

This section contains a grid of 60 small diagrams, organized into 10 rows and 6 columns. Each diagram appears to be a logic test or waveform capture. The diagrams in the first two columns are primarily vertical waveforms, likely representing digital signals over time. The diagrams in the remaining four columns are more complex, featuring various patterns of lines, dots, and text, which could represent truth tables, timing diagrams, or specific test procedures. The text within these diagrams is too small to read clearly, but they appear to be related to the 'BASIC LOGIC TEST PART 2' mentioned in the header.

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

B01

EOF1DZDQACSEQ

00010000

770325

PDP10 411

LEHDR1DZDQBCSEQ

00010000

770325

CO1

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 2
DZDQBC.P11 16-DEC-76 13:23

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDQB-C-D
PRODUCT NAME: BASIC LOGIC TEST PART 2
DATE: MARCH 1977
MAINTAINER: DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1974,1977 BY DIGITAL EQUIPMENT CORPORATION

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. DZDGA [REV] BASIC R/W TEST #1
2. DZDGB [REV] BASIC R/W TEST #2
3. DZDGC [REV] BASIC NPR AND INTERRUPT TEST
4. DZDGD [REV] RECEIVER TRANSMITTER EXERCISER TEST
5. DZDGE [REV] MISC. RX AND TX TESTS. PLUS BCC TESTS.
6. DZDGF [REV] CHARACTER DETECT TESTS.
7. DZDGH [REV] CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.

1. DZDGO [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 EQUIVALENT)
 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.

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 4
 DZDQBC.P11 16-DEC-76 13:23

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:

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 5
 DZDQBC.P11 16-DEC-76 13:23

SMR=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: SMR=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 "SCOPI" 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 "SCOPI")
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
 POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST)
 TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE
 PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION
 CONCERNING THE ERROR REPORT; LOOK IN THE LISTING
 FOR THAT TEST NUMBER WHICH WAS TYPED OUT
 AND THEN NOTE THE PC OF THE ERROR REPORT
 THIS WAY THE EXACT FUNCTIONING OF THE TEST
 CAN BE INTERPEDITED

6. ERRORS

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

INFORMATION WILL BE SUPPLIED 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 PERFORMED.
TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW
BEING PERFORMED.
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
SEQUENTIALLY. 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 OCCURS 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 15 IN APPRIQATE 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 EXISTS

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?

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

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 APPRIORATE 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

525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561

.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 <IG>.

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

562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617

;REGISTER DEFINITIONS

000000	RO=%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	PUSHRO=10046	;SAVE RO ON STACK
012600	POPPO=12600	;RESTORE RO FROM STACK
024646	PUSH2SP=24646	;DECREMENT STACK TWICE
022626	POP2SP=22626	;INCREMENT STACK TWICE
	.EQUIV ENT,HLT	;BASIC DEFINITION OF ERROR CALL

BIT15=100000
 BIT14=40000
 BIT13=20000
 BIT12=10000
 BIT11=4000
 BIT10=2000
 BIT9=1000
 BIT8=400
 BIT7=200
 BIT6=100
 BIT5=40
 BIT4=20
 BIT3=10
 BIT2=4
 BIT1=2
 BIT0=1

;D011 OPTIONAL DEFINITIONS

002000	ABBIT=2000
004000	ACTBIT=4000
010000	BABIT=10000
020000	BBBIT=20000
040000	JUMBIT=40000

618 001000
 619 100000
 620
 621
 622
 623
 624 000000
 625 000001
 626 000002
 627 000003
 628 000004
 629 000005
 630 000006
 631 000007
 632
 633 000010
 634 000011
 635 000012
 636 000013
 637 000014
 638 000015
 639 000016
 640 000017
 641
 642

000BIT=1000
 SYNBIT=100000

;DQ11 SECONDARY REGISTER DEFINATIONS

RXBA.P=0	;RECEIVER BUS ADDRESS PRIMARY.
RXWC.P=1	;RECEIVER WORD COUNT PRIMARY.
TXBA.P=2	;TRANSMITTER BUS ADDRESS PRIMARY.
TXWC.P=3	;TRANSMITTER BUS ADDRESS PRIMARY.
RXBA.S=4	;RECEIVER BUS ADDRESS SECONDARY.
RXWC.S=5	;RECEIVER WORD COUNT SECONDARY.
TXBA.S=6	;TRANSMITTER BUS ADDRESS SECONDARY.
TXWC.S=7	;TRANSMITTER WORD COUNT SECONDARY.
CHARDT=10	;CHARACTER DETECT REGISTER.
SYNC.=11	;SYNC REGISTER.
MISC.=12	;MISCELLANEOUS REGISTER.
TX.MUX=13	;TRANSMITTER MUX REGISTER.
SEQ.=14	;SEQUENCE REGISTER.
RX.BCC=15	;RECEIVER BCC REGISTER.
TX.BCC=16	;TRANSMITTER BCC REGISTER.
POLY.=17	;POLYNOMIAL REGISTER.

```

643          ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
644          .:=0
645 000000 000002      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
646 000002 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
647 000004 000006      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
648 000006 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
649 000010 000012      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
650 000012 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
651 000014 000016      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
652 000016 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
653 000020 000022      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
654 000022 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
655 000024 000026      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
656 000026 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
657 000030 000032      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
658 000032 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
659 000034 000036      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
660 000036 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
661 000040 000042      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
662 000042 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
663 000044 000046      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
664 000046 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
665 000050 000052      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
666 000052 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
667 000054 000056      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
668 000056 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
669 000060 000062      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
670 000062 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
671 000064 000066      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
672 000066 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
673 000070 000072      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
674 000072 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
675 000074 000076      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
676 000076 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
677 000100 000102      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
678 000102 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
679 000104 000106      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
680 000106 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
681 000110 000112      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
682 000112 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
683 000114 000116      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
684 000116 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
685 000120 000122      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
686 000122 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
687 000124 000126      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
688 000126 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
689 000130 000132      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
690 000132 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
691 000134 000136      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
692 000136 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
693 000140 000142      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
694 000142 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
695 000144 000146      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
696 000146 000000      HALT      ;EXAMINE STACK TO FIND CAUSE
697 000150 000152      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
698 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

755	000334	000336	.+2	:UNEXPECTED TRAP TO THIS LOCATION
756	000336	000000	HALT	:EXAMINE STACK TO FIND CAUSE
757	000340	000342	.+2	:UNEXPECTED TRAP TO THIS LOCATION
758	000342	000000	HALT	:EXAMINE STACK TO FIND CAUSE
759	000344	000346	.+2	:UNEXPECTED TRAP TO THIS LOCATION
760	000346	000000	HALT	:EXAMINE STACK TO FIND CAUSE
761	000350	000352	.+2	:UNEXPECTED TRAP TO THIS LOCATION
762	000352	000000	HALT	:EXAMINE STACK TO FIND CAUSE
763	000354	000356	.+2	:UNEXPECTED TRAP TO THIS LOCATION
764	000356	000000	HALT	:EXAMINE STACK TO FIND CAUSE
765	000360	000362	.+2	:UNEXPECTED TRAP TO THIS LOCATION
766	000362	000000	HALT	:EXAMINE STACK TO FIND CAUSE
767	000364	000366	.+2	:UNEXPECTED TRAP TO THIS LOCATION
768	000366	000000	HALT	:EXAMINE STACK TO FIND CAUSE
769	000370	000372	.+2	:UNEXPECTED TRAP TO THIS LOCATION
770	000372	000000	HALT	:EXAMINE STACK TO FIND CAUSE
771	000374	000376	.+2	:UNEXPECTED TRAP TO THIS LOCATION
772	000376	000000	HALT	:EXAMINE STACK TO FIND CAUSE
773	000400	000402	.+2	:UNEXPECTED TRAP TO THIS LOCATION
774	000402	000000	HALT	:EXAMINE STACK TO FIND CAUSE
775	000404	000406	.+2	:UNEXPECTED TRAP TO THIS LOCATION
776	000406	000000	HALT	:EXAMINE STACK TO FIND CAUSE
777	000410	000412	.+2	:UNEXPECTED TRAP TO THIS LOCATION
778	000412	000000	HALT	:EXAMINE STACK TO FIND CAUSE
779	000414	000416	.+2	:UNEXPECTED TRAP TO THIS LOCATION
780	000416	000000	HALT	:EXAMINE STACK TO FIND CAUSE
781	000420	000422	.+2	:UNEXPECTED TRAP TO THIS LOCATION
782	000422	000000	HALT	:EXAMINE STACK TO FIND CAUSE
783	000424	000426	.+2	:UNEXPECTED TRAP TO THIS LOCATION
784	000426	000000	HALT	:EXAMINE STACK TO FIND CAUSE
785	000430	000432	.+2	:UNEXPECTED TRAP TO THIS LOCATION
786	000432	000000	HALT	:EXAMINE STACK TO FIND CAUSE
787	000434	000436	.+2	:UNEXPECTED TRAP TO THIS LOCATION
788	000436	000000	HALT	:EXAMINE STACK TO FIND CAUSE
789	000440	000442	.+2	:UNEXPECTED TRAP TO THIS LOCATION
790	000442	000000	HALT	:EXAMINE STACK TO FIND CAUSE
791	000444	000446	.+2	:UNEXPECTED TRAP TO THIS LOCATION
792	000446	000000	HALT	:EXAMINE STACK TO FIND CAUSE
793	000450	000452	.+2	:UNEXPECTED TRAP TO THIS LOCATION
794	000452	000000	HALT	:EXAMINE STACK TO FIND CAUSE
795	000454	000456	.+2	:UNEXPECTED TRAP TO THIS LOCATION
796	000456	000000	HALT	:EXAMINE STACK TO FIND CAUSE
797	000460	000462	.+2	:UNEXPECTED TRAP TO THIS LOCATION
798	000462	000000	HALT	:EXAMINE STACK TO FIND CAUSE
799	000464	000466	.+2	:UNEXPECTED TRAP TO THIS LOCATION
800	000466	000000	HALT	:EXAMINE STACK TO FIND CAUSE
801	000470	000472	.+2	:UNEXPECTED TRAP TO THIS LOCATION
802	000472	000000	HALT	:EXAMINE STACK TO FIND CAUSE
803	000474	000476	.+2	:UNEXPECTED TRAP TO THIS LOCATION
804	000476	000000	HALT	:EXAMINE STACK TO FIND CAUSE
805	000500	000502	.+2	:UNEXPECTED TRAP TO THIS LOCATION
806	000502	000000	HALT	:EXAMINE STACK TO FIND CAUSE
807	000504	000506	.+2	:UNEXPECTED TRAP TO THIS LOCATION
808	000506	000000	HALT	:EXAMINE STACK TO FIND CAUSE
809	000510	000512	.+2	:UNEXPECTED TRAP TO THIS LOCATION
810	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

DZDGB MACY11 27(1006) 22-DEC-76 11:05 PAGE 19
 DZDGB.P11 16-DEC-76 13:23 TRAPCATCHER FOR UNEXPECTED INTERRUPTS

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

H02

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 20
 DZDQBC.P11 16-DEC-76 13:23 ROUTINES USED FOR AUTO SIZING.

