1$:     SCOPE                   ;SCOPE THIS TEST
1396
1397                                            ;MEMORY EXTENSION READ/WRITE TEST
1398                                            ;READ MEMORY EXTENSION BITS
1399                                            ;WITH WRITE ENABLE SET, ATTEMPT TO CHANGE
1400                                            ;MEMORY EXTENSION BITS
1401                                            ;VERIFY THAT MEMORY EXTENSION BITS WERE CHANGED
1402
1403                                    ;  TEST 4
1404                                    ;*************************************************************************
1405  002774  012737  000004  001226  TST4:    MOV     #4,TSTNO
1406  003002  012737  003222  001216          MOV     #TST5,NEXT
1407  003010  013703  001370                  MOV     @DQREG,R3       ;SET FOR ERROR REPORT
1408  003014  142777  000037  176346          BICB    #37,@DQREG      ;CLEAR UNWANTED BITS.
1409  003022  117700  176342                  MOVB    @DQREG,R0       ;READ DQREG FOR DATA
1410  003026  105100                          COMB    R0              ;DO A COMPLETE CHANGE OF DATA
1411  003030  142700  000237                  BICB    #237,R0         ;CLEAR WRITE ENABLE AND GARBAGE
1412  003034  010005                          MOV     R0,R5           ;STORE DATA.
1413  003036  152705  000020                  BISB    #BIT4,R5        ;SET WRITE ENABLE
1414  003042  110577  176322                  MOVB    R5,@DQREG       ;WRITE DQREG WITH NEW DATA
1415  003046  005077  176320                  CLR     @DQSEC          ;REFERANCE SEL 6 (DQSEC)
1416  003052  117704  176312                  MOVB    @DQREG,R4       ;READ DQREG
1417  003056  042704  000037                  BIC     #37,R4          ;CLEAR UNWANTED BITS.
1418  003062  042705  000020                  BIC     #20,R5          ;CLEAR WRITE ENABLE
1419  003066  020504                          CMP     R5,R4           ;ARE THEY EQUAL?
1420  003070  001401                          BEQ     1$              ;BR IF GOOD
1421  003072  104000                          HLT     0               ;MEMORY EXTENSION READ/WRITE ERROR
1422  003074  104400                  1$:     SCOPE                   ;SCOPE THE TEST.
```

```
1423
1424                                        ;BUS ADDRESS EXTENSION MEMORY TEST
1425                                        ;LOAD EACH BUS ADDRESS MEMORY LOCATION WITH A DIFFERENT
1426                                        ;NUMBER (0-3)
1427                                        ;VERIFY THAT EACH LOCATION WAS ADDRESSED PROPERLY
1428
1429  003076  005003                        CLR     R3                      ;FIRST MEMORY ADDRESS =0
1430  003100  052701  000020                BIS     #BIT4,R1                ;SET WRITE ENABLE
1431  003104  112777  000020  176256        MOVB    #BIT4,@DQREG            ;SET MEMORY EXTENSION WRITE ENABLE
1432  003112  150377  176252                BISB    R3,@DQREG               ;SELECT LOCATION TO BE TESTED
1433  003116  110177  176246                MOVB    R1,@DQREG               ;LOAD DATA INTO MEMORY EXTENSION BITS
1434  003122  005077  176244                CLR     @DQSEC                  ;REFERENCE SEL 6 (DQSEC)
1435  003126  062703  000002                ADD     #2,R3                   ;NEXT LOCATION TO BE TESTED
1436  003132  062705  000040                ADD     #40,R5                  ;DATA TO BE LOADED
1437  003136  005300                        DEC     R0                      ;CONTINUE IF ALL LOCATIONS NOT LOADED
1438  003140  001355                        BNE     1$                      ;BR IF MORE TO GO
1439  003142  005005                        CLR     R5                      ;INITIAL DATA = 0
1440  003144  012700  000004                MOV     #4,R0                   ;4 LOCATIONS TO BE TESTED
1441  003150  005003                        CLR     R3                      ;FIRST LOCATION
1442  003152  142777  000037  176210  2$:   BICB    #37,@DQREG              ;REINITIALIZE  ADDRESS
1443  003160  150377  176204                BISB    R3,@DQREG               ;RESELECT SECONDARY ADDRESS.
1444  003164  117704  176200                MOVB    @DQREG,R4               ;READ MEMORY EXTENSION BITS
1445  003170  142704  000037                BICB    #37,R4                  ;CLEAR UNWANTED BITS
1446  003174  020504                        CMP     R5,R4                   ;ARE EXPECTED AND RECEIVED DATA THE SAME
1447  003176  001401                        BEQ     3$
1448  003200  104001                        HLT     1                       ;BUS ADDRESS EXTENSION MEMORY ERROR
1449  003202  104401                3$:     SCOP1                           ;CHECK FOR LOOP ON CUREENT ADDRESS
1450  003204  062703  000002                ADD     #2,R3                   ;UPDATE SELECT ADDRESS
1451  003210  062705  000040                ADD     #40,R5                  ;UPDATE EXPECTED ATA
1452  003214  005300                        DEC     R0                      ;CONTINUE IF NOT DONE
1453  003216  001355                        BNE     2$                      ;BR IF MORE TO CHECK.
1454  003220  104400                4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1455
1456                                        ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1457                                        ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1458                                        ;TO 0
1459                                        ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1460                                        ;COUNT MEMORY WAS SET TO 0
1461
1462                                ;  TEST 5
1463                                ;*************************************************************************
1464  003222  012737  000005  001226  TST5:  MOV     #5,TSTNO
1465  003230  012737  003336  001216        MOV     #TST6,NEXT
1466  003236  012737  003306  001220        MOV     #2$,LOCK
1467  003244  012705  000000                MOV     #0,R5                   ;DATA TO BE LOADED INTO MEMORY
1468  003250  013703  001372                MOV     DQSEC,R3                ;ADDRESS OF SECONDARY REGISTER
1469  003254  005002                        CLR     R2                      ;FIRST MEMORY LOCATION TO BE TESTED
1470  003256  012700  000010                MOV     #10,R0                  ;# OF MEMORY LOCATIONS TO BE TESTED
1471  003262  110277  176102          1$:   MOVB    R2,@DQREG               ;SELECT MEMORY LOCATION TO BE TESTED
1472  003266  010577  176100                MOV     R5,@DQSEC               ;LOAD DATA INTO MEMORY
1473  003272  005202                        INC     R2                      ;GO TO NEXT LOCATION
1474  003274  005300                        DEC     R0                      ;MORE TO GO??
1475  003276  001371                        BNE     1$                      ;BR IF NOT DONE
1476  003300  005002                        CLR     R2                      ;SET UP TO TEST
1477  003302  012700  000010                MOV     #10,R0                  ;10 (OCTAL) LOCATIONS
1478  003306  110277  176056          2$:   MOVB    R2,@DQREG               ;SELECT LOCATION TO BE TESTED
```

BUS ADDRESS AND CHARACTER COUNT MEMORY TESTS

```
1479  003312  017704  176054              MOV     @DQSEC,R4               ;READ DATA
1480  003316  020504                      CMP     R5,R4                   ;COMPARE TO EXPECTED DATA
1481  003320  001401                      BEQ     3$                      ;BR IF GOOD
1482  003322  104002                      HLT     2                       ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1483  003324  104401              3$:     SCOP1                           ;CHECK FOR FREEZE ON CURRENT LOCATION
1484  003326  005202                      INC     R2                      ;GO TO NEXT LOCATION
1485  003330  005300                      DEC     R0                      ;DONE YET?
1486  003332  001365                      BNE     2$                      ;BR IF FURTHER CHECKING NEEDED
1487  003334  104400              4$:     SCOPE
1488
1489                                      ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1490                                      ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1491                                      ;TO 177777
1492                                      ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1493                                      ;COUNT MEMORY WAS SET TO 177777
1494
1495                                  ;  TEST 6
1496                                  ;*************************************************************************
1497  003336  012737  000006  001226  TST6:   MOV     #6,TSTNO
1498  003344  012737  003452  001216          MOV     #TST7,NEXT
1499  003352  012737  003422  001220          MOV     #2$,LOCK
1500  003360  012705  177777                  MOV     #177777,R5              ;DATA TO BE LOADED INTO MEMORY
1501  003364  013703  001372                  MOV     DQSEC,R3                ;ADDRESS OF SECONDARY REGISTER
1502  003370  005002                          CLR     R2                      ;FIRST MEMORY LOCATION TO BE TESTED
1503  003372  012700  000010                  MOV     #10,R0                  ;# OF MEMORY LOCATIONS TO BE TESTED
1504  003376  110277  175766          1$:     MOVB    R2,@DQREG               ;SELECT MEMORY LOCATION TO BE TESTED
1505  003402  010577  175764                  MOV     R5,@DQSEC               ;LOAD DATA INTO MEMORY
1506  003406  005202                          INC     R2                      ;GO TO NEXT LOCATION
1507  003410  005300                          DEC     R0                      ;MORE TO GO??
1508  003412  001371                          BNE     1$                      ;BR IF NOT DONE
1509  003414  005002                          CLR     R2                      ;SET UP TO TEST
1510  003416  012700  000010                  MOV     #10,R0                  ;10 (OCTAL) LOCATIONS
1511  003422  110277  175742          2$:     MOVB    R2,@DQREG               ;SELECT LOCATION TO BE TESTED
1512  003426  017704  175740                  MOV     @DQSEC,R4               ;READ DATA
1513  003432  020504                          CMP     R5,R4                   ;COMPARE TO EXPECTED DATA
1514  003434  001401                          BEQ     3$                      ;BR IF GOOD
1515  003436  104002                          HLT     2                       ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1516  003440  104401                  3$:     SCOP1                           ;CHECK FOR FREEZE ON CURRENT LOCATION
1517  003442  005202                          INC     R2                      ;GO TO NEXT LOCATION
1518  003444  005300                          DEC     R0                      ;DONE YET?
1519  003446  001365                          BNE     2$                      ;BR IF FURTHER CHECKING NEEDED
1520  003450  104400                  4$:     SCOPE
1521
1522                                      ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1523                                      ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1524                                      ;TO 125252
1525                                      ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1526                                      ;COUNT MEMORY WAS SET TO 125252
1527
1528                                  ;  TEST 7
1529                                  ;*************************************************************************
1530  003452  012737  000007  001226  TST7:   MOV     #7,TSTNO
1531  003460  012737  003566  001216          MOV     #TST10,NEXT
1532  003466  012737  003536  001220          MOV     #2$,LOCK
1533  003474  012705  125252                  MOV     #125252,R5              ;DATA TO BE LOADED INTO MEMORY
1534  003500  013703  001372                  MOV     DQSEC,R3                ;ADDRESS OF SECONDARY REGISTER
```

BUS ADDRESS AND CHARACTER COUNT MEMORY TESTS

```
1535  003504  005002                          CLR     R2              ;FIRST MEMORY LOCATION TO BE TESTED
1536  003506  012700  000010                  MOV     #10,R0          ;# OF MEMORY LOCATIONS TO BE TESTED
1537  003512  110277  175652          1$:     MOVB    R2,@DQREG       ;SELECT MEMORY LOCATION TO BE TESTED
1538  003516  010577  175650                  MOV     R5,@DQSEC       ;LOAD DATA INTO MEMORY
1539  003522  005202                          INC     R2              ;GO TO NEXT LOCATION
1540  003524  005300                          DEC     R0              ;MORE TO GO??
1541  003526  001371                          BNE     1$              ;BR IF NOT DONE
1542  003530  005002                          CLR     R2              ;SET UP TO TEST
1543  003532  012700  000010                  MOV     #10,R0          ;10 (OCTAL) LOCATIONS
1544  003536  110277  175626          2$:     MOVB    R2,@DQREG       ;SELECT LOCATION TO BE TESTED
1545  003542  017704  175624                  MOV     @DQSEC,R4       ;READ DATA
1546  003546  020504                          CMP     R5,R4           ;COMPARE TO EXPECTED DATA
1547  003550  001401                          BEQ     3$              ;BR IF GOOD
1548  003552  104002                          HLT     2               ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1549  003554  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON CURRENT LOCATION
1550  003556  005202                          INC     R2              ;GO TO NEXT LOCATION
1551  003560  005300                          DEC     R0              ;DONE YET?
1552  003562  001365                          BNE     2$              ;BR IF FURTHER CHECKING NEEDED
1553  003564  104400                  4$:     SCOPE
1554
1555                                          ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1556                                          ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1557                                          ;TO 52525
1558                                          ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1559                                          ;COUNT MEMORY WAS SET TO 52525
1560
1561                                  ;  TEST 10
1562                                  ;*****************************************************************
1563  003566  012737  000010  001226  TST10:  MOV     #10,TSTNO
1564  003574  012737  003702  001216          MOV     #TST11,NEXT
1565  003602  012737  003652  001220          MOV     #2$,LOCK
1566  003610  012705  052525                  MOV     #52525,R5       ;DATA TO BE LOADED INTO MEMORY
1567  003614  013703  001372                  MOV     DQSEC,R3        ;ADDRESS OF SECONDARY REGISTER
1568  003620  005002                          CLR     R2              ;FIRST MEMORY LOCATION TO BE TESTED
1569  003622  012700  000010                  MOV     #10,R0          ;# OF MEMORY LOCATIONS TO BE TESTED
1570  003626  110277  175536          1$:     MOVB    R2,@DQREG       ;SELECT MEMORY LOCATION TO BE TESTED
1571  003632  010577  175534                  MOV     R5,@DQSEC       ;LOAD DATA INTO MEMORY
1572  003636  005202                          INC     R2              ;GO TO NEXT LOCATION
1573  003640  005300                          DEC     R0              ;MORE TO GO??
1574  003642  001371                          BNE     1$              ;BR IF NOT DONE
1575  003644  005002                          CLR     R2              ;SET UP TO TEST
1576  003646  012700  000010                  MOV     #10,R0          ;10 (OCTAL) LOCATIONS
1577  003652  110277  175512          2$:     MOVB    R2,@DQREG       ;SELECT LOCATION TO BE TESTED
1578  003656  017704  175510                  MOV     @DQSEC,R4       ;READ DATA
1579  003662  020504                          CMP     R5,R4           ;COMPARE TO EXPECTED DATA
1580  003664  001401                          BEQ     3$              ;BR IF GOOD
1581  003666  104002                          HLT     2               ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1582  003670  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON CURRENT LOCATION
1583  003672  005202                          INC     R2              ;GO TO NEXT LOCATION
1584  003674  005300                          DEC     R0              ;DONE YET?
1585  003676  001365                          BNE     2$              ;BR IF FURTHER CHECKING NEEDED
1586  003700  104400                  4$:     SCOPE
1587
1588                                          ;BUS ADDRESS MEMORY EXTENSION DATA TEST
1589                                          ;LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 40
1590                                          ;VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA
```