```

901                                     ;STANDARD INTERRUPT VECTORS
902
903                                     . =24
904 000024 011056                       .PFAIL                               ;POWER FAIL HANDLER
905 000026 000340                       340                               ;SERVICE AT LEVEL 7
906 000030 010526                       .HLT                               ;ERROR HANDLER
907 000032 000340                       340                               ;SERVICE AT LEVEL 7
908 000034 010474                       .TRPSRV                            ;GENERAL HANDLER DISPATCH SERVICE
909 000036 000340                       340                               ;SERVICE AT LEVEL 7
910                                     . =46
911 000046 007254                       LOGICAL                            ;ACT HOOKS
912                                     . =52
913 000052 000000                       .WORD 0
914                                     ; THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
915                                     ; TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
916                                     ; FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
917                                     ; NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
918                                     ; EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
919                                     ; TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
920                                     . =56
921                                     000056
922 000056
923 000056 010120
924 000060 012721 000004
925 000064 022021
926 000066 020127 001000
927 000072 101771
928 000074 012737 000146 000020
929 000102 013737 001500 001244
930 000110 006037 001244
931 000114 103023
932 000116 005037 177776
933 000122 005722
934 000124 012772 000340 177776
935 000132 105200
936 000134 001376
937 000136 112712 000300
938 000142 005722
939 000144 000761
940 000146 051612
941 000150 042712 000007
942 000154 022626
943 000156 012716 000142
944 000162 000002
945 000164 000207
946
947                                     ;*****SOFTWARE SWITCH REGISTER*****
948                                     . =174
949 000174 000000
950 000176 000000
951
952                                     ;PROGRAM START
953
954                                     . =200
955 000200 000200 001512
956                                     JMP .START                       ;GO TO START OF PROGRAM
  
```

```

VECMAP:
1$:  MOV R1,(R0)+                       ;START FILLING THE VECTOR AREA
     MOV #4,(R1)+                       ;WITH +2; IOT (4)
     CMP (R0)+,(R1)+                     ;UPDATE THE POINTERS
     CMP R1,#1000                         ;IS ALL FLOATING VECTOR AREA DONE
     BLOS 1$                               ;BR IF NOT ALL DONE
     MOV #4$,#20                          ;SET FOR IOT TRAP BY DQ11
     MOV DQACTV,TEMP1                     ;GET THE ACTIVE DQ11 S
2$:  ROR TEMP1                            ;ARE YOU ACTIVE.. DQ11
     BCC 5$                               ;IF CARRY CLEAR.. NO MORE DQ11S
     CLR PS                               ;CLEAR PS
     TST (R2)+                             ;PUT POINTER TO STATUS TABLE
     MOV #340,#2-(R2)                     ;TRY AND SET PRI/SEC DONE AND IE
     INCB R0                               ;DELAY.....
     BNE .-2                               ;.....DELAY
     MOVB #300,(R2)                       ;NO INTERRUPT ASSUME 300 FIX IN TEST C
3$:  TST (R2)+                             ;UPDATE POINTERS
     BR 2$                                ;GO DO IT AGAIN
4$:  BIS (SP),(R2)                         ;ENTERD BY IOT TRAP BY DQ11
     BIC #7,(R2)                          ;CLEAR UNWANTED BITS
     CMP (SP)+,(SP)+                       ;POP IOT JUNK OFF STACK
     MOV #3$, (SP)                         ;SET RETURN PC ON STACK
5$:  RTI PC                               ;GO HOME
     RTS PC                               ;ALL SIZING IS DONE
  
```


1013	000540	012737	000006	000004	MOV	#6, R4	: RESET TIME OUT VECTOR
1014	000546	013737	001500	001502	MOV	DQACTV, SAVACT	: SAVE ACTIVE
1015	000554	012737	000340	000022	MOV	#340, R22	: 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			4S: 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		5S: 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 ITERATIONS 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

K02

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

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
1087
1088 001310 000
1089 001311 000
1090 001312 000
1091 001313 000
1092 000000
1093
1094
1095
1096
1097
1098
1099
1100 001314
1101 001314 104400
1102 001314 007330
1103 001316 104401
1104 001316 007442
1105 001320 104402
1106 001320 007462
1107 001322 104403
1108 001322 007570
1109 001324 104404
1110 001324 007706
1111 001326 104405
1112 001326 007740
1113 001330 104406
1114 001330 010154
1115 001332 104407
1116 001332 010214
1117 001334 104410
1118 001334 010246
1119 001336 104411
1120 001336 010252
1121 001340 104412
1122 001340 012114
1123 001342 104413
1124 001342 012112
1125 001344 104414
1126 001344 011154
1127 001346 104415
1128 001346 011230
1129
1130
1131
1132
1133
1134
1135 001350 000000
1136 001352 000000
1137 001354 000000
1138 001356 000000
1139 001360 000000
1140 001362 000000

;PROGRAM CONTROL FLAGS
INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
STFLG: .BYTE 0 ;TEST START FLAG
ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
SY=0

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

;*****
;*****
;*****
TRPTAB:
SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
;SCOPE
SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
;SCOPI
TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
;TYPE
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
;INSTR
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
;INSTER
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
;PARAM
SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
;SAVOS
RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
;RESOS
CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
;CONVRT
CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
;CNVRT
MSTCLR=TRAP+12 ;CALL TO ISSUE MASTER CLEAR
;MSTCLR
MEMCLR=TRAP+13 ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
;MEMCLR
CKSMR=TRAP+14 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
;CKSMR
CNTLU=TRAP+15 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
;CNTLU

;*****
;*****

;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
DQAVEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
DQRLVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
DQTVEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
DQTLVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
DQRC SR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
DQRC SH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER
    
```

M02

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 25
 DZD08C.P11 16-DEC-76 13:23

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1141	001364	000000	DQTCRS: 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

;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

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 OF 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	; PROGRAM INITIALIZATION
1190	; LOCK OUT INTERRUPTS
1191	; SET UP PROCESSOR STACK
1192	; SET UP POWER FAIL VECTOR
1193	; CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1194	; TYPE TITLE MESSAGE
1195	
1196	

N02

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 26
 DZD08C.P11 16-DEC-76 13:23 PROGRAM INITIALIZATION AND START UP.

```

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,2#24 ;SET UP POWER FAIL VECTOR
1200 001532 013737 001504 001276 MOV #DNUM,SAVNUM
1201 001540 105037 001311 CLR# STFLG ;CLEAR START FLAG
1202 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
1203 001550 105037 001312 CLR# 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 125
1213 001620 104402 001000 TYPE #TITLE ;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 2#6,-(SP) ;SAVE VECTORS
1219 001650 013746 000004 MOV 2#4,-(SP)
1220 001654 012737 001674 000004 MOV #648,2#4 ;SET UP FOR TIMEOUT
1221 001662 022777 177777 177310 CMP #-1,2SWR ;REFERENCE HARDWARE SWITCH REGISTER
1222 001670 001402 BEQ 655
1223 001672 000407 BR 665
1224 001674 022626 645: CMP (SP)+,(SP)+ ;ADJUST STACK
1225 001676 012737 000176 001200 655: 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 665: MOV (SP)+,2#4 ;RESTORE VECTORS
1228 001716 012637 000006 MOV (SP)+,2#6
1229 001722 005737 000042 TST 2#42 ;UNDER MONITOR
1230 001726 001005 BNE 675
1231 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
1232 001736 001001 BNE 675
1233 001740 104415 CNTLU
1234 001742 105777 177232 675: TSTB 2SWR
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 2TEMP1,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 125: BIT #SW00,2SWR
1246 002020 001424 BEQ 15
1247 002022 104402 TYPE
1248 002024 011625 MNEW
1249 002026 005000 CLR RO
1250 002030 000000 HALT
1251 002032 104414 CKSWR
1252 002034 027737 177140 001502 CMP 2SWR,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
1300 ; TEST 1
1301 002254 012737 000001 001226 *****
1302 002262 012737 002644 001214 TST1: MOV      #1,TSTNO
1303 002270 012737 002644 001216 MOV      @TST2,RETURN
1304 002276 105737 001302 MOV      @TST2,NEXT
1305 002302 001010      TSTB     RUNFLG ;IS THIS MY FIRST TIME HERE?
1306 002304 012737 000001 001304 BNE      1$ ;BR IF FLAG IS SET
1307 002312 012737 000020 001306 MOV      #BIT0,RUN ;SET RUN POINTER.
1308 002320 105137 001302 COMB     RUNFLG ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
;SET RUN FLAG

```

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 28
 DZD08C.P11 16-DEC-76 13:23 PROGRAM INITIALIZATION AND START UP.

1309	002324	033737	001304	001500	15:	BIT	RUN,DQACTV	;FIND AN ACTIVE DQ11 TO TEST.
1310	002332	001032				BNE	35	;BR IF I FOUND ONE TO TEST.
1311	002334	005737	001500			TST	DQACTV	;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1312	002340	001423				BEG	25	;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	15	;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	15	;KEEP ON TESTING.
1322	002410	104402			25:	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			35:	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 ADDRESS 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,DQCSR	
1333	002464	013737	001510	001350		MOV	DQSTAT,DQVEC	
1334	002472	042737	177007	001350		BIC	#177007,DQVEC	
1335	002500	013737	001350	001352		MOV	DQVEC,DQRLVL	;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1336	002506	062737	000002	001352		ADD	#2,DQRLVL	
1337	002514	013737	001352	001354		MOV	DQRLVL,DQTEC	;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1338	002522	062737	000002	001354		ADD	#2,DQTEC	
1339	002530	013737	001354	001356		MOV	DQTEC,DQTLVL	;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1340	002536	062737	000002	001356		ADD	#2,DQTLVL	
1341	002544	013737	001360	001362		MOV	DQCSR,DQCSH	
1342	002552	005237	001362			INC	DQCSH	;GENERATE ADDRESS OF HIGH BYTE
1343	002556	013737	001360	001364		MOV	DQCSR,DQCSR	;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1344	002564	062737	000002	001364		ADD	#2,DQCSR	
1345	002572	013737	001364	001366		MOV	DQCSR,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	

1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371

002644

002644 012737 000002 001226
002652 012737 002706 001216
002660 112777 000020 176502

002666 105077 176500
002672 132777 000020 176470
002700 001401
002702 104012
002704 104400

OPT2X:

; TEST 2

TST2:

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

MOV #2,TSTNO
MOV #TST3,NEXT
MOVB #BIT4,JDQREG

CLRB JDQSEC
BITB #BIT4,JDQREG
BEQ 15
HLT 12
15: SCOPE

:SET WRITE ENABLE
:SELECT RECEIVER BUS ADDRESS
:MEMORY
:CLEAR MEMORY
:WAS WRITE ENABLE CLEARED
:BR IF WRITE ENABLE CLEARED
:WRITE ENABLE NOT CLEARED

1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422

002706 012737 000003 001226
002714 012737 002774 001216
002722 013703 001370
002726 142777 000037 176434
002734 117705 176430
002740 010500
002742 105100
002744 142700 000237
002750 110077 176414
002754 117704 176410
002760 142704 000037
002764 020504
002766 001401
002770 104000
002772 104400

```

: MEMORY EXTENSION BITS READ/WRITE TEST
: READ MEMORY EXTENSION BITS
: WITH WRITE ENABLE BITS CLEARED, ATTEMPT
: TO CHANGE MEMORY EXTENSION BITS
: VERIFY THAT NO CHANGE OCCURS

: TEST 3
: *****
TST3: MOV #3, TSTNO
      MOV #TST4, NEXT
      MOV DQREG, R3 ; SET FOR ERROR PRINTOUT
      BICB #37, DQREG ; CLEAR MEM EXT AND WRITE ENABLE
      MOVB DQREG, R5 ; SAVE REGISTER
      MOV R5, R0 ; STORE REG
      COMB R0 ; CHANGE CONTENTS
      BICB #237, R0 ; CLEAR UNWANTED BITS
      MOVB R0, DQREG ; WRITE REGISTER
      MOVB DQREG, R4 ; READ REGISTER
      BICB #37, R4 ; CLEAR UNWANTED BITS
      CMP R5, R4 ; DID IT CHANGE
      BEQ IS ; BR IF NO CHANGE OCCURED
      HLT 0 ; REPORT ERROR
IS: SCOPE ; SCOPE THIS TEST

```

```

: MEMORY EXTENSION READ/WRITE TEST
: READ MEMORY EXTENSION BITS
: WITH WRITE ENABLE SET, ATTEMPT TO CHANGE
: MEMORY EXTENSION BITS
: VERIFY THAT MEMORY EXTENSION BITS WERE CHANGED

```

```

: TEST 4
: *****
TST4: MOV #4, TSTNO
      MOV #TST5, NEXT
      MOV DQREG, R3 ; SET FOR ERROR REPORT
      BICB #37, DQREG ; CLEAR UNWANTED BITS.
      MOVB DQREG, R0 ; READ DQREG FOR DATA
      COMB R0 ; DO A COMPLETE CHANGE OF DATA
      BICB #237, R0 ; CLEAR WRITE ENABLE AND GARBAGE
      MOV R0, R5 ; STORE DATA.
      BISB #BIT4, R5 ; SET WRITE ENABLE
      MOVB R5, DQREG ; WRITE DQREG WITH NEW DATA
      CLR DQSEC ; REFERENCE SEL 6 (DQSEC)
      MOVB DQREG, R4 ; READ DQREG
      BIC #37, R4 ; CLEAR UNWANTED BITS.
      BIC #20, R5 ; CLEAR WRITE ENABLE
      CMP R5, R4 ; ARE THEY EQUAL?
      BEQ IS ; BR IF GOOD
      HLT 0 ; MEMORY EXTENSION READ/WRITE ERROR
IS: SCOPE ; SCOPE THE TEST.

```

```

1423
1424
1425
1426
1427
1428
1429 003076 005003
1430 003100 052701 000020
1431 003104 112777 000020 176256
1432 003112 150377 176252
1433 003116 110177 176246
1434 003122 005077 176244
1435 003126 062703 000002
1436 003132 062705 000040
1437 003136 005300
1438 003140 001355
1439 003142 005005
1440 003144 012700 000004
1441 003150 005003
1442 003152 142777 000037 176210 2S:
1443 003160 150377 176204
1444 003164 117704 176200
1445 003170 142704 000037
1446 003174 020504
1447 003176 001401
1448 003200 104001
1449 003202 104401 3S:
1450 003204 062703 000002
1451 003210 062705 000040
1452 003214 005300
1453 003216 001355
1454 003220 104400 4S:
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464 003222 012737 000005 001226
1465 003230 012737 003336 001216
1466 003236 012737 003306 001220
1467 003244 012705 000000
1468 003250 013703 001372
1469 003254 005002
1470 003256 012700 000010
1471 003262 110277 176102 1S:
1472 003266 010577 176100
1473 003272 005202
1474 003274 005300
1475 003276 001371
1476 003300 005002
1477 003302 012700 000010
1478 003306 110277 176056 2S:

```

;BUS ADDRESS EXTENSION MEMORY TEST
 ;LOAD EACH BUS ADDRESS MEMORY LOCATION WITH A DIFFERENT
 ;NUMBER (0-3)
 ;VERIFY THAT EACH LOCATION WAS ADDRESSED PROPERLY
 ;FIRST MEMORY ADDRESS =0
 ;SET WRITE ENABLE
 ;SET MEMORY EXTENSION WRITE ENABLE
 ;SELECT LOCATION TO BE TESTED
 ;LOAD DATA INTO MEMORY EXTENSION BITS
 ;REFERENCE SEL 6 (DQSEC)
 ;NEXT LOCATION TO BE TESTED
 ;DATA TO BE LOADED
 ;CONTINUE IF ALL LOCATIONS NOT LOADED
 ;BR IF MORE TO GO
 ;INITIAL DATA = 0
 ;4 LOCATIONS TO BE TESTED
 ;FIRST LOCATION
 ;REINITIALIZE ADDRESS
 ;RESELECT SECONDARY ADDRESS.
 ;READ MEMORY EXTENSION BITS
 ;CLEAR UNWANTED BITS
 ;ARE EXPECTED AND RECEIVED DATA THE SAME
 ;BUS ADDRESS EXTENSION MEMORY ERROR
 ;CHECK FOR LOOP ON CUREENT ADDRESS
 ;UPDATE SELECT ADDRESS
 ;UPDATE EXPECTED ATA
 ;CONTINUE IF NOT DONE
 ;BR IF MORE TO CHECK.
 ;CHECK FOR ITERATIONS, LOOP
 ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
 ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
 ;TO 0
 ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
 ;COUNT MEMORY WAS SET TO 0
 ; TEST 5
 ;*****
 TST5: MOV #5,TSTNO
 MOV #TST6,NEXT
 MOV #2S,LOCK
 MOV #0,R5 ;DATA TO BE LOADED INTO MEMORY
 MOV DQSEC,R3 ;ADDRESS OF SECONDARY REGISTER
 CLR R2 ;FIRST MEMORY LOCATION TO BE TESTED
 MOV #10,R0 ;# OF MEMORY LOCATIONS TO BE TESTED
 1S: MOV R2,DQREG ;SELECT MEMORY LOCATION TO BE TESTED
 MOV R5,DQSEC ;LOAD DATA INTO MEMORY
 INC R2 ;GO TO NEXT LOCATION
 DEC R0 ;MORE TO GO??
 BNE 1S ;BR IF NOT DONE
 CLR R2 ;SET UP TO TEST
 MOV #10,R0 ;10 (OCTAL) LOCATIONS
 2S: MOV R2,DQREG ;SELECT LOCATION TO BE TESTED

1479	003312	017704	176054		MOV	3DQSEC,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\$:	SCOPI		: 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
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529

```

: BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
: SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
: TO 177777
: VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
: COUNT MEMORY WAS SET TO 177777

```

: TEST 6
: *****

```

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	R2,3DQREG	: SELECT MEMORY LOCATION TO BE TESTED
1505	003402	010577	175764			MOV	R5,3DQSEC	: 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	R2,3DQREG	: SELECT LOCATION TO BE TESTED
1512	003426	017704	175740			MOV	3DQSEC,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\$:	SCOPI		: 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
1523
1524
1525
1526
1527
1528
1529