```
1591
1592                                        ;  TEST 11
1593                                        ;*************************************************************************
1594  003702  012737  000011  001226  TST11:  MOV     #11,TSTNO
1595  003710  012737  004052  001216          MOV     #TST12,NEXT
1596  003716  012737  004006  001220          MOV     #2$,LOCK
1597  003724  012705  000040                  MOV     #40,R5
1598  003730  012700  000004                  MOV     #4,R0          ;4 LOCATIONS WILL BE ADDRESSED
1599  003734  005003                          CLR     R3             ;FIRST MEMORY ADDRESS =0
1600                                                                 ;(RECEIVER PRIMARY BUS ADDRESS)
1601  003736  010501                  1$:     MOV     R5,R1          ;LOAD "DATA"
1602  003740  050301                          BIS     R3,R1          ;LOAD SECONDARY ADDRESS
1603  003742  052701  000020                  BIS     #BIT4,R1       ;SET WRITE ENABLE.
1604  003746  112777  000020  175414          MOVB    #BIT4,@DQREG   ;SET MEMORY EXTENSION WRITE ENABLE
1605  003754  150377  175410                  BISB    R3,@DQREG      ;SELECT LOCATION TO BE TESTED
1606  003760  110177  175404                  MOVB    R1,@DQREG      ;LOAD DATA INTO MEMORY EXTENSION BITS
1607  003764  005077  175402                  CLR     @DQSEC         ;REFERENCE SEL 6 (DQSEC)
1608  003770  062703  000002                  ADD     #2,R3          ;NEXT LOCATION TO BE TESTED
1609  003774  005300                          DEC     R0             ;CONTINUE IF ALL LOCATIONS NOT LOADED
1610  003776  001357                          BNE     1$             ;
1611  004000  012700  000004                  MOV     #4,R0          ;4 LOCATIONS TO BE TESTED
1612  004004  005003                          CLR     R3             ;FIRST LOCATION
1613  004006  142777  000037  175354  2$:     BICB    #37,@DQREG     ;REINITIALIZE  ADDRESS
1614  004014  150377  175350                  BISB    R3,@DQREG      ;RESELECT SECONDARY ADDRESS
1615  004020  117704  175344                  MOVB    @DQREG,R4      ;READ DQREG
1616  004024  1 2704  000037                  BICB    #37,R4         ;CLEAR UNWANTED BITS
1617  004030  020504                          CMP     R5,R4          ;IS DATA CORRECT?
1618  004032  001401                          BEQ     3$             ;BR IF OK.
1619  004034  104003                          HLT     3              ;BUS ADDRESS EXTENSION MEMORY ERROR
1620  004036  104401                  3$:     SCOP1                  ;FREEZE ON CURRENT DATA? (SW09=1)
1621  004040  062703  000002                  ADD     #2,R3          ;UPDATE POINTER TO NEXT ADDRESS
1622  004044  005300                          DEC     R0             ;ALL ADDRESS DONE?
1623  004046  001357                          BNE     2$             ;BR IF NOT DONE.
1624  004050  104400                  4$:     SCOPE                  ;SCOPE THE TEST.
1625
1626                                        ;BUS ADDRESS MEMORY EXTENSION DATA TEST
1627                                        ;LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 100
1628                                        ;VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA
1629
1630                                        ;  TEST 12
1631                                        ;*************************************************************************
1632  004052  012737  000012  001226  TST12:  MOV     #12,TSTNO
1633  004060  012737  004222  001216          MOV     #TST13,NEXT
1634  004066  012737  004156  001220          MOV     #2$,LOCK
1635  004074  012705  000100                  MOV     #100,R5
1636  004100  012700  000004                  MOV     #4,R0          ;4 LOCATIONS WILL BE ADDRESSED
1637  004104  005003                          CLR     R3             ;FIRST MEMORY ADDRESS =0
1638                                                                 ;(RECEIVER PRIMARY BUS ADDRESS)
1639  004106  010501                  1$:     MOV     R5,R1          ;LOAD "DATA"
1640  004110  050301                          BIS     R3,R1          ;LOAD SECONDARY ADDRESS
1641  004112  052701  000020                  BIS     #BIT4,R1       ;SET WRITE ENABLE.
1642  004116  112777  000020  175244          MOVB    #BIT4,@DQREG   ;SET MEMORY EXTENSION WRITE ENABLE
1643  004124  150377  175240                  BISB    R3,@DQREG      ;SELECT LOCATION TO BE TESTED
1644  004130  110177  175234                  MOVB    R1,@DQREG      ;LOAD DATA INTO MEMORY EXTENSION BITS
1645  004134  005077  175232                  CLR     @DQSEC         ;REFERENCE SEL 6 (DQSEC)
1646  004140  062703  000002                  ADD     #2,R3          ;NEXT LOCATION TO BE TESTED
```

```
1647  004144  005300                          DEC     R0              ;CONTINUE IF ALL LOCATIONS NOT LOADED
1648  004146  001357                          BNE     1$              ;
1649  004150  012700  000004                  MOV     #4,R0           ;4 LOCATIONS TO BE TESTED
1650  004154  005003                          CLR     R3              ;FIRST LOCATION
1651  004156  142777  000037  175204  2$:     BICB    #37,@DQREG      ;REINITIALIZE  ADDRESS
1652  004164  150377  175200                  BISB    R3,@DQREG       ;RESELECT SECONDARY ADDRESS
1653  004170  117704  175174                  MOVB    @DQREG,R4       ;READ DQREG
1654  004174  142704  000037                  BICB    #37,R4          ;CLEAR UNWANTED BITS
1655  004200  020504                          CMP     R5,R4           ;IS DATA CORRECT?
1656  004202  001401                          BEQ     3$              ;BR IF OK.
1657  004204  104003                          HLT     3               ;BUS ADDRESS EXTENSION MEMORY ERROR
1658  004206  104401                  3$:     SCOP1                   ;FREEZE ON CURRENT DATA? (SW09=1)
1659  004210  062703  000002                  ADD     #2,R3           ;UPDATE POINTER TO NEXT ADDRESS
1660  004214  005300                          DEC     R0              ;ALL ADDRESS DONE?
1661  004216  001357                          BNE     2$              ;BR IF NOT DONE.
1662  004220  104400                  4$:     SCOPE                   ;SCOPE THE TEST.
1663
1664                                          ;BUS ADDRESS MEMORY EXTENSION DATA TEST
1665                                          ;LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 140
1666                                          ;VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA
1667
1668                                  ;  TEST 13
1669                                  ;*************************************************************************
1670  004222  012737  000013  001226  TST13:  MOV     #13,TSTNO
1671  004230  012737  004372  001216          MOV     #CHKBB1,NEXT
1672  004236  012737  004326  001220          MOV     #2$,LOCK
1673  004244  012705  000140                  MOV     #140,R5
1674  004250  012700  000004                  MOV     #4,R0           ;4 LOCATIONS WILL BE ADDRESSED
1675  004254  005003                          CLR     R3              ;FIRST MEMORY ADDRESS =0
1676                                                                  ;(RECEIVER PRIMARY BUS ADDRESS)
1677  004256  010501                  1$:     MOV     R5,R1           ;LOAD "DATA"
1678  004260  050301                          BIS     R3,R1           ;LOAD SECONDARY ADDRESS
1679  004262  052701  000020                  BIS     #BIT4,R1        ;SET WRITE ENABLE.
1680  004266  112777  000020  175074          MOVB    #BIT4,@DQREG    ;SET MEMORY EXTENSION WRITE ENABLE
1681  004274  150377  175070                  BISB    R3,@DQREG       ;SELECT LOCATION TO BE TESTED
1682  004300  110177  175064                  MOVB    R1,@DQREG       ;LOAD DATA INTO MEMORY EXTENSION BITS
1683  004304  005077  175062                  CLR     @DQSEC          ;REFERENCE SEL 6 (DQSEC)
1684  004310  062703  000002                  ADD     #2,R3           ;NEXT LOCATION TO BE TESTED
1685  004314  005300                          DEC     R0              ;CONTINUE IF ALL LOCATIONS NOT LOADED
1686  004316  001357                          BNE     1$              ;
1687  004320  012700  000004                  MOV     #4,R0           ;4 LOCATIONS TO BE TESTED
1688  004324  005003                          CLR     R3              ;FIRST LOCATION
1689  004326  142777  000037  175034  2$:     BICB    #37,@DQREG      ;REINITIALIZE  ADDRESS
1690  004334  150377  175030                  BISB    R3,@DQREG       ;RESELECT SECONDARY ADDRESS
1691  004340  117704  175024                  MOVB    @DQREG,R4       ;READ DQREG
1692  004344  142704  000037                  BICB    #37,R4          ;CLEAR UNWANTED BITS
1693  004350  020504                          CMP     R5,R4           ;IS DATA CORRECT?
1694  004352  001401                          BEQ     3$              ;BR IF OK.
1695  004354  104003                          HLT     3               ;BUS ADDRESS EXTENSION MEMORY ERROR
1696  004356  104401                  3$:     SCOP1                   ;FREEZE ON CURRENT DATA? (SW09=1)
1697  004360  062703  000002                  ADD     #2,R3           ;UPDATE POINTER TO NEXT ADDRESS
1698  004364  005300                          DEC     R0              ;ALL ADDRESS DONE?
1699  004366  001357                          BNE     2$              ;BR IF NOT DONE.
1700  004370  104400                  4$:     SCOPE                   ;SCOPE THE TEST.
1701
1702                                          ;IF CHARACTER DETECT OPTION IS INSTALLED
```

```
                                        ;TESTS 14 THRU 25 WILL BE EXECUTED
                                        ;
004372  032737  020000  001510  CHKB81: BIT     #BBBIT,DQSTAT
004400  001002                        BNE     TST14
004402  000137  006206                JMP     TST26
                                        ;
                                        ;CHARACTER MEMORY ADDRESSING TEST
                                        ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH ITS
                                        ;ADDRESS (DUPLICATED EVERY 4 BITS)
                                        ;VERIFY THAT EACH LOCATION WAS ADDRESSED
                                        ;
                                        ;  TEST 14
                                        ;*************************************************************************
004406  012737  000014  001226  TST14:  MOV     #14,TSTNO
004414  012737  004546  001216          MOV     #TST15,NEXT
004422  012737  004512  001220          MOV     #2$,LOCK
004430  005005                        CLR     R5              ;CLEAR DATA TO BE LOADED
004432  012700  000020                MOV     #16.,R0         ;16(DECIMAL) REGISTERS WILL BE TESTED
004436  005002                        CLR     R2              ;FIRST ADDRESS =0
004440  012703  000010                MOV     #10,R3          ;CHARACTER MEMORY WILL BE SELECTED
004444  112777  000010  174716        MOVB    #10,@DQREG      ;SELECT CHARACTER MEMORY
004452  110277  174704          1$:   MOVB    R2,@DQRCSH      ;SELECT LOCATION IN CHARACTER MEMORY
004456  010577  174710                MOV     R5,@DQSEC       ;LOAD DATA
004462  005202                        INC     R2              ;UPDATE ADDRESS
004464  062705  010421                ADD     #10421,R5       ;UPDATE DATA
004470  005300                        DEC     R0              ;CONTINUE IF NOT DONE
004472  001367                        BNE     1$              ;
004474  005005                        CLR     R5              ;CLEAR EXPECTED DATA
004476  012700  000020                MOV     #16.,R0         ;16 (DECIMAL) LOCATIONS WILL BE TESTED
004502  005002                        CLR     R2              ;FIRST LOCATION =0
004504  112777  000010  174656        MOVB    #10,@DQREG      ;SELECT CHARACTER REGISTER
004512  110277  174644          2$:   MOVB    R2,@DQRCSH      ;SELECT ADDRESS TO BE TESTED
004516  017704  174650                MOV     @DQSEC,R4       ;READ ADDRESS TO BE TESTED
004522  020504                        CMP     R5,R4           ;ARE EXPECTED AND RECEIVED DATA THE SAME
004524  001401                        BEQ     3$              ;BR IF GOOD.
004526  104004                        HLT     4               ;CHARACTER MEMORY ADDRESS ERROR
004530  104401                3$:     SCOP1                   ;CHECK FOR LOOP ON CURRENT ADDRESS
004532  005202                        INC     R2              ;UPDATE ADDRESS TO BE TESTED
004534  062705  010421                ADD     #10421,R5       ;UPDATE EXPECTED DATA
004540  005300                        DEC     R0              ;CONTINUE IF NOT DONE
004542  001363                        BNE     2$              ;
004544  104400                4$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP
                                        ;
                                        ;SEQUENCE MEMORY ADDRESSING TEST
                                        ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH ITS
                                        ;ADDRESS (DUPLICATED EVERY 4 BITS)
                                        ;VERIFY THAT EACH LOCATION WAS ADDRESSED
                                        ;
                                        ;  TEST 15
                                        ;*************************************************************************
004546  012737  000015  001226  TST15:  MOV     #15,TSTNO
004554  012737  004706  001216          MOV     #TST16,NEXT
004562  012737  004652  001220          MOV     #2$,LOCK
004570  005005                        CLR     R5              ;CLEAR DATA TO BE LOADED
004572  012700  000020                MOV     #16.,R0         ;16(DECIMAL) REGISTERS WILL BE TESTED
004576  005002                        CLR     R2              ;FIRST ADDRESS =0
```