```

: BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
: SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
: TO 125252
: VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
: COUNT MEMORY WAS SET TO 125252

```

: TEST 7
: *****

```

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

H03

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 33
DZDQBC.P11 16-DEC-76 13:23 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	1S:	MOVB	R2,2DQREG		:SELECT MEMORY LOCATION TO BE TESTED
1538	003516	010577	175650		MOV	R5,2DQSEC		: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	1S		: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	2S:	MOVB	R2,2DQREG		:SELECT LOCATION TO BE TESTED
1545	003542	017704	175624		MOV	2DQSEC,R4		:READ DATA
1546	003546	020504			CMP	R5,R4		:COMPARE TO EXPECTED DATA
1547	003550	001401			BEQ	3S		:BR IF GOOD
1548	003552	104002			HLT	2		:CHAR COUNT-BUS ADDRESS MEMORY ERROR
1549	003554	104401		3S:	SCOPI			: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	2S		:BR IF FURTHER CHECKING NEEDED
1553	003564	104400		4S:	SCOPE			

:BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
:SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
:TO 52525
:VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
:COUNT MEMORY WAS SET TO 52525

: TEST 10

:*****

1563	003566	012737	000010	001226	TST10:	MOV	#10,TSTNO	
1564	003574	012737	003702	001216		MOV	#TST11,NEXT	
1565	003602	012737	003652	001220		MOV	#2S,LOCK	
1566	003610	012705	052525			MOV	#52525,R5	:DATA TO BE LOADED INTO MEMORY
1567	003614	013703	001372			MOV	0QSEC,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		1S:	MOVB	R2,2DQREG	:SELECT MEMORY LOCATION TO BE TESTED
1571	003632	010577	175534			MOV	R5,2DQSEC	: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	1S	: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		2S:	MOVB	R2,2DQREG	:SELECT LOCATION TO BE TESTED
1578	003656	017704	175510			MOV	2DQSEC,R4	:READ DATA
1579	003662	020504				CMP	R5,R4	:COMPARE TO EXPECTED DATA
1580	003664	001401				BEQ	3S	:BR IF GOOD
1581	003666	104002				HLT	2	:CHAR COUNT-BUS ADDRESS MEMORY ERROR
1582	003670	104401			3S:	SCOPI		: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	2S	:BR IF FURTHER CHECKING NEEDED
1586	003700	104400			4S:	SCOPE		

:BUS ADDRESS MEMORY EXTENSION DATA TEST
:LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 40
:VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA

1587
1588
1589
1590

```

1591
1592
1593
1594 003702 012737 000011 001226
1595 003710 012737 004052 001216
1596 003716 012737 004006 001220
1597 003724 012705 000040
1598 003730 012700 000004
1599 003734 005003
1600
1601 003736 010501
1602 003740 050301
1603 003742 052701 000020
1604 003746 112777 000020 175414
1605 003754 150377 175410
1606 003760 110177 175404
1607 003764 005077 175402
1608 003770 062703 000002
1609 003774 005300
1610 003776 001357
1611 004000 012700 000004
1612 004004 005003
1613 004006 142777 000037 175354
1614 004014 150377 175350
1615 004020 117704 175344
1616 004024 142704 000037
1617 004030 020504
1618 004032 001401
1619 004034 104003
1620 004036 104401
1621 004040 062703 000002
1622 004044 005300
1623 004046 001357
1624 004050 104400
1625
1626
1627
1628
1629
1630
1631
1632 004052 012737 000012 001226
1633 004060 012737 004222 001216
1634 004066 012737 004156 001220
1635 004074 012705 000100
1636 004100 012700 000004
1637 004104 005003
1638
1639 004106 010501
1640 004110 050301
1641 004112 052701 000020
1642 004116 112777 000020 175244
1643 004124 150377 175240
1644 004130 110177 175234
1645 004134 005077 175232
1646 004140 062703 000002
    
```

TEST 11

```

TST11: MOV #11,TSTNO
        MOV #TST12,NEXT
        MOV #2$,LOCK
        MOV #40,R5
        MOV #4,R0
        CLR R3
        1$: MOV R5,R1
           BIS R3,R1
           BIS #BIT4,R1
           MOVB #BIT4,DQREG
           BISB R3,DQREG
           MOVB R1,DQREG
           CLR DQSEC
           ADD #2,R3
           DEC R0
           BNE 1$
           MOV #4,R0
           CLR R3
           2$: BICB #37,DQREG
              BISB R3,DQREG
              MOVB DQREG,R4
              BICB #37,R4
              CMP R5,R4
              BEQ 3$
              HLT 3
           3$: SCOP1
              ADD #2,R3
              DEC R0
              BNE 2$
           4$: SCOPE
    
```

```

: 4 LOCATIONS WILL BE ADDRESSED
: FIRST MEMORY ADDRESS =0
: (RECEIVER PRIMARY BUS ADDRESS)
: LOAD "DATA"
: LOAD SECONDARY ADDRESS
: SET WRITE ENABLE.
: SET MEMORY EXTENSION WRITE ENABLE
: SELECT LOCATION TO BE TESTED
: LOAD DATA INTO MEMORY EXTENSION BITS
: REFERENCE SEL 6 (DQSEC)
: NEXT LOCATION TO BE TESTED
: CONTINUE IF ALL LOCATIONS NOT LOADED

: 4 LOCATIONS TO BE TESTED
: FIRST LOCATION
: REINITIALIZE ADDRESS
: RESELECT SECONDARY ADDRESS
: READ DQREG
: CLEAR UNWANTED BITS
: IS DATA CORRECT?
: BR IF OK.
: BUS ADDRESS EXTENSION MEMORY ERROR
: FREEZE ON CURRENT DATA? (SW09=1)
: UPDATE POINTER TO NEXT ADDRESS
: ALL ADDRESS DONE?
: BR IF NOT DONE.
: SCOPE THE TEST.
    
```

```

: BUS ADDRESS MEMORY EXTENSION DATA TEST
: LOAD EACH LOCATION IN BUS ADDRESS EXTENSION MEMORY WITH 100
: VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA
    
```

TEST 12

```

TST12: MOV #12,TSTNO
        MOV #TST13,NEXT
        MOV #2$,LOCK
        MOV #100,R5
        MOV #4,R0
        CLR R3
        1$: MOV R5,R1
           BIS R3,R1
           BIS #BIT4,R1
           MOVB #BIT4,DQREG
           BISB R3,DQREG
           MOVB R1,DQREG
           CLR DQSEC
           ADD #2,R3
    
```

```

: 4 LOCATIONS WILL BE ADDRESSED
: FIRST MEMORY ADDRESS =0
: (RECEIVER PRIMARY BUS ADDRESS)
: LOAD "DATA"
: LOAD SECONDARY ADDRESS
: SET WRITE ENABLE.
: SET MEMORY EXTENSION WRITE ENABLE
: SELECT LOCATION TO BE TESTED
: LOAD DATA INTO MEMORY EXTENSION BITS
: REFERENCE SEL 6 (DQSEC)
: 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\$:	SCOPI		: FREEZE ON CURRENT DATA? (SM09=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.

: BUS ADDRESS MEMORY EXTENSION DATA TEST
 : LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 140
 : VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA

: TEST 13

: *****

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	
1675	004254	005003				CLR	R3	
1676								: 4 LOCATIONS WILL BE ADDRESSED
1677	004256	010501			1\$:	MOV	R5,R1	: FIRST MEMORY ADDRESS =0
1678	004260	050301				BIS	R3,R1	: (RECEIVER PRIMARY BUS ADDRESS)
1679	004262	052701	000020			BIS	#BIT4,R1	: LOAD "DATA"
1680	004266	112777	000020	175074		MOVB	#BIT4,DQREG	: LOAD SECONDARY ADDRESS
1681	004274	150377	175070			BISB	R3,DQREG	: SET WRITE ENABLE
1682	004300	110177	175064			MOVB	R1,DQREG	: SET MEMORY EXTENSION WRITE ENABLE
1683	004304	005077	175062			CLR	DQSEC	: SELECT LOCATION TO BE TESTED
1684	004310	062703	000002			ADD	#2,R3	: LOAD DATA INTO MEMORY EXTENSION BITS
1685	004314	005300				DEC	R0	: REFERENCE SEL 6 (DQSEC)
1686	004316	001357				BNE	1\$: NEXT LOCATION TO BE TESTED
1687	004320	012700	000004			MOV	#4,R0	: CONTINUE IF ALL LOCATIONS NOT LOADED
1688	004324	005003				CLR	R3	
1689	004326	142777	000037	175034	2\$:	BICB	#37,DQREG	: 4 LOCATIONS TO BE TESTED
1690	004334	150377	175030			BISB	R3,DQREG	: FIRST LOCATION
1691	004340	117704	175024			MOVB	DQREG,R4	: REINITIALIZE ADDRESS
1692	004344	142704	000037			BICB	#37,R4	: RESELECT SECONDARY ADDRESS
1693	004350	020504				CMP	R5,R4	: READ DQREG
1694	004352	001401				BEQ	3\$: CLEAR UNWANTED BITS
1695	004354	104003				HLT	3	: IS DATA CORRECT?
1696	004356	104401			3\$:	SCOPI		: BR IF OK.
1697	004360	062703	000002			ADD	#2,R3	: BUS ADDRESS EXTENSION MEMORY ERROR
1698	004364	005300				DEC	R0	: FREEZE ON CURRENT DATA? (SM09=1)
1699	004366	001357				BNE	2\$: UPDATE POINTER TO NEXT ADDRESS
1700	004370	104400			4\$:	SCOPE		: ALL ADDRESS DONE?
1701								: BR IF NOT DONE.
1702								: SCOPE THE TEST.

: IF CHARACTER DETECT OPTION IS INSTALLED

K03

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 36
DZD08C.P11 16-DEC-76 13:23 CHARACTER AND SEQUENCE MEMORY TESTS