```
1759  004600  012703  000014                  MOV     #14,R3                  ;SEQUENCE MEMORY WILL BE SELECTED
1760  004604  112777  000014  174556          MOVB    #14,@DQREG              ;SELECT SEQUENCE MEMORY
1761  004612  110277  174544          1$:     MOVB    R2,@DQRCSH              ;SELECT LOCATION IN SEQUENCE MEMORY
1762  004616  010577  174550                  MOV     R5,@DQSEC               ;LOAD DATA
1763  004622  005202                          INC     R2                      ;UPDATE ADDRESS
1764  004624  062705  010421                  ADD     #10421,R5               ;UPDATE DATA
1765  004630  005300                          DEC     R0                      ;CONTINUE IF NOT DONE
1766  004632  001367                          BNE     1$                      ;
1767  004634  005005                          CLR     R5                      ;CLEAR EXPECTED DATA
1768  004636  012700  000020                  MOV     #16.,R0                 ;16 (DECIMAL) LOCATIONS WILL BE TESTED
1769  004642  005002                          CLR     R2                      ;FIRST LOCATION =0
1770  004644  112777  000014  174516          MOVB    #14,@DQREG              ;SELECT SEQUENCE REGISTER
1771  004652  110277  174504          2$:     MOVB    R2,@DQRCSH              ;SELECT ADDRESS TO BE TESTED
1772  004656  017704  174510                  MOV     @DQSEC,R4               ;READ ADDRESS TO BE TESTEC
1773  004662  020504                          CMP     R5,R4                   ;ARE EXPECTED AND RECEIVED DATA THE SAME
1774  004664  001401                          BEQ     3$                      ;BR IF GOOD.
1775  004666  104005                          HLT     5                       ;SEQUENCE MEMORY ADDRESS ERROR
1776  004670  104401                  3$:     SCOP1                           ;CHECK FOR LOOP ON CUREENT ADDRESS
1777  004672  005202                          INC     R2                      ;UPDATE ADDRESS TO BE TESTED
1778  004674  062705  010421                  ADD     #10421,R5               ;UPDATE EXPECTED DATA
1779  004700  005300                          DEC     R0                      ;CONTINUE IF NOT DONE
1780  004702  001363                          BNE     2$                      ;
1781  004704  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1782
1783                                          ;CHARACTER MEMORY TEST
1784                                          ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 000000
1785                                          ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
1786                                          ;LOADED WITH 000000
1787
1788                                  ;  TEST 16
1789                                  ;*************************************************************************
1790  004706  012737  000016  001226  TST16:  MOV     #16,TSTNO
1791  004714  012737  005036  001216          MOV     #TST17,NEXT
1792  004722  012737  005006  001220          MOV     #2$,LOCK
1793  004730  012705  000000                  MOV     #000000,R5              ;LOAD THE DATA INTO R5
1794  004734  012700  000020                  MOV     #16.,R0                 ;SET COUNT FOR 16
1795  004740  005002                          CLR     R2                      ;CLEAR REGISTER POINTER
1796  004742  012703  000010                  MOV     #10,R3                  ;GET REGISTER IN R3
1797  004746  112777  000010  174414          MOVB    #10,@DQREG              ;SELECT THE REGISTER
1798  004754  110277  174402          1$:     MOVB    R2,@DQRCSH              ;SELECT THE CHARACTER DET REG
1799  004760  010577  174406                  MOV     R5,@DQSEC               ;LOAD THE DATA
1800  004764  005202                          INC     R2                      ;UPDATE THE POINTER
1801  004766  005300                          DEC     R0                      ;ALL DONE YET?
1802  004770  001371                          BNE     1$                      ;BR IF NOT DONE
1803  004772  012700  000020                  MOV     #16.,R0                 ;SET FOR 16 REGISTERS TO BE TESTED
1804  004776  005002                          CLR     R2                      ;ZERO REG POINTER
1805  005000  112777  000010  174362          MOVB    #10,@DQREG              ;SELECT THE REGISTER
1806  005006  110277  174350          2$:     MOVB    R2,@DQRCSH              ;SELECT THE CHARACTER DET REGISTER
1807  005012  017704  174354                  MOV     @DQSEC,R4               ;READ THE DATA
1808  005016  020504                          CMP     R5,R4                   ;WAS IT LOADED CORRECTLY?
1809  005020  001401                          BEQ     3$                      ;BR IF DATA OK
1810  005022  104006                          HLT     6                       ;INCORRECT DATA REPORT ERROR
1811  005024  104401                  3$:     SCOP1                           ;CHECK FOR FREEZE ON DATA
1812  005026  005202                          INC     R2                      ;UPDATE POINTER
1813  005030  005300                          DEC     R0                      ;CHEACK FOR ALL DONE
1814  005032  001365                          BNE     2$                      ;BR IF NOT DONE
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 38
DZDQBC.P11    16-DEC-76 13:23          CHARACTER AND SEQUENCE MEMORY TESTS

 1815  005034  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
 1816
 1817                                          ;CHARACTER MEMORY TEST
 1818                                          ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 177777
 1819                                          ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
 1820                                          ;LOADED WITH 177777
 1821
 1822                                  ;  TEST 17
 1823                                  ;*****************************************************************************
 1824  005036  012737  000017  001226  TST17:  MOV     #17,TSTNO
 1825  005044  012737  005166  001216          MOV     #TST20,NEXT
 1826  005052  012737  005136  001220          MOV     #2$,LOCK
 1827  005060  012705  177777                 MOV     #177777,R5      ;LOAD THE DATA INTO R5
 1828  005064  012700  000020                 MOV     #16.,R0         ;SET COUNT FOR 16
 1829  005070  005002                          CLR     R2              ;CLEAR REGISTER POINTER
 1830  005072  012703  000010                 MOV     #10,R3          ;GET REGISTER IN R3
 1831  005076  112777  000010  174264         MOVB    #10,@DQREG      ;SELECT THE REGISTER
 1832  005104  110277  174252          1$:    MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
 1833  005110  010577  174256                 MOV     R5,@DQSEC       ;LOAD THE DATA
 1834  005114  005202                          INC     R2              ;UPDATE THE POINTER
 1835  005116  005300                          DEC     R0              ;ALL DONE YET?
 1836  005120  001371                          BNE     1$              ;BR IF NOT DONE
 1837  005122  012700  000020                 MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
 1838  005126  005002                          CLR     R2              ;ZERO REG POINTER
 1839  005130  112777  000010  174232         MOVB    #10,@DQREG      ;SELECT THE REGISTER
 1840  005136  110277  174220          2$:    MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
 1841  005142  017704  174224                 MOV     @DQSEC,R4       ;READ THE DATA
 1842  005146  020504                          CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
 1843  005150  001401                          BEQ     3$              ;BR IF DATA OK
 1844  005152  104006                          HLT     6               ;INCORRECT DATA REPORT ERROR
 1845  005154  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON DATA
 1846  005156  005202                          INC     R2              ;UPDATE POINTER
 1847  005160  005300                          DEC     R0              ;CHEACK FOR ALL DONE
 1848  005162  001365                          BNE     2$              ;BR IF NOT DONE
 1849  005164  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
 1850
 1851                                          ;CHARACTER MEMORY TEST
 1852                                          ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 125252
 1853                                          ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
 1854                                          ;LOADED WITH 125252
 1855
 1856                                  ;  TEST 20
 1857                                  ;*****************************************************************************
 1858  005166  012737  000020  001226  TST20:  MOV     #20,TSTNO
 1859  005174  012737  005316  001216          MOV     #TST21,NEXT
 1860  005202  012737  005266  001220          MOV     #2$,LOCK
 1861  005210  012705  125252                 MOV     #125252,R5      ;LOAD THE DATA INTO R5
 1862  005214  012700  000020                 MOV     #16.,R0         ;SET COUNT FOR 16
 1863  005220  005002                          CLR     R2              ;CLEAR REGISTER POINTER
 1864  005222  012703  000010                 MOV     #10,R3          ;GET REGISTER IN R3
 1865  005226  112777  000010  174134         MOVB    #10,@DQREG      ;SELECT THE REGISTER
 1866  005234  110277  174122          1$:    MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
 1867  005240  010577  174126                 MOV     R5,@DQSEC       ;LOAD THE DATA
 1868  005244  005202                          INC     R2              ;UPDATE THE POINTER
 1869  005246  005300                          DEC     R0              ;ALL DONE YET?
 1870  005250  001371                          BNE     1$              ;BR IF NOT DONE
```

```
[illegible]   012700  000020                   MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
[illegible]   005002                           CLR     R2              ;ZERO REG POINTER
[illegible]   112777  000010  174102           MOVB    #10,@DQREG      ;SELECT THE REGISTER
[illegible]   110277  174070          2$:      MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
[illegible]   017704  174074                   MOV     @DQSEC,R4       ;READ THE DATA
[illegible]   020504                           CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
[illegible]   001401                           BEQ     3$              ;BR IF DATA OK
[illegible]   104006                           HLT     6               ;INCORRECT DATA REPORT ERROR
[illegible]   104401                  3$:      SCOP1                   ;CHECK FOR FREEZE ON DATA
005306  005202                                 INC     R2              ;UPDATE POINTER
005310  005300                                 DEC     R0              ;CHEACK FOR ALL DONE
005312  001365                                 BNE     2$              ;BR IF NOT DONE
005314  104400                        4$:      SCOPE                   ;SCOPE THIS TEST

                                               ;CHARACTER MEMORY TEST
                                               ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 52525
                                               ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
                                               ;LOADED WITH 52525

                                      ;  TEST 21
                                      ;*************************************************************
005316  012737  000021  001226        TST21:   MOV     #21,TSTNO
005324  012737  005446  001216                 MOV     #TST22,NEXT
005332  012737  005416  001220                 MOV     #2$,LOCK
005340  012705  052525                         MOV     #52525,R5       ;LOAD THE DATA INTO R5
005344  012700  000020                         MOV     #16.,R0         ;SET COUNT FOR 16
005350  005002                                 CLR     R2              ;CLEAR REGISTER POINTFR
[illegible]   012703  000010                   MOV     #10,R3          ;GET REGISTER IN R3
[illegible]   112777  000010  174004           MOVB    #10,@DQREG      ;SELECT THE REGISTER
[illegible]   110277  173772          1$:      MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
[illegible]   010577  173776                   MOV     R5,@DQSEC       ;LOAD THE DATA
[illegible]   005202                           INC     R2              ;UPDATE THE POINTER
[illegible]   005300                           DEC     R0              ;ALL DONE YET?
[illegible]   001371                           BNE     1$              ;BR IF NOT DONE
[illegible]   012700  000020                   MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
[illegible]   005002                           CLR     R2              ;ZERO REG POINTER
[illegible]   112777  000010  173752           MOVB    #10,@DQREG      ;SELECT THE REGISTER
[illegible]   110277  173740          2$:      MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
[illegible]   017704  173744                   MOV     @DQSEC,R4       ;READ THE DATA
[illegible]   020504                           CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
[illegible]   001401                           BEQ     3$              ;BR IF DATA OK
[illegible]   104006                           HLT     6               ;INCORRECT DATA REPORT ERROR
[illegible]   104401                  3$:      SCOP1                   ;CHECK FOR FREEZE ON DATA
[illegible]   005202                           INC     R2              ;UPDATE POINTER
[illegible]   005300                           DEC     R0              ;CHEACK FOR ALL DONE
005442  001365                                 BNE     2$              ;BR IF NOT DONE
005444  104400                        4$:      SCOPE                   ;SCOPE THIS TEST

                                               ;SEQUENCE MEMORY TEST
                                               ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 000000
                                               ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
                                               ;LOADED WITH 000000

                                      ;  TEST 22
                                      ;*************************************************************
005446  012737  000022  001226        TST22:   MOV     #22,TSTNO
```

```
1927  005454  012737  005576  001216          MOV     #TST23,NEXT
1928  005462  012737  005546  001220          MOV     #2$,LOCK
1929  005470  012705  000000                  MOV     #000000,R5      ;LOAD THE DATA INTO R5
1930  005474  012700  000020                  MOV     #16.,R0         ;SET COUNT FOR 16
1931  005500  005002                          CLR     R2              ;CLEAR REGISTER POINTER
1932  005502  012703  000014                  MOV     #14,R3          ;GET REGISTER IN R3
1933  005506  112777  000014  173654          MOVB    #14,@DQREG      ;SELECT THE REGISTER
1934  005514  110277  173642          1$:     MOVB    R2,@DQRCSH      ;SELECT THE     ACTER DET REG
1935  005520  010577  173646                  MOV     R5,@DQSEC       ;LOAD THE D
1936  005524  005202                          INC     R2              ;UPDATE THE     NTER
1937  005526  005300                          DEC     R0              ;ALL DONE
1938  005530  001371                          BNE     1$              ;BR IF NOT DONE
1939  005532  012700  000020                  MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
1940  005536  005002                          CLR     R2              ;ZERO REG POINTER
1941  005540  112777  000014  173622          MOVB    #14,@DQREG      ;SELECT THE REGISTER
1942  005546  110277  173610          2$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
1943  005552  017704  173614                  MOV     @DQSEC,R4       ;READ THE DATA
1944  005556  020504                          CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
1945  005560  001401                          BEQ     3$              ;BR IF DATA OK
1946  005562  104007                          HLT     7               ;INCORRECT DATA REPORT ERROR
1947  005564  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON DATA
1948  005566  005202                          INC     R2              ;UPDATE POINTER
1949  005570  005300                          DEC     R0              ;CHEACK FOR ALL DONE
1950  005572  001365                          BNE     2$              ;BR IF NOT DONE
1951  005574  104400                  4$:     SCOPE                   ;SCOPE THIS TEST
1952
1953                                          ;SEQUENCE MEMORY TEST
1954                                          ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 177777
1955                                          ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
1956                                          ;LOADED WITH 177777
1957
1958                                  ;  TEST 23
1959                                  ;*************************************************************************
1960  005576  012737  000023  001226  TST23:  MOV     #23,TSTNO
1961  005604  012737  005726  001216          MOV     #TST24,NEXT
1962  005612  012737  005676  001220          MOV     #2$,LOCK
1963  005620  012705  177777                  MOV     #177777,R5      ;LOAD THE DATA INTO R5
1964  005624  012700  000020                  MOV     #16.,R0         ;SET COUNT FOR 16
1965  005630  005002                          CLR     R2              ;CLEAR REGISTER POINTER
1966  005632  012703  000014                  MOV     #14,R3          ;GET REGISTER IN R3
1967  005636  112777  000014  173524          MOVB    #14,@DQREG      ;SELECT THE REGISTER
1968  005644  110277  173512          1$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
1969  005650  010577  173516                  MOV     R5,@DQSEC       ;LOAD THE DATA
1970  005654  005202                          INC     R2              ;UPDATE THE POINTER
1971  005656  005300                          DEC     R0              ;ALL DONE YET?
1972  005660  001371                          BNE     1$              ;BR IF NOT DONE
1973  005662  012700  000020                  MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
1974  005666  005002                          CLR     R2              ;ZERO REG POINTER
1975  005670  112777  000014  173472          MOVB    #14,@DQREG      ;SELECT THE REGISTER
1976  005676  110277  173460          2$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
1977  005702  017704  173464                  MOV     @DQSEC,R4       ;READ THE DATA
1978  005706  020504                          CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
1979  005710  001401                          BEQ     3$              ;BR IF DATA OK
1980  005712  104007                          HLT     7               ;INCORRECT DATA REPORT ERROR
1981  005714  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON DATA
1982  005716  005202                          INC     R2              ;UPDATE POINTER
```

```
                                        DEC     R0              ;CHEACK FOR ALL DONE
                                        BNE     2$              ;BR IF NOT DONE
                                4$:     SCOPE                   ;SCOPE THIS TEST

                                        ;SEQUENCE MEMORY TEST
                                        ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 125252
                                        ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
                                        ;LOADED WITH 125252

                                ;  TEST 24
                                ;*************************************************************************
005726  012737  000024  001226  TST24:  MOV     #24,TSTNO
005734  012737  006056  001216          MOV     #TST25,NEXT
005742  012737  006026  001220          MOV     #2$,LOCK
005750  012705  125252                  MOV     #125252,R5      ;LOAD THE DATA INTO R5
005754  012700  000020                  MOV     #16.,R0         ;SET COUNT FOR 16
005760  005002                          CLR     R2              ;CLEAR REGISTER POINTER
005762  012703  000014                  MOV     #14,R3          ;GET REGISTER IN R3
005766  112777  000014  173374          MOVB    #14,@DQPEG      ;SELECT THE REGISTER
005774  110277  173362          1$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
006000  010577  173366                  MOV     R5,@DQSEC       ;LOAD THE DATA
006004  005202                          INC     R2              ;UPDATE THE POINTER
006006  005300                          DEC     R0              ;ALL DONE YET?
006010  001371                          BNE     1$              ;BR IF NOT DONE
006012  012700  000020                  MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
006016  005002                          CLR     R2              ;ZERO REG POINTER
006020  112777  000014  173342          MOVB    #14,@DQREG      ;SELECT THE REGISTER
006026  110277  173330          2$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
006032  017704  173334                  MOV     @DQSEC,R4       ;READ THE DATA
006036  020504                          CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
006040  001401                          BEQ     3$              ;BR IF DATA OK
006042  104007                          HLT     7               ;INCORRECT DATA REPORT ERROR
006044  104401                  3$:     SCOP1                   ;CHECK FOR FREEZE ON DATA
006046  005202                          INC     R2              ;UPDATE POINTER
006050  005300                          DEC     R0              ;CHEACK FOR ALL DONE
006052  001365                          BNE     2$              ;BR IF NOT DONE
006054  104400                  4$:     SCOPE                   ;SCOPE THIS TEST

                                        ;SEQUENCE MEMORY TEST
                                        ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 52525
                                        ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
                                        ;LOADED WITH 52525

                                ;  TEST 25
                                ;*************************************************************************
006056  012737  000025  001226  TST25:  MOV     #25,TSTNO
006064  012737  006206  001216          MOV     #TST26,NEXT
006072  012737  006156  001220          MOV     #2$,LOCK
006100  012705  052525                  MOV     #52525,R5       ;LOAD THE DATA INTO R5
006104  012700  000020                  MOV     #16.,R0         ;SET COUNT FOR 16
006110  005002                          CLR     R2              ;CLEAR REGISTER POINTER
006112  012703  000014                  MOV     #14,R3          ;GET REGISTER IN R3
006116  112777  000014  173244          MOVB    #14,@DQREG      ;SELECT THE REGISTER
006124  110277  173232          1$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REG
006130  010577  173236                  MOV     R5,@DQSEC       ;LOAD THE DATA
006134  005202                          INC     R2              ;UPDATE THE POINTER
```

```
                                        DEC     R0              ;ALL DONE YET?
                                        BNE     1$              ;BR IF NOT DONE
                                        MOV     #16.,R0         ;SET FOR 16 REGISTERS TO BE TESTED
                                        CLR     R2              ;ZERO REG POINTER
                                        MOVB    #14,@DQREG      ;SELECT THE REGISTER
                                2$:     MOVB    R2,@DQRCSH      ;SELECT THE CHARACTER DET REGISTER
                                        MOV     @DQSEC,R4       ;READ THE DATA
                                        CMP     R5,R4           ;WAS IT LOADED CORRECTLY?
                                        BEQ     3$              ;BR IF DATA OK
                                        HLT     7               ;INCORRECT DATA REPORT ERROR
                                3$:     SCOP1                   ;CHECK FOR FREEZE ON DATA
                                        INC     R2              ;UPDATE POINTER
                                        DEC     R0              ;CHEACK FOR ALL DONE
                                        BNE     2$              ;BR IF NOT DONE
                                4$:     SCOPE                   ;SCOPE THIS TEST

                                        ;RECEIVER CONTROL REGISTER MASTER CLEAR TEST
                                        ;SET ALL READ/WRITE BITS IN RECEIVER CONTROL REGISTER
                                        ;ISSUE MASTER CLEAR
                                        ;VERIFY THAT RECEIVER CONTROL WAS CLEARED

                                ;  TEST 26
                                ;*************************************************************************
006206  012737  000026  001226  TST26:  MOV     #26,TSTNO
006214  012737  006272  001216          MOV     #TST27,NEXT
006222  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERRUPTS
006230  013703  001360                  MOV     DQRCSR,R3               ;BUS ADDRESS OF
                                                                        ;RECEIVER CONTROL REGISTER
006234  012777  170372  173116          MOV     #170372,@DQRCSR         ;LOAD RECEIVER CONTROL REGISTER
006242  112777  000012  173120          MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
006250  052777  000040  173114          BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
006256  005005                          CLR     R5                      ;(R5)=EXPECTED 170372 IN
                                                                        ;RECEIVER CONTROL REGISTER, 0
006260  017704  173074                  MOV     @DQRCSR,R4              ;(R4)=ACTUAL 170372 IN
                                                                        ;RECEIVER CONTROL REGISTER
006264  001401                          BEQ     1$                      ;BR IF ALL OK.
006266  104000                          HLT                             ;PRIMARY REGISTER MASTER CLEAR ERROR
006270  104400                  1$:     SCOPE

                                        ;RECEIVER CONTROL REGISTER MASTER CLEAR TEST
                                        ;SET ALL READ/WRITE BITS IN RECEIVER CONTROL REGISTER
                                        ;ISSUE MASTER CLEAR
                                        ;VERIFY THAT RECEIVER CONTROL WAS CLEARED

                                ;  TEST 27
                                ;*************************************************************************
006272  012737  000027  001226  TST27:  MOV     #27,TSTNO
006300  012737  006356  001216          MOV     #CHKBA1,NEXT
006306  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERRUPTS
006314  013703  001360                  MOV     DQRCSR,R3               ;BUS ADDRESS OF
                                                                        ;RECEIVER CONTROL REGISTER
006320  012777  007400  173032          MOV     #07400,@DQRCSR          ;LOAD RECEIVER CONTROL REGISTER
006326  112777  000012  173034          MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
006334  052777  000040  173030          BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
006342  005005                          CLR     R5                      ;(R5)=EXPECTED 07400 IN
                                                                        ;RECEIVER CONTROL REGISTER, 0
```

```
                        VERIFICATION OF MASTER CLEAR LOGIC

2095  006344  017704  173010                MOV     @DQRCSR,R4              ;(R4)=ACTUAL 07400 IN
2096                                                                          ;RECEIVER CONTROL REGISTER
2097  006350  001401                       BEQ     1$                      ;BR IF ALL OK.
2098  006352  104000                       HLT                             ;PRIMARY REGISTER MASTER CLEAR ERROR
2099  006354  104400                1$:    SCOPE
2100
2101                                        ;IF DATASET CONTROL OPTION IS INSTALLED
2102                                        ;TEST 30 WILL BE EXECUTED
2103
2104  006356  032737  010000  001510  CHKBA1: BIT     #BABIT,DQSTAT
2105  006364  001436                       BEQ     TST31
2106
2107
2108                                        ;TRANSMITTER CONTROL REGISTER MASTER CLEAR TEST
2109                                        ;SET ALL READ/WRITE BITS IN TRANSMITTER CONTROL REGISTER
2110                                        ;ISSUE MASTER CLEAR
2111                                        ;VERIFY THAT TRANSMITTER CONTROL WAS CLEARED
2112
2113                                ;  TEST 30
2114                                ;*************************************************************************
2115  006366  012737  000030  001226  TST30:  MOV     #30,TSTNO
2116  006374  012737  006460  001216         MOV     #CHKBA2,NEXT
2117  006402  012737  000340  177776         MOV     #340,PS                 ;LOCK OUT INTERRUPTS
2118  006410  013703  001364                MOV     DQTCSR,R3               ;BUS ADDRESS OF
2119                                                                          ;TRANSMITTER CONTROL REGISTER
2120  006414  012777  101772  172742         MOV     #101772,@DQTCSR         ;LOAD TRANSMITTER CONTROL REGISTER
2121  006422  112777  000012  172740         MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
2122  006430  052777  000040  172734         BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
2123  006436  005005                        CLR     R5                      ;(R5)=EXPECTED 101772 IN
2124                                                                          ;TRANSMITTER CONTROL REGISTER, 0
2125  006440  017704  172720                MOV     @DQTCSR,R4              ;(R4)=ACTUAL 101772 IN
2126                                                                          ;TRANSMITTER CONTROL REGISTER
2127  006444  042704  100000                BIC     #100000,R4              ;100000 OFF UNWANTED BITS.
2128  006450  020504                        CMP     R5,R4                   ;EXPECTED=ACTUAL?
2129  006452  001401                        BEQ     1$                      ;BR IF ALL OK.
2130  006454  104000                        HLT                             ;PRIMARY REGISTER MASTER CLEAR ERROR
2131  006456  104400                1$:    SCOPE
2132  006460  000435                CHKBA2: BR      TST32
2133
2134                                        ;IF DATASET CONTROL OPTION IS NOT INSTALLED
2135                                        ;TEST 31 WILL BE EXECUTED
2136
2137
2138                                        ;TRANSMITTER CONTROL REGISTER MASTER CLEAR TEST
2139                                        ;SET ALL READ/WRITE BITS IN TRANSMITTER CONTROL REGISTER
2140                                        ;ISSUE MASTER CLEAR
2141                                        ;VERIFY THAT TRANSMITTER CONTROL WAS CLEARED
2142
2143                                ;  TEST 31
2144                                ;*************************************************************************
2145  006462  012737  000031  001226  TST31:  MOV     #31,TSTNO
2146  006470  012737  006554  001216         MOV     #TST32,NEXT
2147  006476  012737  000340  177776         MOV     #340,PS                 ;LOCK OUT INTERRUPTS
2148  006504  013703  001364                MOV     DQTCSR,R3               ;BUS ADDRESS OF
2149                                                                          ;TRANSMITTER CONTROL REGISTER
2150  006510  012777  101772  172646         MOV     #101772,@DQTCSR         ;LOAD TRANSMITTER CONTROL REGISTER
```

```
2151  006516  112777  000012  172644          MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
2152  006524  052777  000040  172640          BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
2153  006532  005005                          CLR     R5                      ;(R5)=EXPECTED 101772 IN
2154                                                                          ;TRANSMITTER CONTROL REGISTER, 0
2155  006534  017704  172624                  MOV     @DQTCSR,R4              ;(R4)=ACTUAL 101772 IN
2156                                                                          ;TRANSMITTER CONTROL REGISTER
2157  006540  042704  001400                  BIC     #1400,R4                ;1400 OFF UNWANTED BITS.
2158  006544  020504                          CMP     R5,R4                   ;EXPECTED=ACTUAL?
2159  006546  001401                          BEQ     1$                      ;BR IF ALL OK.
2160  006550  104000                          HLT                             ;PRIMARY REGISTER MASTER CLEAR ERROR
2161  006552  104400              1$:         SCOPE
2162
2163                                          ;ERROR REGISTER MASTER CLEAR TEST
2164                                          ;SET ALL READ/WRITE BITS IN ERROR REGISTER
2165                                          ;ISSUE MASTER CLEAR
2166                                          ;VERIFY THAT ERROR WAS CLEARED
2167
2168                                  ;  TEST 32
2169                                  ;*************************************************************************
2170  006554  012737  000032  001226  TST32:  MOV     #32,TSTNO
2171  006562  012737  006646  001216          MOV     #TST33,NEXT
2172  006570  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERRUPTS
2173  006576  013703  001366                  MOV     @#ERR,R3                ;BUS ADDRESS OF
2174                                                                          ;ERROR REGISTER
2175  006602  012777  107777  172556          MOV     #107777,@DQERR          ;LOAD ERROR REGISTER
2176  006610  112777  000012  172552          MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
2177  006616  052777  000040  172546          BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
2178  006624  005005                          CLR     R5                      ;(R5)=EXPECTED 107777 IN
2179                                                                          ;ERROR REGISTER, 0
2180  006626  017704  172534                  MOV     @DQERR,R4               ;(R4)=ACTUAL 107777 IN
2181                                                                          ;ERROR REGISTER
2182  006632  042704  060000                  BIC     #60000,R4               ;60000 OFF UNWANTED BITS.
2183  006636  020504                          CMP     R5,R4                   ;EXPECTED=ACTUAL?
2184  006640  001401                          BEQ     1$                      ;BR IF ALL OK.
2185  006642  104000                          HLT                             ;PRIMARY REGISTER MASTER CLEAR ERROR
2186  006644  104400              1$:         SCOPE
2187
2188                                          ;SYNC REGISTER MASTER CLEAR TEST
2189                                          ;SET ALL READ/WRITE BITS IN SYNC REGISTER
2190                                          ;ISSUE MASTER CLEAR
2191                                          ;VERIFY THAT SYNC WAS CLEARED
2192
2193                                  ;  TEST 33
2194                                  ;*************************************************************************
2195  006646  012737  000033  001226  TST33:  MOV     #33,TSTNO
2196  006654  012737  006742  001216          MOV     #TST34,NEXT
2197  006662  012737  000340  177776          MOV     #340,PS                 ;LOCK OUT INTERRUPTS
2198  006670  012703  000011                  MOV     #11,R3                  ;GET ADDRESS OF SECONDARY
2199                                                                          ;REGISTER TO BE TESTED
2200  006674  110377  172470                  MOVB    R3,@DQREG               ;SELECT CEREG
2201                                                                          ;SECONDARY REGISTER
2202  006700  012777  177777  172464          MOV     #177777,@DQSEC          ;LOAD SYNC REGISTER
2203  006706  112777  000012  172454          MOVB    #12,@DQREG              ;SELECT MISCELLANEOUS REGISTER
2204  006714  052777  000040  172450          BIS     #BIT5,@DQSEC            ;ISSUE MASTER CLEAR
2205  006722  110377  172442                  MOVB    R3,@DQREG               ;SELECT SYNC SECONDARY REGISTER
2206  006726  005005                          CLR     R5                      ;(R5)=EXPECTED 177777 IN
```