```
1703 ;TESTS 14 THRU 25 WILL BE EXECUTED
1704
1705 004372 032737 020000 001510 CHKBB1: BIT #BBBIT,DQSTAT
1706 004400 001002 BNE TST14
1707 004402 000137 006206 JMP TST26
1708
1709 ;CHARACTER MEMORY ADDRESSING TEST
1710 ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH ITS
1711 ;ADDRESS (DUPLICATED EVERY 4 BITS)
1712 ;VERIFY THAT EACH LOCATION WAS ADDRESSED
1713
1714 ; TEST 14
1715 ;*****
1716 004406 012737 000014 001226 TST14: MOV #14,TSTNO
1717 004414 012737 004546 001216 MOV #TST15,NEXT
1718 004422 012737 004512 001220 MOV #25,LOCK
1719 004430 005005 CLR R5 ;CLEAR DATA TO BE LOADED
1720 004432 012700 000020 MOV #16.,R0 ;16(DECIMAL) REGISTERS WILL BE TESTED
1721 004436 005002 CLR R2 ;FIRST ADDRESS =0
1722 004440 012703 000010 MOV #10,R3 ;CHARACTER MEMORY WILL BE SELECTED
1723 004444 112777 000010 174716 MOVB #10,30QREG ;SELECT CHARACTER MEMORY
1724 004452 110277 174704 15: MOVB R2,30QRC5H ;SELECT LOCATION IN CHARACTER MEMORY
1725 004456 010577 174710 MOV #30QSEC,R4 ;LOAD DATA
1726 004462 005202 INC R2 ;UPDATE ADDRESS
1727 004464 062705 010421 ADD #10421,R5 ;UPDATE DATA
1728 004470 005300 DEC R0 ;CONTINUE IF NOT DONE
1729 004472 001367 BNE 1$
1730 004474 005005 CLR R5 ;CLEAR EXPECTED DATA
1731 004476 012700 000020 MOV #16.,R0 ;16 (DECIMAL) LOCATIONS WILL BE TESTED
1732 004502 005002 CLR R2 ;FIRST LOCATION =0
1733 004504 112777 000010 174656 MOVB #10,30QREG ;SELECT CHARACTER REGISTER
1734 004512 110277 174644 25: MOVB R2,30QRC5H ;SELECT ADDRESS TO BE TESTED
1735 004516 017704 174650 MOV #30QSEC,R4 ;READ ADDRESS TO BE TESTED
1736 004522 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
1737 004524 001401 BEQ 3$ ;BR IF GOOD.
1738 004526 104004 HLT 4 ;CHARACTER MEMORY ADDRESS ERROR
1739 004530 104401 3$: SCOP1 ;CHECK FOR LOOP ON CURENT ADDRESS
1740 004532 005202 INC R2 ;UPDATE ADDRESS TO BE TESTED
1741 004534 062705 010421 ADD #10421,R5 ;UPDATE EXPECTED DATA
1742 004540 005300 DEC R0 ;CONTINUE IF NOT DONE
1743 004542 001363 BNE 2$
1744 004544 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1745
1746 ;SEQUENCE MEMORY ADDRESSING TEST
1747 ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH ITS
1748 ;ADDRESS (DUPLICATED EVERY 4 BITS)
1749 ;VERIFY THAT EACH LOCATION WAS ADDRESSED
1750
1751 ; TEST 15
1752 ;*****
1753 004546 012737 000015 001226 TST15: MOV #15,TSTNO
1754 004554 012737 004706 001216 MOV #TST16,NEXT
1755 004562 012737 004652 001220 MOV #25,LOCK
1756 004570 005005 CLR R5 ;CLEAR DATA TO BE LOADED
1757 004572 012700 000020 MOV #16.,R0 ;16(DECIMAL) REGISTERS WILL BE TESTED
1758 004576 005002 CLR R2 ;FIRST ADDRESS =0
```

L03

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 37
 DZD08C.P11 16-DEC-76 13:23 CHARACTER AND SEQUENCE MEMORY TESTS

1759	004600	012703	000014		MOV	#14,R3	:SEQUENCE MEMORY WILL BE SELECTED
1760	004604	112777	000014	174556	MOVB	#14,20QREG	:SELECT SEQUENCE MEMORY
1761	004612	110277	174544		MOVB	R2,20QRC5H	:SELECT LOCATION IN SEQUENCE MEMORY
1762	004616	010577	174550		MOV	R5,20QSEC	: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	#14,20QREG	:SELECT SEQUENCE REGISTER
1771	004652	110277	174504		MOVB	R2,20QRC5H	:SELECT ADDRESS TO BE TESTED
1772	004656	017704	174510		MOV	20QSEC,R4	:READ ADDRESS TO BE TESTED
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			SCOPI		: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			SCOPE		:CHECK FOR ITERATIONS, LOOP

: CHARACTER MEMORY TEST
 : LOAD EACH LOCATION IN CHARACTER MEMORY WITH 00000
 : VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
 : LOADED WITH 00000

: TEST 16
 : *****

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	#00000,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	#10,20QREG	:SELECT THE REGISTER
1798	004754	110277	174402			MOVB	R2,20QRC5H	:SELECT THE CHARACTER DET REG
1799	004760	010577	174406		1\$:	MOV	R5,20QSEC	: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	#10,20QREG	:SELECT THE REGISTER
1806	005006	110277	174350		2\$:	MOVB	R2,20QRC5H	:SELECT THE CHARACTER DET REGISTER
1807	005012	017704	174354			MOV	20QSEC,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\$:	SCOPI		:CHECK FOR FREEZE ON DATA
1812	005026	005202				INC	R2	:UPDATE POINTER
1813	005030	005300				DEC	R0	:CHECK FOR ALL DONE
1814	005032	001365				BNE	2\$:BR IF NOT DONE

M03

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          4S:  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 18:  TST17:  MOV      #17,TSTNO
1825 005044 012737 005166 001216          MOV      #TST20,NEXT
1826 005052 012737 005136 001220          MOV      #25,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 1S:  MOVB    #10,30QREG      ;SELECT THE REGISTER
1832 005104 110277 174252          MOVB    R2,30QRC5H      ;SELECT THE CHARACTER DET REG
1833 005110 010577 174256          MOV      R5,30QSEC      ;LOAD THE DATA
1834 005114 005202          INC      R2             ;UPDATE THE POINTER
1835 005116 005300          DEC      R0             ;ALL DONE YET?
1836 005120 001371          BNE     1S             ;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 2S:  MOVB    #10,30QREG      ;SELECT THE REGISTER
1840 005136 110277 174220          MOVB    R2,30QRC5H      ;SELECT THE CHARACTER DET REGISTER
1841 005142 017704 174224          MOV      30QSEC,R4      ;READ THE DATA
1842 005146 020504          CMP      R5,R4          ;WAS IT LOADED CORRECTLY?
1843 005150 001401          BEQ     3S             ;BR IF DATA OK
1844 005152 104006          HLT     6              ;INCORRECT DATA REPORT ERROR
1845 005154 104401 3S:  SCOPI          ;CHECK FOR FREEZE ON DATA
1846 005156 005202          INC      R2             ;UPDATE POINTER
1847 005160 005300          DEC      R0             ;CHECK FOR ALL DONE
1848 005162 001365          BNE     2S            ;BR IF NOT DONE
1849 005164 104400 4S:  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 18:  TST20:  MOV      #20,TSTNO
1859 005174 012737 005316 001216          MOV      #TST21,NEXT
1860 005202 012737 005266 001220          MOV      #25,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 1S:  MOVB    #10,30QREG      ;SELECT THE REGISTER
1866 005234 110277 174122          MOVB    R2,30QRC5H      ;SELECT THE CHARACTER DET REG
1867 005240 010577 174126          MOV      R5,30QSEC      ;LOAD THE DATA
1868 005244 005202          INC      R2             ;UPDATE THE POINTER
1869 005246 005300          DEC      R0             ;ALL DONE YET?
1870 005250 001371          BNE     1S            ;BR IF NOT DONE

```

```

1871 005252 012700 000020      MOV      #16.,R0      ;SET FOR 16 REGISTERS TO BE TESTED
1872 005256 005002      CLR      R2          ;ZERO REG POINTER
1873 005260 112777 000010 174102      MOVB    #10,ADQREG   ;SELECT THE REGISTER
1874 005266 110277 174070 25:      MOVB    R2,ADQRC5H   ;SELECT THE CHARACTER DET REGISTER
1875 005272 017704 174074      MOV     ADQSEC,R4    ;READ THE DATA
1876 005276 020504      CMP     R5,R4        ;WAS IT LOADED CORRECTLY?
1877 005300 001401      BEQ     35           ;BR IF DATA OK
1878 005302 104006      HLT     6            ;INCORRECT DATA REPORT ERROR
1879 005304 104401 35:      SCOP1                   ;CHECK FOR FREEZE ON DATA
1880 005306 005202      INC     R2           ;UPDATE POINTER
1881 005310 005300      DEC     R0           ;CHECK FOR ALL DONE
1882 005312 001365      BNE     25           ;BR IF NOT DONE
1883 005314 104400 45:      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
;*****