```
2207                                                                    ;SYNC REGISTER, 0
2208  006730  017704  172436                MOV     @DQSEC,R4         ;(R4)=ACTUAL 177777 IN
2209                                                                    ;SYNC REGISTER
2210  006734  001401                        BEQ     1$                ;BR IF ALL OK.
2211  006736  104000                        HLT                       ;SECONDARY REGISTER MASTER CLEAR ERROR
2212  006740  104400                1$:     SCOPE
2213
2214                                        ;MISCELLANEOUS REGISTER MASTER CLEAR TEST
2215                                        ;SET ALL READ/WRITE BITS IN MISCELLANEOUS REGISTER
2216                                        ;ISSUE MASTER CLEAR
2217                                        ;VERIFY THAT MISCELLANEOUS WAS CLEARED
2218
2219                                ;  TEST 34
2220                                ;*************************************************************************
2221  006742  012737  000034  001226 TST34:  MOV     #34,TSTNO
2222  006750  012737  007036  001216        MOV     #CHKAB1,NEXT
2223  006756  012737  000340  177776        MOV     #340,PS           ;LOCK OUT INTERRUPTS
2224  006764  012703  000012                MOV     #12,R3            ;GET ADDRESS OF SECONDARY
2225                                                                    ;REGISTER TO BE TESTED
2226  006770  110377  172374                MOVB    R3,@DQREG         ;SELECT CEREG
2227                                                                    ;SECONDARY REGISTER
2228  006774  012777  007717  172370        MOV     #7717,@DQSEC      ;LOAD MISCELLANEOUS REGISTER
2229  007002  112777  000012  172360        MOVB    #12,@DQREG        ;SELECT MISCELLANEOUS REGISTER
2230  007010  052777  000040  172354        BIS     #BIT5,@DQSEC      ;ISSUE MASTER CLEAR
2231  007016  110377  172346                MOVB    R3,@DQREG         ;SELECT MISCELLANEOUS SECONDARY REGISTER
2232  007022  005005                        CLR     R5                ;(R5)=EXPECTED 7717 IN
2233                                                                    ;MISCELLANEOUS REGISTER, 0
2234  007024  017704  172342                MOV     @DQSEC,R4         ;(R4)=ACTUAL 7717 IN
2235                                                                    ;MISCELLANEOUS REGISTER
2236  007030  001401                        BEQ     1$                ;BR IF ALL OK.
2237  007032  104000                        HLT                       ;SECONDARY REGISTER MASTER CLEAR ERROR
2238  007034  104400                1$:     SCOPE
2239
2240                                        ;IF BCC OPTION IS INSTALLED
2241                                        ;TEST 35 WILL BE EXECUTED
2242
2243  007036  032737  002000  001510 CHKAB1: BIT     #ABBIT,DQSTAT
2244  007044  001436                        BEQ     .EOP
2245
2246                                        ;BCC POLYNOMIAL REGISTER MASTER CLEAR TEST
2247                                        ;SET ALL READ/WRITE BITS IN BCC POLYNOMIAL REGISTER
2248                                        ;ISSUE MASTER CLEAR
2249                                        ;VERIFY THAT BCC POLYNOMIAL WAS CLEARED
2250
2251                                ;  TEST 35
2252                                ;*************************************************************************
2253  007046  012737  000035  001226 TST35:  MOV     #35,TSTNO
2254  007054  012737  007142  001216        MOV     #.EOP,NEXT
2255  007062  012737  000340  177776        MOV     #340,PS           ;LOCK OUT INTERRUPTS
2256  007070  012703  000017                MOV     #17,R3            ;GET ADDRESS OF SECONDARY
2257                                                                    ;REGISTER TO BE TESTED
2258  007074  110377  172270                MOVB    R3,@DQREG         ;SELECT CEREG
2259                                                                    ;SECONDARY REGISTER
2260  007100  012777  177777  172264        MOV     #177777,@DQSEC    ;LOAD BCC POLYNOMIAL REGISTER
2261  007106  112777  000012  172254        MOVB    #12,@DQREG        ;SELECT MISCELLANEOUS REGISTER
2262  007114  052777  000040  172250        BIS     #BIT5,@DQSEC      ;ISSUE MASTER CLEAR
```

```
2263  007122  110377  172242              MOVB    R3,@DQREG       ;SELECT BCC POLYNOMIAL SECONDARY REGISTER
2264  007126  005005                      CLR     R5              ;(R5)=EXPECTED 177777 IN
2265                                                              ;BCC POLYNOMIAL REGISTER, 0
2266  007130  017704  172236              MOV     @DQSEC,R4       ;(R4)=ACTUAL 177777 IN
2267                                                              ;BCC POLYNOMIAL REGISTER
2268  007134  001401                      BEQ     1$              ;BR IF ALL OK.
2269  007136  104000                      HLT                     ;SECONDARY REGISTER MASTER CLEAR ERROR
2270  007140  104400              1$:     SCOPE
```

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

007142  005037  001234          .EOP:   CLR     LSTERR                  ;CLEAR LAST ERROR PC
007146  005037  001312                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
007152  005237  001230                  INC     PASCNT                  ;UPDATE PASS COUNT
007156  104402                          TYPE
007160  011372                          MEPASS
007162  104402                          TYPE
007164  011553                          MCSRX
007166  104411                          CNVRT
007170  007300                          XCSR
007172  104402                          TYPE
007174  011561                          MVECX
007176  104411                          CNVRT
007200  007306                          XVEC
007202  104402                          TYPE
007204  011567                          MPASSX
007206  104411                          CNVRT
007210  007314                          XPASS
007212  104402                          TYPE
007214  011600                          MERRX
007216  104411                          CNVRT
007220  007322                          XERR
007222  013777  001230  171752          MOV     PASCNT,@LIGHTS          ;DISPLAY PASS COUNT
007230  005337  001276                  DEC     SAVNUM
007234  001013                          BNE     RESTRT
007236  013737  001504  001276          MOV     DQNUM,SAVNUM
007244  013701  000042                  MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
007250  001405                          BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
007252  000005                          RESET
007254                          LOGICAL:
007254  004711                          JSR     PC,(R1)
007256  000240                          NOP
007260  000240                          NOP
007262  000240                          NOP
007264  104414                  RESTRT: CKSWR
007266  012737  002254  001214          MOV     #TST1,RETURN
007274  000137  002254                  JMP     TST1
007300  000001                  XCSR:   1
007302     006     002                  .BYTE   6,2
007304  001360                          DQRCSR
007306  000001                  XVEC:   1
007310     003     002                  .BYTE   3,2
007312  001350                          DQRVEC
007314  000001                  XPASS:  1
007316     006     002                  .BYTE   6,2
007320  001230                          PASCNT
007322  000001                  XERR:   1
007324     006     002                  .BYTE   6,2
007326  001232                          ERRCNT
```

```
2327                                                    ;SCOPE LOOP AND INTERATION HANDLER
2328
2329  007330  104414                          .SCOPE:   CKSWR
2330  007332  032777  040000  171640                    BIT     #BIT14,@SWR
2331  007340  001407                          TTST:     BEQ     1$
2332  007342  000432                                    BR      3$
2333  007344  105777  171634                            TSTB    @TKCSR
2334  007350  100027                                    BPL     3$
2335  007352  017700  171630                            MOV     @TKDBR,R0
2336  007356  000412                                    BR      2$
2337  007360  032777  004000  171612  1$:       BIT     #SW11,@SWR
2338  007366  001006                                    BNE     2$
2339  007370  005237  001224                            INC     LPCNT
2340  007374  023737  001224  001222                    CMP     LPCNT,ICOUNT
2341  007402  001012                                    BNE     3$
2342  007404  105037  001312                  2$:       CLRB    ERRFLG
2343  007410  005037  001224                            CLR     LPCNT
2344  007414  012737  002000  001222                    MOV     #2000,ICOUNT
2345  007422  013737  001216  001214                    MOV     NEXT,RETURN
2346  007430  013716  001214                  3$:       MOV     RETURN,(SP)
2347  007434  000002                                    RTI
2348  007436  001407                          BRW:      1407
2349  007440  000432                          BRX:      432
2350
2351                                                    ;CHECK FOR FREEZE ON CURRENT DATA
2352
2353  007442  104414                          .SCOP1:   CKSWR
2354  007444  032777  001000  171526                    BIT     #SW09,@SWR
2355  007452  001402                                    BEQ     1$
2356  007454  013716  001220                            MOV     LOCK,(SP)
2357  007460  000002                          1$:       RTI
2358
2359                                                    ;TELETYPE OUTPUT ROUTINE
2360
2361  007462  010546                          .TYPE:    MOV     R5,-(SP)
2362  007464  017605  000002                            MOV     @2(SP),R5
2363  007470  062766  000002  000002                    ADD     #2,2(SP)
2364  007476  005737  011152                  1$:       TST     @#RDSW
2365  007502  001004                                    BNE     300$
2366  007504  032777  010000  171466                    BIT     #SW12,@SWR
2367  007512  001024                                    BNE     3$
2368  007514  105715                          300$:     TSTB    (R5)
2369  007516  100014                                    BPL     2$
2370  007520  105777  171464                            TSTB    @TPCSR
2371  007524  100375                                    BPL     .-4
2372  007526  012777  000015  171456                    MOV     #15,@TPDBR
2373  007534  105777  171450                            TSTB    @TPCSR
2374  007540  100375                                    BPL     .-4
2375  007542  012777  000012  171442                    MOV     #12,@TPDBR
2376  007550  105777  171434                  2$:       TSTB    @TPCSR
2377  007554  100375                                    BPL     2$
2378  007556  112577  171430                            MOVB    (R5)+,@TPDBR
2379  007562  001345                                    BNE     1$
2380  007564  012605                          3$:       MOV     (SP)+,R5
2381  007566  000002                                    RTI
2382
```

```
2383                                                    ;ASCII STRING INPUT ROUTINE
2384
2385  007570  010346                        .INSTR:   MOV     R3,-(SP)
2386  007572  010446                                  MOV     R4,-(SP)
2387  007574  017637  000004  007612                  MOV     @4(SP),.MSG
2388  007602  062766  000002  000004                  ADD     #2,4(SP)
2389  007610  104402                        .INST1:   TYPE
2390  007612  0000 0                        .MSG:     0
2391  007614  012704  011744                          MOV     #INBUF,R4
2392  007620  012703  000007                          MOV     #7,R3
2393  007624  105777  171354              1$:         TSTB    @TKCSR
2394  007630  100375                                  BPL     1$
2395  007632  117714  171350                          MOVB    @TKDBR,(R4)
2396  007636  142714  000200                          BICB    #200,(R4)
2397  007642  121427  000025                          CMPB    (R4),#25                ;IS IT <↑G>
2398  007646  001003                                  BNE     200$
2399  007650  104402  011332                          TYPE,MCRLF
2400  007654  000755                                  BR      .INST1
2401  007656  122427  000015              200$:       CMPB    (R4)+,#15
2402  007662  001423                                  BEQ     INSTR2
2403  007664  117777  171316  171320                  MOVB    @TKDBR,@TPDBR
2404  007672  105777  171312              2$:         TSTB    @TPCSR
2405  007676  100375                                  BPL     2$
2406  007700  005303                                  DEC     R3
2407  007702  001350                                  BNE     1$
2408  007704  000402                                  BR      .INSTG
2409  007706  010346                        .INSTE:   MOV     R3,-(SP)
2410  007710  010446                                  MOV     R4,-(SP)
2411  007712  104402                        .INSTG:   TYPE
2412  007714  011326                                  MQM
2413  007716  005737  011152                          TST     @#RDSW
2414  007722  001402                                  BEQ     400$
2415  007724  104402  011332                          TYPE,MCRLF
2416  007730  000727                        400$:     BR      .INST1
2417  007732  012604                        INSTR2:   MOV     (SP)+,R4
2418  007734  012603                                  MOV     (SP)+,R3
2419  007736  000002                                  RTI
2420
2421                                                    ;CONVERT ASCII STRING TO OCTAL
2422
2423  007740  010546                        .PARAM:   MOV     R5,-(SP)
2424  007742  010446                                  MOV     R4,-(SP)
2425  007744  016605  000004                          MOV     4(SP),R5
2426  007750  012537  010144                          MOV     (R5)+,LOLIM
2427  007754  012537  010146                          MOV     (R5)+,HILIM
2428  007760  012537  010150                          MOV     (R5)+,DEVADR
2429  007764  112537  010152                          MOVB    (R5)+,LOBITS
2430  007770  112537  010153                          MOVB    (R5)+,ADRCNT
2431  007774  016 6   000004                          MOV     R5,4(SP)
2432  010000  005 5                         PARAM1:   CLR     R5
2433  010002  012704  011744                          MOV     #INBUF,R4
2434  010006  122714  000015                          CMPB    #15,(R4)
2435  010012  001420                                  BEQ     PARERR
2436  010014  121427  000060              1$:         CMPB    (R4),#60
2437  010020  002415                                  BLT     PARERR
2438  010022  121427  000067                          CMPB    (R4),#67
```

```
2439  010026  003012                        BGT     PARERR
2440  010030  142714  000060                BICB    #60,(R4)
2441  010034  152405                        BISB    (R4)+,R5
2442  010036  122714  000015                CMPB    #15,(R4)
2443  010042  001414                        BEQ     LIMITS
2444  010044  006305                        ASL     R5
2445  010046  006305                        ASL     R5
2446  010050  006305                        ASL     R5
2447  010052  000760                        BR      1$
2448  010054  122714  000015        PARERR: CMPB    #15,(R4)                ;IS FIRST CHARACTER A <CR>
2449  010060  001003                        BNE     120$
2450  010062  005737  011152                TST     @#RDSW                  ;IS CKSWR ROUTINE BEING USED
2451  010066  001023                        BNE     PARTI
2452  010070  104404                120$:   INSTER
2453  010072  000742                        BR      PARAM1
2454
2455                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2456
2457  010074  020537  010146        LIMITS: CMP     R5,HILIM
2458  010100  101365                        BHI     PARERR
2459  010102  020537  010144                CMP     R5,LOLIM
2460  010106  103762                        BLO     PARERR
2461  010110  133705  010152                BITB    LOBITS,R5
2462  010114  001357                        BNE     PARERR
2463
2464                                        ;STORE NUMBER AT SPECIFIED ADDRESS
2465
2466  010116  013704  010150                MOV     DEVADR,R4
2467  010122  010524                1$:     MOV     R5,(R4)+
2468  010124  062705  000002                ADD     #2,R5
2469  010130  105337  010153                DECB    ADRCNT
2470  010134  001372                        BNE     1$
2471  010136  012604                PARTI:  MOV     (SP)+,R4
2472  010140  012605                        MOV     (SP)+,R5
2473  010142  000002                        RTI
2474  010144  000000                LOLIM:  0
2475  010146  000000                HILIM:  0
2476  010150  000000                DEVADR: 0
2477  010152  000000                LOBITS: 0
2478          010153                ADRCNT=LOBITS+1
2479
2480                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
2481
2482  010154  016637  000004  001274  .SAV05: MOV     4(SP),SAVPC
2483
2484                                        ;SAVE R0-R5
2485
2486  010162  010537  001270        SV05:   MOV     R5,SAVR5
2487  010166  010437  001266                MOV     R4,SAVR4
2488  010172  010337  001264                MOV     R3,SAVR3
2489  010176  010237  001262                MOV     R2,SAVR2
2490  010202  010137  001260                MOV     R1,SAVR1
2491  010206  010037  001256                MOV     R0,SAVR0
2492  010212  000002                        RTI
2493
2494                                        ;RESTORE R0-R5
```

```
2495
2496  010214  013700  001256          .RESO5: MOV     SAVR0,R0
2497  010220  013701  001260                  MOV     SAVR1,R1
2498  010224  013702  001262                  MOV     SAVR2,R2
2499  010230  013703  001264                  MOV     SAVR3,R3
2500  010234  013704  001266                  MOV     SAVR4,R4
2501  010240  013705  001270                  MOV     SAVR5,R5
2502  010244  000002                         RTI
2503
2504                                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2505
2506  010246  104402                .CONVR: TYPE
2507  010250  011332                        MCRLF
2508  010252  010046                .CNVRT: MOV     R0,-(SP)
2509  010254  010146                        MOV     R1,-(SP)
2510  010256  010346                        MOV     R3,-(SP)
2511  010260  010446                        MOV     R4,-(SP)
2512  010262  010546                        MOV     R5,-(SP)
2513  010264  017601  C.0012                MOV     @12(SP),R1
2514  010270  013737  012006  001250        MOV     TEMP,TEMP3
2515  010276  062766  000002  000012        ADD     #2,12(SP)
2516  010304  012137  010466                MOV     (R1)+,WRDCNT
2517  010310  112137  010470          1$:   MOVB    (R1)+,CHRCNT
2518  010314  112137  010471                MOVB    (R1)+,SPACNT
2519  010320  013137  010472                MOV     @(R1)+,BINWRD
2520  010324  013704  010472          2$:   MOV     BINWRD,R4
2521  010330  113705  010470                MOVB    CHRCNT,R5
2522  010334  012700  012006                MOV     #TEMP,R0
2523  010340  010403                  3$:   MOV     R4,R3
2524  010342  042703  177770                BIC     #177770,R3
2525  010346  062703  000060                ADD     #060,R3
2526  010352  110320                        MOVB    R3,(R0)+
2527  010354  000241                        CLC
2528  010356  006004                        ROR     R4
2529  010360  000241                        CLC
2530  010362  006004                        ROR     R4
2531  010364  000241                        CLC
2532  010366  006004                        ROR     R4
2533  010370  005305                        DEC     R5
2534  010372  001362                        BNE     3$
2535  010374  012703  012050                MOV     #MDATA,R3
2536  010400  114023                  4$:   MOVB    -(R0),(R3)+
2537  010402  105337  010470                DECB    CHRCNT
2538  010406  001374                        BNE     4$
2539  010410  105737  010471                TSTB    SPACNT
2540  010414  001405                        BEQ     6$
2541  010416  112723  000040          5$:   MOVB    #040,(R3)+
2542  010422  105337  010471                DECB    SPACNT
2543  010426  001373                        BNE     5$
2544  010430  105013                  6$:   CLRB    (R3)
2545  010432  104402                        TYPE
2546  010434  012050                        MDATA
2547  010436  005337  010466                DEC     WRDCNT
2548  010442  001322                        BNE     1$
2549  010444  013737  001250  012006        MOV     TEMP3,TEMP
2550  010452  012605                        MOV     (SP)+,R5
```

GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```
                                        MOV     (SP)+,R4
                                        MOV     (SP)+,R3
                                        MOV     (SP)+,R1
                                        MOV     (SP)+,R0
                                        RTI
                                WRDCNT: 0
                                CHRCNT: 0
                                SPACNT=CHRCNT+1
                                BINWRD: 0
                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE

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

                                        ;ERROR HANDLER

                                .HLT:   CKSWR
010530  032777  010000  170442          BIT     #SW12,@SWR
010536  001406                          BEQ     XBX
010540  105777  170444                  TSTB    @TPCSR
010544  100003                          BPL     XBX
010546  112777  000207  170436          MOVB    #207,@TPDBR
010554  032777  020000  170416  XBX:    BIT     #SW13,@SWR
010562  001074                          BNE     HALTS
010564  021637  001234                  CMP     (SP),LSTERR
010570  001404                          BEQ     1$
010572  011637  001234                  MOV     (SP),LSTERR
010576  105037  001312                  CLRB    ERRFLG
010602  104406                  1$:     SAV05
010604  011605                          MOV     (SP),R5
010606  162705  000002                  SUB     #2,R5
010612  011504                          MOV     (R5),R4
010614  006304                          ASL     R4
010616  061504                          ADD     (R5),R4
010620  006304                          ASL     R4
010622  042704  177001                  BIC     #177001,R4
010626  062704  012760                  ADD     #.ERRTAB,R4
010632  012437  010724                  MOV     (R4)+,ERRMSG
010636  012437  010736                  MOV     (R4)+,DATAHD
010642  011437  010750                  MOV     (R4),DATABP
010646  105737  001312                  TSTB    ERRFLG
010652  001403                          BEQ     TYPMSG
010654  005737  010750                  TST     DATABP
010660  001027                          BNE     TYPDAT
010662  104402                  TYPMSG: TYPE
010664  011611                          MTSTN
010666  104411                          CNVRT
```

```
2607  010670  011050                          XTSTN
2608  010672  104402                          TYPE
2609  010674  011677                          MERRPC
2610  010676  104411                          CNVRT
2611  010700  011042                          ERTABO
2612  010702  104402                          TYPE
2613  010704  011332                          MCRLF
2614  010706  112737  177777  001312          MOVB    #-1,ERRFLG
2615  010714  005737  010724                  TST     ERRMSG
2616  010720  001402                          BEQ     WRKO.FM
2617  010722  104402                          TYPE
2618  010724  000000                  ERRMSG: 0
2619  010726                          WRKO.FM:
2620  010726  005737  010736                  TST     DATAHD
2621  010732  001402                          BEQ     TYPDAT
2622  010734  104402                          TYPE
2623  010736  000000                  DATAHD: 0
2624  010740  005737  010750          TYPDAT: TST     DATABP
2625  010744  001402                          BEQ     RESREG
2626  010746  104410                          CONVRT
2627  010750  000000                  DATABP: 0
2628  010752  104407                  RESREG: RESOS
2629  010754  005777  170220          HALTS:  TST     @SWR
2630  010760  100005                          BPL     EXITER
2631  010762  010046                          PUSHRO
2632  010764  016600  000002                  MOV     2(SP),RO
2633  010770  000000                          HALT
2634  010772  012600                          POPRO
2635  010774  104414                  EXITER: CKSWR
2636  010776  005237  001232                  INC     ERRCNT
2637  011002  032777  000400  170170          BIT     #SW08,@SWR
2638  011010  001007                          BNE     1$
2639  011012  032777  002000  170160          BIT     #SW10,@SWR
2640  011020  001407                          BEQ     2$
2641  011022  013737  001216  001214          MOV     NEXT,RETURN
2642  011030  012706  001200          1$:     MOV     #STACK,SP
2643  011034  000177  170154                  JMP     @RETURN
2644  011040  000002                  2$:     RTI
2645  011042  000001                  ERTABO: 1
2646  011044     006     002                  .BYTE   6,2
2647  011046  001274                          SAVPC
2648  011050  000001                  XTSTN:  1
2649  011052     003     002                  .BYTE   3,2
2650  011054  001226                          TSTNO
2651                                          ;ENTER HERE ON POWER FAILURE
2652
2653
2654  011056                          .PFAIL:
2655  011056  012737  011070  000024          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
2656  011064  000000                          HALT                            ;HALT ON POWER DOWN NORMAL
2657  011066  000777                          BR      .
2658
2659                                          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2660
2661  011070                          RESTAR:
2662  011070  012737  011056  000024          MOV     #.PFAIL,24              ;SET UP FOR POWER FAILURE
```

```
2663  011076  012706  001200                    MOV     #STACK,SP
2664  011102  005037  012006                    CLR     TEMP
2665  011106  005237  012006                    INC     TEMP
2666  011112  001375                            BNE     .-4
2667  011114  104402                            TYPE
2668  011116  011334                            MPFAIL
2669  011120  104411                            CNVRT
2670  011122  011144                            PFTAB
2671  011124  005037  001312                    CLR     ERRFLG
2672  011130  005037  001234                    CLR     LSTERR
2673  011134  104412                            MSTCLR
2674  011136  104413                            MEMCLR
2675  011140  000177  170050                    JMP     @RETURN
2676  011144  000001                  PFTAB:    1
2677  011146     003     002          .BYTE     3,2
2678  011150  001226                            TSTNO
2679
2680
2681                                            ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
2682                                            ;OF LOC.176.
2683                                            ;LOCATIONS USED:
2684  011152  000000                            RDSW:   .WORD   0
2685
2686
2687  011154  005737  000042          .CKSWR:   TST     @#42
2688  011160  001042                            BNE     OUT
2689  011162  022737  000176  001200            CMP     #SWREG,SWR      ;SOFTWARE SWITCH REGISTER PRESENT
2690  011170  001036                            BNE     OUT             ;NO, GET OUT
2691  011172  105777  170006                    TSTB    @TKCSR          ;YES,WAIT FOR
2692  011176  100033                            BPL     OUT             ;READY, GET CHARACTER
2693  011200  017737  170002  007612            MOV     @TKDBR,.MSG     ;AND STRIP OFF
2694  011206  042737  177600  007612            BIC     #177600,.MSG    ;THE GARBAGE
2695  011214  122737  000007  007612            CMPB    #7,.MSG         ;IS IT A <↑G>
2696  011222  001021                            BNE     OUT
2697  011224  104402  011302                    TYPE,$CNTG
2698  011230  005137  011152          .CNTLU:   COM     @#RDSW
2699  011234  104402  011306                    TYPE,$MSWR
2700  011240  104411  011274                    CNVRT,SWREGC
2701  011244  104403  011315                    INSTR,$MNEW
2702  011250  104405                            PARAM
2703  011252  000000                            0
2704  011254  177777                            177777
2705  011256  000176                            SWREG
2706  011260     000     001          .BYTE     0,1
2707  011262  104402  011332                    TYPE,MCRLF
2708  011266  005037  011152          OUT:      CLR     @#RDSW
2709  011272  000002                            RTI
2710  011274  000001                  SWREGC:   1
2711  011276     006     002          .BYTE     6,2
2712  011300  000176                            SWREG
2713  011302  057377  000107          $CNTG:    .ASCIZ  <377>/↑G/
2714  011306  051777  051127  020075  $MSWR:    .ASCIZ  <377>/SWR= /
2715  011314     000
2716  011315     040  047040  053505  $MNEW:    .ASCIZ  /  NEW= /
2717  011322  020075     000
2718          011326                    .EVEN
```

GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```
2719  011326  020040  000077          MQM:     .ASCIZ  /  ?/
2720  011332  000377                  MCRLF:   .ASCIZ  <377>
2721  011334  050377  051127  043040  MPFAIL:  .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
2722  011342  044501  042514  027104
2723  011350  051040  051505  040524
2724  011356  072122  040440  020124
2725  011364  042524  052123  000040
2726  011372  042777  042116  050040  MEPASS:  .ASCIZ  <377>/END PASS DZDQ8  /
2727  011400  051501  020123  055104
2728  011406  050504  020102  000040
2729  011414  051377     000          MR:      .ASCIZ  <377>/R/
2730  011417     377  051120  043517  MERR2:   .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
2731  011424  040522  020115  047111
2732  011432  044504  040503  042524
2733  011440  020123  047516  042040
2734  011446  053105  041511  051505
2735  011454  050040  042522  042523
2736  011462  052116  000056
2737  011466  044777  051516  043125  MERR3:   .ASCIZ  <377>/INSUFFICIENT DATA!/
2738  011474  044506  044503  047105
2739  011502  020124  040504  040524
2740  011510  000041
2741  011512  052377  051505  020124  MTSTPC:  .ASCIZ  <377>/TEST PC-/
2742  011520  041520  000055
2743  011524  046377  041517  020113  MLOCK:   .ASCIZ  <377>/LOCK ON SELECTED TEST/
2744  011532  047117  051440  046105
2745  011540  041505  042524  020104
2746  011546  042524  052123     000
2747  011553     103  051123  020072  MCSRX:   .ASCIZ   /CSR: /
2748  011560     000
2749  011561     126  041505  020072  MVECX:   .ASCIZ  /VEC: /
2750  011566     000
2751  011567     120  051501  042523  MPASSX:  .ASCIZ  /PASSES: /
2752  011574  035123  000040
2753  011600  051105  047522  051522  MERRX:   .ASCIZ  /ERRORS: /
2754  011606  020072     000
2755  011611     377  052377  051505  MTSTN:   .ASCIZ  <377><377> /TEST NO: /
2756  011616  020124  047516  020072
2757  011624     000
2758  011625     377  042523  020124  MNEW:    .ASCIZ  <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./
2759  011632  053523  052111  044103
2760  011640  051040  043505  052040
2761  011646  020117  050504  030461
2762  011654  051447  042040  051505
2763  011662  051111  042105  040440
2764  011670  052103  053111  027105
2765  011676     000
2766  011677     120  035103  000040  MERRPC:  .ASCIZ  /PC: /
2767  011704  046777  050101  047440  XHEAD:   .ASCIZ  <377>/MAP OF DQ11 STATUS/<377>
2768  011712  020106  050504  030461
2769  011720  051440  040524  052524
2770  011726  177523     000
2771          011732                  .EVEN
2772  011732  000002                  XSTATQ:  2
2773  011734     006     003                   .BYTE   6,3
2774  011736  001244                          TEMP1
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 56
DZDQBC.P11    16-DEC-76 13:23          GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

  2775  011740      006      002                 .BYTE   6,2
  2776  011742  001246                          TEMP2
  2777                                .EVEN
  2778
  2779                                          ;BUFFERS FOR INPUT-OUTPUT
  2780
  2781  011744  000000                INBUF:  0
  2782          012006                .=.+40
  2783  012006  000000                TEMP:   0
  2784          012050                .=.+40
  2785  012050  000000                MDATA:  0
  2786          012112                .=.+40
```

```
2787
2788  012112  000002                  .MEMCLR:        RTI
2789  012114  000002                  .MSTCLR:        RTI
2790                                                  ;TABLE OF ERROR MESSAGES AND DATA HEADERS
2791
2792  012116  042515  047515  054522  EM0:    .ASCIZ  /MEMORY EXTENSION READ-WRITE ERROR/
      012160  042515  047515  054522  EM1:    .ASCIZ  /MEMORY EXTENSION ADDRESS ERROR/
      012217     102  051525  040440  EM2:    .ASCIZ  /BUS ADR-CHAR COUNT MEMORY DATA ERROR/
      012264  042515  047515  054522  EM3:    .ASCIZ  /MEMORY EXTENSION DATA ERROR/
      012320  044103  051101  041501  EM4:    .ASCIZ  /CHARACTER MEMORY ADDRESS ERROR/
      012357     123  050505  042525  EM5:    .ASCIZ  /SEQUENCE MEMORY ADDRESS ERROR/
      012415     103  040510  040522  EM6:    .ASCIZ  /CHARACTER MEMORY DATA ERROR/
      012451     123  050505  042525  EM7:    .ASCIZ  /SEQUENCE MEMORY DATA ERROR/
      012504  051120  046511  051101  EM10:   .ASCIZ  /PRIMARY REGISTER MASTER CLEAR ERROR/
      012550  042523  047503  042116  EM11:   .ASCIZ  /SECONDARY REGISTER MASTER CLEAR ERROR/
      012616  051127  052111  020105  EM12:   .ASCIZ  /WRITE ENABLE NOT CLEARED/

      012647     377  054105  042520  DH0:    .ASCIZ  <377>/EXPECTED  RECEIVED  SEC ADR   SEC REG/
      012716  042777  050130  041505  DH1:    .ASCIZ  <377>/EXPECTED  RECEIVED  REG ADDRESS/
              012760                  .EVEN

2793                                                  ;TABLE OF ERROR POINTERS
2794
2795  012760  012116                  .ERRTAB:EM0
2796  012762  012716                          DH1
2797  012764  013126                          DT2
2798  012766  012160                          EM1
2799  012770  012716                          DH1
2800  012772  013126                          DT2
2801  012774  012217                          EM2
2802  012776  012647                          DH0
2803  013000  013104                          DT1
2804  013002  012264                          EM3
2805  013004  012647                          DH0
2806  013006  013062                          DT0
2807  013010  012320                          EM4
2808  013012  012647                          DH0
2809  013014  013104                          DT1
2810  013016  012357                          EM5
2811  013020  012647                          DH0
2812  013022  013104                          DT1
2813  013024  012415                          EM6
2814  013026  012647                          DH0
2815  013030  013104                          DT1
2816  013032  012451                          EM7
2817  013034  012647                          DH0
2818  013036  013104                          DT1
2819  013040  012504                          EM10
2820  013042  012716                          DH1
2821  013044  013126                          DT2
2822  013046  012550                          EM11
2823  013050  012647                          DH0
2824  013052  013104                          DT1
2825  013054  012616                          EM12
2826  013056  000000                          0
2827  013060  000000                          0
```

```
                                                ;TABLE OF DATA POINTER FOR ERROR TYPEOUT

013062  000004                        DT0:    4
013064      003     007               .BYTE   3,7
013066  001270                              SAVR5
013070      003     007               .BYTE   3,7
013072  001266                              SAVR4
013074      006     004               .BYTE   6,4
013076  001372                              DQSEC
013100      002     000               .BYTE   2,0
013102  001264                              SAVR3
013104  000004                        DT1:    4
013106      006     004               .BYTE   6,4
013110  001270                              SAVR5
013112      006     004               .BYTE   6,4
013114  001266                              SAVR4
013116      006     004               .BYTE   6,4
013120  001372                              DQSEC
013122      002     000               .BYTE   2,0
013124  001262                              SAVR2
013126  000003                        DT2:    3
013130      006     004               .BYTE   6,4
013132  001270                              SAVR5
013134      006     004               .BYTE   6,4
013136  001266                              SAVR4
013140      006     004               .BYTE   6,4
013142  001264                              SAVR3
        000001                        .END
```

```
ABBIT = 002000            613#   983    2243
ACTBIT= 004000            614#   991
ADRCNT= 010153           2430*  2469*  2478#
BABIT = 010000            615#   987    2104
BBBIT = 020000            616#   978    1705
BINWRD  010472           2519*  2520   2559#
BIT0  = 000001            608#  1306
BIT1  = 000002            607#
BIT10 = 002000            598#
BIT11 = 004000            597#
BIT12 = 010000            596#
BIT13 = 020000            595#
BIT14 = 040000            594#  2330
BIT15 = 100000            593#
BIT2  = 000004            606#  1277
BIT3  = 000010            605#
BIT4  = 000020            604#  1364   1368   1413   1430   1431   1603   1604   1641   1642   1679   1680
BIT5  = 000040            603#  2069   2092   2122   2152   2177   2204   2230   2262
BIT6  = 000100            602#
BIT7  = 000200            601#
BIT8  = 000400            600#
BIT9  = 001000            599#
BRW     007436           1283   2348#
BRX     007440           1284   2349#
CHARDT= 000010             633#
CHAR1   001236            1064#
CHAR2   001240            1065#
CHAR3   001242            1066#
CHKAB1  007036            2222   2243#
CHKBA1  006356            2086   2104#
CHKBA2  006460            2116   2132#
CHKLB1  004372            1671   1705#
CHRCNT  010470            2517*  2521   2537*  2557#  2558
CKSWR = 104414            1125#  1251   1261   1276   2311   2329   2353   2576   2635
CNTLU = 104415            1127#  1233
CNVRT = 104411            1119#  2285   2289   2293   2297   2606   2610   2669   2700
CONVRT= 104410            1117#  1241   2626
CREAM   001300            1081#  1205*  1315*  1319*  1328   1329*  1330   1331*
CS.MAP  000220             958#  1236
DATABP  010750            2599*  2602   2624   2627#
DATAHD  010736            2598*  2620   2623#
DEVADR  010150            2428*  2466   2476#
DH0     012647            2792#  2802   2805   2808   2811   2814   2817   2823
DH1     012716            2792#  2796   2799   2820
DISPRE  000174             949#  1226
DLIGHT= 177570             578#  1217
DQACTV  001500             929   1003*  1009*  1010*  1014   1184#  1258*  1259   1309   1311
DQCR00  001400            1152#
DQCR01  001404            1154#
DQCR02  001410            1156#
DQCR03  001414            1158#
DQCR04  001420            1160#
DQCR05  001424            1162#
DQCR06  001430            1164#
DQCR07  001434            1166#
DQCR10  001440            1168#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
DQCR11  001444            1170#
DQCR12  001450            1172#
DQCR13  001454            1174#
DQCR14  001460            1176#
DQCR15  001464            1178#
DQCR16  001470            1180#
DQCR17  001474            1182#
DQCSR   001506            1187#  1328*  1332
DQFRR   001366            1142#  1345*  1346*  1347   2173   2175*  2180
DQNUM   001504             962*   999*  1004   1006   1186#  1200   2302
DQRCSH  001362            1140#  1341*  1342*  1724*  1734*  1761*  1771*  1798*  1806*  1832*  1840*  1866*  1874*
                          1900*  1908*  1934*  1942*  1968*  1976*  2002*  2010*  2036*  2044*
DQRCSR  001360            1139#  1332*  1341   1343   2065   2067*  2072   2088   2090*  2095   2316
DQREG   001370            1143#  1347*  1348*  1349   1364*  1368   1383   1384*  1385   1389*  1390   1407   1408*
                          1409   1414*  1416   1431*  1432*  1433*  1442*  1443*  1444   1471*  1478*  1504*  1511*
                          1537*  1544*  1570*  1577*  1604*  1605*  1606*  1613*  1614*  1615   1642*  1643*  1644*
                          1651*  1652*  1653   1680*  1681*  1682*  1689*  1690*  1691   1723*  1733*  1760*  1770*
                          1797*  1805*  1831*  1839*  1865*  1873*  1899*  1907*  1933*  1941*  1967*  1975*  2001*
                          2009*  2035*  2043*  2068*  2091*  2121*  2151*  2176*  2200*  2203*  2205*  2226*  2229*
                          2231*  2258*  2261*  2263*
DQRLVL  001352            1136#  1335*  1336*  1337
DQRVEC  001350            1135#  1333*  1334*  1335   2319
DQSEC   001372            1144#  1349*  1350*  1351   1367*  1415*  1434*  1468   1472*  1479   1501   1505*  1512
                          1534   1538*  1545   1567   1571*  1578   1607*  1645*  1683*  1725*  1735   1762*  1772
                          1799*  1807   1833*  1841   1867*  1875   1901*  1909   1935*  1943   1969*  1977   2003*
                          2011   2037*  2045   2069*  2092*  2122*  2152*  2177*  2202*  2204*  2208   2228*  2230*
                          2234   2260*  2262*  2266   2837   2846
DQSECH  001374            1145#  1351*  1352*
DQSTAT  001510            1188#  1330*  1333   1705   2104   2243
DQST00  001402            1153#
DQST01  001406            1155#
DQST02  001412            1157#
DQST03  001416            1159#
DQST04  001422            1161#
DQST05  001426            1163#
DQST06  001432            1165#
DQST07  001436            1167#
DQST10  001442            1169#
DQST11  001446            1171#
DQST12  001452            1173#
DQST13  001456            1175#
DQST14  001462            1177#
DQST15  001466            1179#
DQST16  001472            1181#
DQST17  001476            1183#
DQTCSR  001364            1141#  1343*  1344*  1345   2118   2120*  2125   2148   2150*  2155
DQTLVL  001356            1138#  1339*  1340*
DQTVEC  001354            1137#  1337*  1338*  1339
DSWR  = 177570             577#  1216
DT0     013062            2806   2831#
DT1     013104            2803   2809   2812   2815   2818   2824   2840#
DT2     013126            2797   2800   2821   2849#
EM0     012116            2792#  2795
EM1     012160            2792#  2798
EM10    012504            2792#  2819
EM11    012550            2792#  2822
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM12    012616          2792#   2825
EM2     012217          2792#   2801
EM3     012264          2792#   2804
EM4     012320          2792#   2807
EM5     012357          2792#   2810
EM6     012415          2792#   2813
EM7     012451          2792#   2816
ERRCNT  001232          1059#   1206*   2325    2636*
ERRFLG  001312          1090#   1203*   2279*   2342*   2587*   2600    2614*   2671*
ERRMSG  010724          2597*   2615    2618*
ERTABO  011042          2611    2645#
EXITER  010774          2630    2635#
HALTS   010754          2583    2629#
HILIM   010146          2427*   2457    2475#
ICOUNT  001222          1055#   2340    2344*
INBUF   011744          2391    2433    2781#
INIFLG  001310          1088#   1211    1214*
INSTER= 104404          1109#   2452
INSTR = 104403          1107#   1287    2701
INSTR2  007732          2402    2417#
JUMBIT= 040000           617#    990
LIGHTS  001202          1044#   1217*   1226*   2299*
LIMITS  010074          2443    2457#
LOBITS  010152          2429*   2461    2477#   2478
LOCK    001220          1054#   1466*   1499*   1532*   1565*   1596*   1634*   1672*   1718*   1755*   1792*   1826*   1860*
                        1894*   1928*   1962*   1996*   2030*   2356
LOGICA  007254           911    2306#
LOKFLG  001313          1091#
LOLIM   010144          2426*   2459    2474#
LPCNT   001224          1056#   2339*   2340    2343*
LSTERR  001234          1060#   1207*   2278*   2584    2586*   2672*
MCRLF   011332          2399    2415    2507    2613    2707    2720#
MCSRX   011553          2284    2747#
MDATA   012050          2535    2546    2785#
MEMCLR= 104413          1123#   2674
MEPASS  011372          2282    2726#
MERRPC  011677          2609    2766#
MERRX   011600          2296    2753#
MERR2   011417          1021    1323    2730#
MERR3   011466          1255    2737#
MISC. = 000012           635#
MLOCK   011524          1279    2743#
MNEW    011625          1248    2758#
MPHSSX  011567          2292    2751#
MPFAIL  011334          2668    2721#
MQM     011326          2412    2719#
MR      011414          1297    2729#
MSTCLR= 104412          1121#   2673
MTITLE  001000          1030#   1213
MTSTN   011611          2605    2755#
MTSTPC  011512          1288    2741#
MVECX   011561          2288    2749#
NEXT    001216          1053#   1303*   1363*   1382*   1406*   1465*   1498*   1531*   1564*   1595*   1633*   1671*   1717*
                        1754*   1791*   1825*   1859*   1893*   1927*   1961*   1995*   2029*   2063*   2086*   2116*   2146*
                        2171*   2196*   2222*   2254*   2345    2641
OOOBIT= 001000           618#    992
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
OPT2X   002644           1359#
OUT     011266           2688    2690    2692    2696    2708#
PAPAM = 104405           1111#   1289    2702
PS: M1  010000           2432#   2453
PS:ERR  010054           2435    2437    2439    2448#   2458    2460    2462
PARTI   010136           2451    2471#
PRSCNT  001230           1058#   1202*   2280*   2299    2322
PFTAB   011144           2670    2676#
POLY. = 000017            640#
POPRO = 012600            587#   2634
POP1SP= 005726            585#
POP2SP= 022626            589#
PS    = 177776            579#    932*   1197*   1272*   2064*   2087*   2117*   2147*   2172*   2197*   2223*   2255*
PUSHRO= 010046            586#   2631
PUSH1S= 005746            584#
PUSH2S= 024646            588#
RDSW    011152           2364    2413    2450    2684#   2698*   2708*
RESREG  010752           2625    2628#
RESTAR  011070           2655    2661#
RESTRT  007264           2301    2304    2311#
RESOS = 104407           1115#   2628
RETURN  001214           1052#   1209*   1292    1296*   1298    1302*   2312*   2345*   2346    2641*   2643    2675
RUN     001304           1083#   1306*   1309    1314*   1320*   1327*
RUNCNT  001306           1084#   1307*   1316*   1318*
RUNFLG  001302           1082#   1204*   1304    1308*
RXBA.P= 000000            624#
RXBA.S= 000004            628#
RXWC.P= 000001            625#
RXWC.S= 000005            629#
RX.BCC= 000015            638#
SAVACT  001502           1014*   1185#   1252
SAVNUM  001276           1007*   1080#   1200*   2300*   2302*
SAVPC   001274           1079#   2482*   2647
SAVR0   001256           1072#   2491*   2496
SAVR1   001260           1073#   2490*   2497
SAVR2   001262           1074#   2489*   2498    2848
SAVR3   001264           1075#   2488*   2499    2839    2855
SAVR4   001266           1076#   2487*   2500    2835    2844    2853
SAVR5   001270           1077#   2486*   2501    2833    2842    2851
SAVSP   001272           1078#
SAVOS = 104406           1113#   2588
SCOPE = 104400           1101#   1371    1395    1422    1454    1487    1520    1553    1586    1624    1662    1700    1744
                         1781    1815    1849    1883    1917    1951    1985    2019    2053    2076    2099    2131    2161
                         2186    2212    2238    2270
SCOP1 = 104401           1103#   1449    1483    1516    1549    1582    1620    1658    1696    1739    1776    1811    1845
                         1879    1913    1947    1981    2015    2049
SEQ.  = 000014            637#
SPACNT= 010471           2518*   2539    2542*   2558#
STACK = 001200            580#   1198    1273    2642    2663
STFLG   001311           1089#   1201*
SVOS    010162           2486#
SWR     001200           1043#   1216*   1221    1225*   1231    1234    1245    1252    1258    1277    1285    2330    2337
                         2354    2366    2577    2582    2629    2637    2639    2689
SWREG   000176            950#   1225    1231    2689    2705    2712
SWREGC  011274           2700    2710#
SW00  = 000001            560#   1245
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
SW01  = 000002            559#   1285
SW02  = 000004            558#
SW03  = 000010            557#
SW04  = 000020            556#
SW05  = 000040            555#
SW06  = 000100            554#
SW08  = 000400            553#   2637
SW09  = 001000            552#   2354
SW10  = 002000            551#   2639
SW11  = 004000            550#   2337
SW12  = 010000            549#   2366    2577
SW13  = 020000            548#   2582
SW14  = 040000            547#
SW15  = 100000            546#
SYNBIT= 100000            619#    972
SYNC. = 000011            634#
TEMP    012006           2514    2522    2549*   2664*   2665*   2783#
TEMP1   001244            929#    930#   1067#   1238#   1239    1243#   2774
TEMP2   001246           1068#   1239#   2776
TEMP3   001250           1069#   2514#   2549
TEMP4   001252           1070#
TEMP5   001254           1071#
TKCSR   001204           1045#   2333    2393    2691
TKDBR   001206           1046#   2335    2395    2403    2693
TLAST = 007046           1291    2272#
TPCSR   001210           1047#   2370    2373    2376    2404    2579
TPDBR   001212           1048#   2372#   2375#   2378#   2403#   2581#
TKPOK   010506           2568#
TSTNO   001226           1057#   1208#   1301#   1362#   1381#   1405#   1464#   1497#   1530#   1563#   1594#   1632#   1670#
                         1716#   1753#   1790#   1824#   1858#   1892#   1926#   1960#   1994#   2028#   2062#   2085#   2115#
                         2145#   2170#   2195#   2221#   2253#   2650    2678
TST1    002254           1290    1296    1301#   2312    2313
TST10   003566           1531    1563#
TST11   003702           1564    1594#
TST12   004052           1595    1632#
TST13   004222           1633    1670#
TST14   004406           1706    1716#
TST15   004546           1717    1753#
TST16   004706           1754    1790#
TST17   005036           1791    1824#
TST2    002[illegible]   1302    1303    1362#
TST20   005166           1825    1858#
TST21   005316           1859    1892#
TST22   005446           1893    1926#
TST23   005576           1927    1960#
TST24   005726           1961    1994#
TST25   006056           1995    2028#
TST26   006206           1707    2029    2062#
TST27   006272           2063    2085#
TST3    002706           1363    1381#
TST30   006366           2115#
TST31   006462           2105    2145#
TST32   006554           2132    2146    2170#
TST33   006646           2171    2195#
TST34   006742           2196    2221#
TST35   007046           2253#   2272
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
TST36 = ****** U          2244    2254
TST4    002774            1382    1405#
TST5    003222            1406    1464#
TST6    003336            1465    1497#
TST7    003452            1498    1530#
TTST    007340            1280*   1281*   1283*   1284*   2331#
TXBA.P= 000002             626#
TXBA.S= 000006             630#
TXWC.P= 000003             627#
TXWC.S= 000007             631#
TX.BCC= 000016             639#
TX.MUX= 000013             636#
TYPDAT  010740            2603    2621    2624#
TYPE  = 104402            1020    1105#   1213    1237    1247    1254    1279    1297    1322    2281    2283    2287    2291
                          2295    2389    2399    2411    2415    2506    2545    2604    2608    2612    2617    2622    2667
                          2697    2699    2707
TYPMSG  010662            2601    2604#
VECMAP  000056             922#   1019
WRDCNT  010466            2516*   2547*   2556#
WRKO.F  010726            2616    2619#
XBX     010554            2578    2580    2582#
XCSR    007300            2286    2314#
XERR    007322            2298    2323#
XHEAD   011704            1237    2767#
XPASS   007314            2294    2320#
XSTATQ  011732            1242    2772#
XTSTN   011050            2607    2648#
XVEC    007306            2290    2317#
$CNTG   011302            2697    2713#
$E    = 000037               1#   1303    1304#   1363    1364#   1382    1383#   1406    1407#   1465    1467#   1498    1500#
                          1531    1533#   1564    1566#   1595    1597#   1633    1635#   1673#   1717    1719#   1754    1756#
                          1791    1793#   1825    1827#   1859    1861#   1893    1895#   1927    1929#   1961    1963#   1995
                          1997#   2029    2031#   2063    2064#   2087#   2117#   2146    2147#   2171    2172#   2196    2197#
                          2223#   2254    2255#
$MNEW   011315            2701    2716#
$MSWR   011306            2699    2714#
$N    = 000035               1#   1299    1304#   1360    1364#   1379    1393#   1403    1407#   1462    1467#   1495    1500#
                          1528    1533#   1561    1566#   1592    1597#   1630    1635#   1668    1673#   1701    1714    1719#
                          1751    1756#   1788    1793#   1822    1827#   1856    1861#   1890    1895#   1924    1929#   1958
                          1963#   1992    1997#   2026    2031#   2060    2064#   2083    2087#   2100    2113    2117#   2132
                          2133    2143    2147#   2168    2172#   2193    2197#   2219    2223#   2239    2251    2255#   2272#
$N1   = 000035            1701#   2100#   2132#   2133#   2239#
$N2   = 000036            1701#   2100#   2239#
$N3   = 000026            1701#
$Y    = 000016               1#   1092#   1101    1103#   1105#   1107#   1109#   1111#   1113#   1115#   1117#   1119#   1121#
                          1123#   1125#   1127#   1129#
.     = 013144             644#    645     647     649     651     653     655     657     659     661     663     665     667
                           669     671     673     675     677     679     681     683     685     687     689     691     693
                           695     697     699     701     703     705     707     709     711     713     715     717     719
                           721     723     725     727     729     731     733     735     737     739     741     743     745
                           747     749     751     753     755     757     759     761     763     765     767     769     771
                           773     775     777     779     781     783     785     787     789     791     793     795     797
                           799     801     803     805     807     809     811     813     815     817     819     821     823
                           825     827     829     831     833     835     837     839     841     843     845     847     849
                           851     853     855     857     859     861     863     865     867     869     871     873     875
                           877     879     881     883     885     887     889     891     893     895     897     899     903#
```