```

```

1892 005316 012737 000021 001226 18:      TST21: MOV     #21,TSTNO
1893 005324 012737 005446 001216      MOV     #TST22,NEXT
1894 005332 012737 005416 001220      MOV     #25,LOCK
1895 005340 012705 052525      MOV     #52525,R5    ;LOAD THE DATA INTO R5
1896 005344 012700 000020      MOV     #16.,R0      ;SET COUNT FOR 16
1897 005350 005002      CLR     R2           ;CLEAR REGISTER POINTER
1898 005352 012703 000010      MOV     #10,R3       ;GET REGISTER IN R3
1899 005356 112777 000010 174004      MOVB    #10,ADQREG   ;SELECT THE REGISTER
1900 005364 110277 173772 15:      MOVB    R2,ADQRC5H   ;SELECT THE CHARACTER DET REG
1901 005370 010577 173776      MOV     R5,ADQSEC    ;LOAD THE DATA
1902 005374 005202      INC     R2           ;UPDATE THE POINTER
1903 005376 005300      DEC     R0           ;ALL DONE YET?
1904 005400 001371      BNE     15           ;BR IF NOT DONE
1905 005402 012700 000020      MOV     #16.,R0      ;SET FOR 16 REGISTERS TO BE TESTED
1906 005406 005002      CLR     R2           ;ZERO REG POINTER
1907 005410 112777 000010 173752      MOVB    #10,ADQREG   ;SELECT THE REGISTER
1908 005416 110277 173740 25:      MOVB    R2,ADQRC5H   ;SELECT THE CHARACTER DET REGISTER
1909 005422 017704 173744      MOV     ADQSEC,R4    ;READ THE DATA
1910 005426 020504      CMP     R5,R4        ;WAS IT LOADED CORRECTLY?
1911 005430 001401      BEQ     35           ;BR IF DATA OK
1912 005432 104006      HLT     6            ;INCORRECT DATA REPORT ERROR
1913 005434 104401 35:      SCOP1                   ;CHECK FOR FREEZE ON DATA
1914 005436 005202      INC     R2           ;UPDATE POINTER
1915 005440 005300      DEC     R0           ;CHECK FOR ALL DONE
1916 005442 001365      BNE     25           ;BR IF NOT DONE
1917 005444 104400 45:      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
;*****

```

```

1926 005446 012737 000022 001226 18:      TST22: MOV     #22,TSTNO

```

1927	005454	012737	005576	001216		MOV	#TST23, NEXT		
1928	005462	012737	005546	001220		MOV	#25, 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		1S:	MOVB	R2, #DQRC5H	: SELECT THE CHARACTER DET REG	
1935	005520	010577	173646			MOV	R5, #DQSEC	: LOAD THE DATA	
1936	005524	005202				INC	R2	: UPDATE THE POINTER	
1937	005526	005300				DEC	R0	: ALL DONE YET?	
1938	005530	001371				BNE	1S	: 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		2S:	MOVB	R2, #DQRC5H	: 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	3S	: BR IF DATA OK	
1946	005562	104007				HLT	7	: INCORRECT DATA REPORT ERROR	
1947	005564	104401			3S:	SCOPE		: CHECK FOR FREEZE ON DATA	
1948	005566	005202				INC	R2	: UPDATE POINTER	
1949	005570	005300				DEC	R0	: CHECK FOR ALL DONE	
1950	005572	001365				BNE	2S	: BR IF NOT DONE	
1951	005574	104400			4S:	SCOPE		: SCOPE THIS TEST	
1952									
1953									
1954									
1955									
1956									
1957									
1958									
1959									
1960	005576	012737	000023	001226		†TST23:	MOV	#23, TSTNO	
1961	005604	012737	005726	001216			MOV	#TST24, NEXT	
1962	005612	012737	005676	001220			MOV	#25, 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		1S:	MOVB	R2, #DQRC5H	: 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	1S	: 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		2S:	MOVB	R2, #DQRC5H	: 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	3S	: BR IF DATA OK	
1980	005712	104007				HLT	7	: INCORRECT DATA REPORT ERROR	
1981	005714	104401			3S:	SCOPE		: CHECK FOR FREEZE ON DATA	
1982	005716	005202				INC	R2	: UPDATE POINTER	

: SEQUENCE MEMORY TEST
 : LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 177777
 : VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
 : LOADED WITH 177777

: TEST 23
 : *****

```

1983 005720 005300
1984 005722 001365
1985 005724 104400
1986
1987
1988
1989
1990
1991
1992
1993
1994 005726 012737 000024 001226
1995 005734 012737 006056 001216
1996 005742 012737 006026 001220
1997 005750 012705 125252
1998 005754 012700 000020
1999 005760 005002
2000 005762 012703 000014
2001 005766 112777 000014 173374
2002 005774 110277 173362
2003 006000 010577 173366
2004 006004 005202
2005 006006 005300
2006 006010 001371
2007 006012 012700 000020
2008 006016 005002
2009 006020 112777 000014 173342
2010 006026 110277 173330
2011 006032 017704 173334
2012 006036 020504
2013 006040 001401
2014 006042 104007
2015 006044 104401
2016 006046 005202
2017 006050 005300
2018 006052 001365
2019 006054 104400
2020
2021
2022
2023
2024
2025
2026
2027
2028 006056 012737 000025 001226
2029 006064 012737 006206 001216
2030 006072 012737 006156 001220
2031 006100 012705 052525
2032 006104 012700 000020
2033 006110 005002
2034 006112 012703 000014
2035 006116 112777 000014 173244
2036 006124 110277 173232
2037 006130 010577 173236
2038 006134 005202

; CHECK FOR ALL DONE
; BR IF NOT DONE
; SCOPE THIS TEST
4S: DEC R0
;
; SEQUENCE MEMORY TEST
; LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 125252
; VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
; LOADED WITH 125252

; TEST 24
; *****
†ST24: MOV #24,TSTNO
MOV #TST25,NEXT
MOV #25,LOCK
MOV #125252,R5 ; LOAD THE DATA INTO R5
MOV #16.,R0 ; SET COUNT FOR 16
CLR R2 ; CLEAR REGISTER POINTER
MOV #14,R3 ; GET REGISTER IN R3
MOV #14,JDQREG ; SELECT THE REGISTER
1S: MOV R2,JDQRC5H ; SELECT THE CHARACTER DET REG
MOV R5,JDQSEC ; LOAD THE DATA
INC R2 ; UPDATE THE POINTER
DEC R0 ; ALL DONE YET?
BNE 1S ; BR IF NOT DONE
MOV #16.,R0 ; SET FOR 16 REGISTERS TO BE TESTED
CLR R2 ; ZERO REG POINTER
MOV #14,JDQREG ; SELECT THE REGISTER
2S: MOV R2,JDQRC5H ; SELECT THE CHARACTER DET REGISTER
MOV JDQSEC,R4 ; READ THE DATA
CMP R5,R4 ; WAS IT LOADED CORRECTLY?
BEQ 3S ; BR IF DATA OK
HLT 7 ; INCORRECT DATA REPORT ERROR
3S: SCOPI ; CHECK FOR FREEZE ON DATA
INC R2 ; UPDATE POINTER
DEC R0 ; CHECK FOR ALL DONE
BNE 2S ; BR IF NOT DONE
4S: 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
; *****
†ST25: MOV #25,TSTNO
MOV #TST26,NEXT
MOV #25,LOCK
MOV #52525,R5 ; LOAD THE DATA INTO R5
MOV #16.,R0 ; SET COUNT FOR 16
CLR R2 ; CLEAR REGISTER POINTER
MOV #14,R3 ; GET REGISTER IN R3
MOV #14,JDQREG ; SELECT THE REGISTER
1S: MOV R2,JDQRC5H ; SELECT THE CHARACTER DET REG
MOV R5,JDQSEC ; LOAD THE DATA
INC R2 ; UPDATE THE POINTER

```

```

2039 006136 005300          DEC      R0          ; ALL DONE YET?
2040 006140 001371          BNE      1$          ; BR IF NOT DONE
2041 006142 012700 000020    MOV      #16.,R0     ; SET FOR 16 REGISTERS TO BE TESTED
2042 006146 005002          CLR      R2          ; ZERO REG POINTER
2043 006150 112777 000014 173212  MOVB     #14,DDREG   ; SELECT THE REGISTER
2044 006156 110277 173200 2$:  MOVB     R2,DDRCSH   ; SELECT THE CHARACTER DET REGISTER
2045 006162 017704 173204    MOV      DDSEC,R4    ; READ THE DATA
2046 006166 020504          CMP      R5,R4       ; WAS IT LOADED CORRECTLY?
2047 006170 001401          BEQ      3$          ; BR IF DATA OK
2048 006172 104007          HLT      7           ; INCORRECT DATA REPORT ERROR
2049 006174 104401          SCOPE1            ; CHECK FOR FREEZE ON DATA
2050 006176 005202          INC      R2          ; UPDATE POINTER
2051 006200 005300          DEC      R0          ; CHECK FOR ALL DONE
2052 006202 001365          BNE      2$          ; BR IF NOT DONE
2053 006204 104400          SCOPE            ; SCOPE THIS TEST
2054
2055          ; RECEIVER CONTROL REGISTER MASTER CLEAR TEST
2056          ; SET ALL READ/WRITE BITS IN RECEIVER CONTROL REGISTER
2057          ; ISSUE MASTER CLEAR
2058          ; VERIFY THAT RECEIVER CONTROL WAS CLEARED
2059
2060          ; TEST 26
2061          ; *****
2062 006206 012737 000026 001226  TST26:  MOV      #26,TSTNO
2063 006214 012737 006272 001216    MOV      #TST27,NEXT
2064 006222 012737 000340 177776    MOV      #340,PS
2065 006230 013703 001360          MOV      DGRCSR,R3   ; LOCK OUT INTERRUPTS
2066          ; BUS ADDRESS OF
2067 006234 012777 170372 173116    MOV      #170372,DDRCSR ; RECEIVER CONTROL REGISTER
2068 006242 112777 000012 173120  MOVB     #12,DDREG   ; LOAD RECEIVER CONTROL REGISTER
2069 006250 052777 000040 173114    BIS      #BITS,DDSEC ; SELECT MISCELLANEOUS REGISTER
2070 006256 005005          CLR      R5          ; ISSUE MASTER CLEAR
2071          ; (R5)=EXPECTED 170372 IN
2072 006260 017704 173074    MOV      DDGRCSR,R4  ; RECEIVER CONTROL REGISTER, 0
2073          ; (R4)=ACTUAL 170372 IN
2074 006264 001401          BEQ      1$          ; RECEIVER CONTROL REGISTER
2075 006266 104000          HLT      1$         ; BR IF ALL OK.
2076 006270 104400          SCOPE            ; PRIMARY REGISTER MASTER CLEAR ERROR
2077
2078          ; RECEIVER CONTROL REGISTER MASTER CLEAR TEST
2079          ; SET ALL READ/WRITE BITS IN RECEIVER CONTROL REGISTER
2080          ; ISSUE MASTER CLEAR
2081          ; VERIFY THAT RECEIVER CONTROL WAS CLEARED
2082
2083          ; TEST 27
2084          ; *****
2085 006272 012737 000027 001226  TST27:  MOV      #27,TSTNO
2086 006300 012737 006356 001216    MOV      #CHKBA1,NEXT
2087 006306 012737 000340 177776    MOV      #340,PS
2088 006314 013703 001360          MOV      DGRCSR,R3   ; LOCK OUT INTERRUPTS
2089          ; BUS ADDRESS OF
2090 006320 012777 007400 173032    MOV      #07400,DDGRCSR ; RECEIVER CONTROL REGISTER
2091 006326 112777 000012 173034  MOVB     #12,DDREG   ; LOAD RECEIVER CONTROL REGISTER
2092 006334 052777 000040 173030    BIS      #BITS,DDSEC ; SELECT MISCELLANEOUS REGISTER
2093 006342 005005          CLR      R5          ; ISSUE MASTER CLEAR
2094          ; (R5)=EXPECTED 07400 IN
          ; RECEIVER CONTROL REGISTER, 0