DZDQBC.P11 16-DEC-76 13:23 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        910#   912#   920#   936    948#   954#   957#   961    977    982    986    989    1024
                       1029#  1040#  1081#  1151#  1152#  1153#  1154#  1155#  1156#  1157#  1158#  1159#  1160#
                       1161#  1162#  1163#  1164#  1165#  1166#  1167#  1168#  1169#  1170#  1171#  1172#  1173#
                       1174#  1175#  1176#  1177#  1178#  1179#  1180#  1181#  1182#  1183#  1184#  1185#  1186#
                       1187#  1188#  1235   1240   1244   1257   1325   2371   2374   2657   2666   2718#  2771#
                       2782#  2784#  2786#  2792#
.BEGIN  002116         1272#
.CKSWR  011154         1126   2687#
.CNTLU  011230         1128   2698#
.CNVRT  010252         1120   2508#
.CONVR  010246         1118   2506#
.EOP    007142         2244   2254   2278#
.ERRTA  012760         2596   2795#
.HLT    010526          906   2576#
.INSTE  007706         1110   2409#
.INSTG  007712         2408   2411#
.INSTR  007570         1108   2385#
.INSTI  007610         2389#  2400   2416
.MEMCL  012112         1124   2788#
.MSG    007612         2387*  2390#  2693*  2694*  2695
.MSTCL  012114         1122   2789#
.PARAM  007740         1112   2423#
.PFAIL  011056          904   1199   2654#  2662
.RESOS  010214         1116   2496#
.SAVOS  010154         1114   2482#
.SCOPE  007330         1102   2329#
.SCOPI  007442         1104   2353#
.START  001512          955   1197#  1209
.TRPSR  010474          908   2565#
.TRPTA  001314         1100#  2570
.TYPE   007462         1106   2361#
```

DZDQBC.P11 16-DEC-76 13:23 CROSS REFERENCE TABLE -- MACRO NAMES

```
ADRTSX   1353#   1708    1745
ADRTS1   1353#   1423
COMMEN   1701#   2100#   2132#   2133#   2239#
DQEND       1#   2271
DQFRNT      1#    526
HLT       590#   1370    1394    1421    1448    1482    1515    1548    1581    1619    1657    1695    1738    1775    1810
         1844    1878    1912    1946    1980    2014    2048    2075    2098    2130    2160    2185    2211    2237    2269
IDENT       1#
ORANGE      1#   1299
REGTS1   1353#   1455    1488    1521    1554
REGTS2   1353#   1587    1625    1663
REGTS3   1353#   1782    1816    1850    1884    1918    1952    1986    2020
REGTS4   1353#   2054    2077    2107    2137    2162    2187    2213    2245
TESTA1      1#
TESTB1      1#     50
TESTC1      1#
TESTD1      1#
TESTE1      1#
TESTF1      1#
TESTH1      1#
TESTH2      1#
$BEGIN      1#   1269
$BUFFE      1#   2778
$CATCH      1#    643
$CLRVE      1#   1234
$CONVR      1#   2503
$EOP        1#   2271
$GETFL      1#
$GETPA      1#   1287
$HEADE      1#    526
$HLT        1#   2573
$INSTR      1#   2382
$INTMP      1#
$MAINT      1#
$MSG        1#   2719
$PARAM      1#   2420
$PFAIL      1#   2651
$REG        1#   2479
$SCOPE      1#   2326
$SCOP1      1#   2350
$SETFL      1#
$SETVE      1#    901
$START      1#   1189
$SYMBO      1#    543
$TRAPS      1#   1092
$TRPDE      1#   1101    1103    1105    1107    1109    1111    1113    1115    1117    1119    1121    1123    1125    1127
$TRPSR      1#   2560
$TSTN       1#   1299    1360    1379    1403    1462    1495    1528    1561    1592    1630    1668    1714    1751    1788
          1822   1856    1890    1924    1958    1992    2026    2060    2083    2113    2143    2168    2193    2219    2251
$TYPE       1#   2358
$VARIA      1#   1028

. ABS.  013144     000
```

```
DZDQB   MACY11 27(1006)  22-DEC-76  11:05  PAGE 69
DZDQBC.P11    16-DEC-76 13:23          CROSS REFERENCE TABLE -- MACRO NAMES

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 MULE:DZDQBC.BIN,MULE:DZDQBC.SEQ/SOL/CRF=DSKZ:UNIV.P11,DSKZ:DZDQBC.P11
 RUN-TIME:  20 30 2 SECONDS
 RUN-TIME RATIO: 200/53=3.7
 CORE USED:  21K  (41 PAGES)
```