```

E04

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 43
 DZDQBC.P11 16-DEC-76 13:23 VERIFICATION OF MASTER CLEAR LOGIC

```

2095 006344 017704 173010      MOV      @DQRCR,R4      ;(R4)=ACTUAL 07400 IN
2096                                ;RECEIVER CONTROL REGISTER
2097 006350 001401      BEQ      IS           ;BR IF ALL OK.
2098 006352 104000      HLT                                ;PRIMARY REGISTER MASTER CLEAR ERROR
2099 006354 104400      IS:      SCOPE
2100
2101                                ; IF DATASET CONTROL OPTION IS INSTALLED
2102                                ; TEST 30 WILL BE EXECUTED
2103
2104 006356 032737 010000 001510  CHKBA1: BIT      @BBIT,DQSTAT
2105 006364 001436      BEQ      TST31
2106
2107                                ; TRANSMITTER CONTROL REGISTER MASTER CLEAR TEST
2108                                ; SET ALL READ/WRITE BITS IN TRANSMITTER CONTROL REGISTER
2109                                ; ISSUE MASTER CLEAR
2110                                ; VERIFY THAT TRANSMITTER CONTROL WAS CLEARED
2111
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
2118 006410 013703 001364      MOV      @DQTCR,R3      ; LOCK OUT INTERRUPTS
2119                                ; BUS ADDRESS OF
2120 006414 012777 101772 172742      MOV      #101772,@DQTCR ; TRANSMITTER CONTROL REGISTER
2121 006422 112777 000012 172740      MOV      #12,@DQREG      ; LOAD TRANSMITTER CONTROL REGISTER
2122 006430 052777 000040 172734      BIS      @BITS,@DQSEC    ; SELECT MISCELLANEOUS REGISTER
2123 006436 005005      CLR      R5           ; ISSUE MASTER CLEAR
2124                                ; (R5)=EXPECTED 101772 IN
2125 006440 017704 172720      MOV      @DQTCR,R4      ; TRANSMITTER CONTROL REGISTER, 0
2126                                ; (R4)=ACTUAL 101772 IN
2127 006444 042704 100000      BIC      #100000,R4     ; TRANSMITTER CONTROL REGISTER
2128 006450 020504      CMP      R5,R4         ; 100000 OFF UNWANTED BITS.
2129 006452 001401      BEQ      IS           ; EXPECTED=ACTUAL?
2130 006454 104000      HLT                                ; BR IF ALL OK.
2131 006456 104400      IS:      SCOPE        ; PRIMARY REGISTER MASTER CLEAR ERROR
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                                ;
2144                                ; TEST 31
2145                                ; *****
2146 006462 012737 000031 001226  TST31: MOV      #31,TSTNO
2147 006470 012737 006554 001216      MOV      @TST32,NEXT
2148 006476 012737 000340 177776      MOV      #340,PS
2149 006504 013703 001364      MOV      @DQTCR,R3      ; LOCK OUT INTERRUPTS
2150 006510 012777 101772 172646      MOV      #101772,@DQTCR ; BUS ADDRESS OF
                                ; TRANSMITTER CONTROL REGISTER
                                ; LOAD TRANSMITTER CONTROL REGISTER
  
```

```

2151 006516 112777 000012 172644      MOVB  #12,JDQREG      ;SELECT MISCELLANEOUS REGISTER
2152 006524 052777 000040 172640      BIS   #BITS,JDQSEC   ;ISSUE MASTER CLEAR
2153 006532 005005                    CLR   R5              ;(R5)=EXPECTED 101772 IN
2154                                ;TRANSMITTER CONTROL REGISTER, 0
2155 006534 017704 172624      MOV   JDQTCR,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   IS              ;BR IF ALL OK.
2160 006550 104000                    HLT                                ;PRIMARY REGISTER MASTER CLEAR ERROR
2161 006552 104400      IS:    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      †ST32: MOV   #32,TSTNO
2171 006562 012737 006646 001216      MOV   #TST33,NEXT
2172 006570 012737 000340 177776      MOV   #340,PS
2173 006576 013703 001366      MOV   DQERR,R3      ;LOCK OUT INTERRUPTS
2174                                ;BUS ADDRESS OF
2175                                ;ERROR REGISTER
2176 006602 012777 107777 172556      MOV   #107777,JDQERR ;LOAD ERROR REGISTER
2177 006610 112777 000012 172552      MOVB  #12,JDQREG     ;SELECT MISCELLANEOUS REGISTER
2178 006616 052777 000040 172546      BIS   #BITS,JDQSEC   ;ISSUE MASTER CLEAR
2179 006624 005005                    CLR   R5              ;(R5)=EXPECTED 107777 IN
2180                                ;ERROR REGISTER, 0
2181 006626 017704 172534      MOV   JDQERR,R4      ;(R4)=ACTUAL 107777 IN
2182                                ;ERROR REGISTER
2183 006632 042704 060000      BIC   #60000,R4      ;60000 OFF UNWANTED BITS.
2184 006636 020504                    CMP   R5,R4          ;EXPECTED=ACTUAL?
2185 006640 001401                    BEQ   IS              ;BR IF ALL OK.
2186 006642 104000                    HLT                                ;PRIMARY REGISTER MASTER CLEAR ERROR
2187 006644 104400      IS:    SCOPE
2188
2189                                ;SYNC REGISTER MASTER CLEAR TEST
2190                                ;SET ALL READ/WRITE BITS IN SYNC REGISTER
2191                                ;ISSUE MASTER CLEAR
2192                                ;VERIFY THAT SYNC WAS CLEARED
2193
2194                                ; TEST 33
2195                                ;*****
2196 006646 012737 000033 001226      †ST33: MOV   #33,TSTNO
2197 006654 012737 006742 001216      MOV   #TST34,NEXT
2198 006662 012737 000340 177776      MOV   #340,PS
2199 006670 012703 000011      MOV   #11,R3         ;LOCK OUT INTERRUPTS
2200                                ;GET ADDRESS OF SECONDARY
2201                                ;REGISTER TO BE TESTED
2202 006674 110377 172470      MOVB  R3,JDQREG      ;SELECT CEREK
2203                                ;SECONDARY REGISTER
2204 006700 012777 177777 172464      MOV   #177777,JDQSEC ;LOAD SYNC REGISTER
2205 006706 112777 000012 172454      MOVB  #12,JDQREG     ;SELECT MISCELLANEOUS REGISTER
2206 006714 052777 000040 172450      BIS   #BITS,JDQSEC   ;ISSUE MASTER CLEAR
2207 006722 110377 172442      MOVB  R3,JDQREG      ;SELECT SYNC SECONDARY REGISTER
2208 006726 005005                    CLR   R5              ;(R5)=EXPECTED 177777 IN
    
```


H04

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 46
DZD08C.P11 16-DEC-76 13:23 VERIFICATION OF MASTER CLEAR LOGIC

2263 007122 110377 172242
2264 007126 005005
2265
2266 007130 017704 172236
2267
2268 007134 001401
2269 007136 104000
2270 007140 104400

MOV B R3, @DQREG
CLR R5
MOV @DQSEC, R4
BEQ 15
HLT
15: SCOPE

; SELECT BCC POLYNOMIAL SECONDARY REGISTER
;(R5)=EXPECTED 177777 IN
; BCC POLYNOMIAL REGISTER, 0
;(R4)=ACTUAL 177777 IN
; BCC POLYNOMIAL REGISTER
; BR IF ALL OK.
; SECONDARY REGISTER MASTER CLEAR ERROR

```

2271
2272
2273
2274
2275
2276
2277
2278 007142 005037 001234 .EOP: CLR LSTERR ;CLEAR LAST ERROR PC
2279 007146 005037 001312 CLR ERRFLG ;CLEAR ERROR FLAG
2280 007152 005237 001230 INC PASCNT ;UPDATE PASS COUNT
2281 007156 104402 TYPE
2282 007160 011372 MEPASS
2283 007162 104402 TYPE
2284 007164 011553 MCSRX
2285 007166 104411 CNVRT
2286 007170 007300 XCSR
2287 007172 104402 TYPE
2288 007174 011561 MVECX
2289 007176 104411 CNVRT
2290 007200 007306 XVEC
2291 007202 104402 TYPE
2292 007204 011567 MPASSX
2293 007206 104411 CNVRT
2294 007210 007314 XPASS
2295 007212 104402 TYPE
2296 007214 011600 MERRX
2297 007216 104411 CNVRT
2298 007220 007322 XERR
2299 007222 013777 001230 171752 MOV PASCNT,ALIGHTS ;DISPLAY PASS COUNT
2300 007230 005337 001276 DEC SAVNUM
2301 007234 001013 BNE RESTRT
2302 007236 013737 001504 001276 MOV DQNUM,SAVNUM
2303 007244 013701 000042 MOV #42,R1 ;CHECK FOR ACT-11 OR DDP
2304 007250 001405 BEQ RESTRT ;IF NOT, CONTINUE TESTING
2305 007252 000005
2306 007254 LOGICAL:
2307 007254 004711 JSR PC,(R1)
2308 007256 000240 NOP
2309 007260 000240 NOP
2310 007262 000240 NOP
2311 007264 104414 RESTRT: CKSWR
2312 007266 012737 002254 001214 MOV #TST1,RETURN
2313 007274 000137 002254 JMP TST1
2314 007300 000001 XCSR: 1
2315 007302 006 002 .BYTE 6,2
2316 007304 001360 DQRCR
2317 007306 000001 XVEC: 1
2318 007310 003 002 .BYTE 3,2
2319 007312 001350 DQRVEC
2320 007314 000001 XPASS: 1
2321 007316 006 002 .BYTE 6,2
2322 007320 001230 PASCNT
2323 007322 000001 XERR: 1
2324 007324 006 002 .BYTE 6,2
2325 007326 001232 ERRCNT
2326

```

J04

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 48
 DZD08C.P11 16-DEC-76 13:23 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

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 2$
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 .SCOPE1: 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 24(SP), MSG
2388 007602 062766 000002 000004 ADD #2, 4(SP)
2389 007610 104402 .INST1: TYPE
2390 007612 000000 .MSG: 0
2391 007614 012704 011744 MOV #INBUF, R4
2392 007620 012703 000007 MOV #7, R3
2393 007624 105777 171354 1S: TSTB @TKCSR
2394 007630 100375 BPL 1S
2395 007632 117714 171350 MOVB @TKDBR, (R4)
2396 007636 142714 000200 BICB #200, (R4)
2397 007642 121427 000025 CMPB (R4), #25 ;IS IT <IG>
2398 007646 001003 BNE 200S
2399 007650 104402 011332 TYPE, MCRLF
2400 007654 000755 BR .INST1
2401 007656 122427 000015 200S: CMPB (R4)+, #15
2402 007662 001423 BEQ INSTR2
2403 007664 117777 171316 171320 MOVB @TKDBR, @TPDBR
2404 007672 105777 171312 2S: TSTB @TPCSR
2405 007676 100375 BPL 2S
2406 007700 005303 DEC R3
2407 007702 001350 BNE 1S
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 MCM
2413 007716 005737 011152 TST @RDSW
2414 007722 001402 BEQ 400S
2415 007724 104402 011332 TYPE, MCRLF
2416 007730 000727 400S: 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 .PARAM: MOV R5, -(SP)
2424 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)+, ADCNT
2431 007774 010566 000004 MOV R5, 4(SP)
2432 010000 005005 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 1S: 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    2#RDSW          ;IS CKSWR ROUTINE BEING USED
2451 010066 001023          BNE    PARTI
2452 010070 104404          120$:  INSTER
2453 010072 000742          BR     PARAM1
2454
2455          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2456
2457 010074 020537 010146    LIMITS: CMP    R5,HILIM
2458 010100 101365          BHI    PARERR
2459 010102 020537 010144    CMP    R5,LOLIM
2460 010106 103762          BLO    PARERR
2461 010110 133705 010152    BITB   LOBITS,R5
2462 010114 001357          BNE    PARERR
2463
2464          ;STORE NUMBER AT SPECIFIED ADDRESS
2465
2466 010116 013704 010150          1$:  MOV    DEVADR,R4
2467 010122 010524          MOV    R5,(R4)+
2468 010124 062705 000002    ADD    #2,R5
2469 010130 105337 010153    DECB   ADCNT
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          ADCNT=LOBITS+1
2479
2480          ;SAVE PC OF TEST THAT FAILED AND RO-R5
2481
2482 010154 016637 000004 001274 .SAV05: MOV    4(SP),SAVPC
2483
2484          ;SAVE RO-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 RO-R5
    
```

```

249 010214 013700 001256
250 010220 013701 001260
251 010224 013702 001262
252 010230 013703 001264
253 010234 013704 001266
254 010240 013705 001270
255 010244 000002

256
257
258
259
260
261
262
263
264
265
266
267 010246 104402
268 010250 011332
269 010254 010046
270 010258 010146
271 010262 010346
272 010266 010446
273 010270 010546
274 010274 017601 00012
275 010278 013737 012006 001250
276 010282 062766 000002 000012
277 010304 012137 010466
278 010310 112137 010470
279 010314 112137 010471
280 010320 013137 010472
281 010324 013704 010472
282 010330 113705 010470
283 010334 012700 012006
284 010340 010403
285 010344 042703 177770
286 010348 062703 000060
287 010352 110320
288 010354 000241
289 010358 006004
290 010362 000241
291 010366 006004
292 010370 000241
293 010374 006004
294 010378 000241
295 010382 006004
296 010386 000241
297 010390 006004
298 010394 005305
299 010398 001362
300 010402 012703 012050
301 010406 114023
302 010410 105337 010470
303 010414 001374
304 010418 105737 010471
305 010422 001405
306 010426 112723 000040
307 010430 105337 010471
308 010434 001373
309 010438 105013
310 010442 104402
311 010446 012050
312 010450 005337 010466
313 010454 001322
314 010458 013737 001250 012006
315 010462 012605

```

```

.RES05: MOV SAVR0,R0
MOV SAVR1,R1
MOV SAVR2,R2
MOV SAVR3,R3
MOV SAVR4,R4
MOV SAVR5,R5
RTI

```

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

```

.CONVR: TYPE
MCRLF
.CNVRT: MOV R0,-(SP)
MOV R1,-(SP)
MOV R3,-(SP)
MOV R4,-(SP)
MOV R5,-(SP)
MOV #12(SP),R1
MOV TEMP,TEMP3
ADD #2,12(SP)
MOV (R1)+,WRDCNT
1S: MOVB (R1)+,CHRCNT
MOVB (R1)+,SPACNT
MOV #2(R1)+,BINWRD
2S: MOV BINWRD,R4
MOVB CHRCNT,R5
MOV #TEMP,R0
3S: MOV R4,R3
BIC #177770,R3
ADD #060,R3
MOVB R3,(R0)+
CLC
ROR R4
CLC
ROR R4
CLC
ROR R4
DEC R5
BNE 3S
MOV #MDATA,R3
4S: MOVB -(R0),(R3)+
DECB CHRCNT
BNE 4S
TSTB SPACNT
BEQ 6S
5S: MOVB #040,(R3)+
DECB SPACNT
BNE 5S
6S: CLRB (R3)
TYPE
MDATA
DEC WRDCNT
1S
MOV TEMP3,TEMP
MOV (SP)+,R5

```

```

2551 010454 012604      MOV      (SP)+,R4
2552 010456 012603      MOV      (SP)+,R3
2553 010460 012601      MOV      (SP)+,R1
2554 010462 012600      MOV      (SP)+,R0
2555 010464 000002      RTI
2556 010466 000000
2557 010470 000000
2558 010471 010471
2559 010472 000000
2560
2561
2562
2563
2564
2565 010474 011646      .TRPSR: MOV      (SP)-(SP)      ;GET PC OF RETURN
2566 010476 162716 000002      SUB      #2,(SP)          ;=PC OF TRAP
2567 010502 017616 000000      MOV      @2(SP),(SP)      ;GET TRAP
2568 010506 006316      TRPOK:  ASL      (SP)          ;MULTIPLY TRAP ARG BY 2
2569 010510 042716 177001      BIC      #177001,(SP)     ;CLEAR UNWANTED BITS
2570 010514 062716 001314      ADD      @.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
2571 010520 017616 000000      MOV      @2(SP),(SP)     ;SUBROUTINE ADDRESS
2572 010524 000136      JMP      @2(SP)+         ;GO TO SUBROUTINE
2573
2574
2575
2576
2576 010526 104414      ;ERROR HANDLER
2577 010530 032777 010000 170442 .HLT:  CKSWR
2578 010536 001406      BIT      #SW12,@SWR
2579 010540 105777 170444      BEQ      XBX
2580 010544 100003      TSTB    @TPCSR
2581 010546 112777 000207 170436      BPL      XBX
2582 010554 032777 020000 170416      MOVB    #207,@TPDBR
2583 010562 001074      XBX:    BIT      #SW13,@SWR
2584 010564 021637 001234      BNE      HALTS
2585 010570 001404      CMP      (SP),LSTERR
2586 010572 011637 001234      BEQ      1$
2587 010576 105037 001312      MOV      (SP),LSTERR
2588 010602 104406      CLRB    ERRFLG
2589 010604 011605      1$:    SAVOS
2590 010606 162716 000002      MOV      (SP),R5
2591 010612 011504      SUB      #2,R5
2592 010614 006304      MOV      (R5),R4
2593 010616 061504      ASL      R4
2594 010620 006304      ADD      (R5),R4
2595 010622 042704 177001      ASL      R4
2596 010626 062704 012760      BIC      #177001,R4
2597 010632 012437 010724      ADD      @.ERRTAB,R4
2598 010636 012437 010736      MOV      (R4)+,ERRMSG
2599 010642 011437 010750      MOV      (R4)+,DATAHD
2600 010646 105737 001312      MOV      (R4),DATABP
2601 010652 001403      TSTB    ERRFLG
2602 010654 005737 010750      BEQ      TYPMSG
2603 010660 001027      TST     DATABP
2604 010662 104402      BNE     TYPDAT
2605 010664 011611      TYPMSG: TYPE
2606 010666 104411      MTSTN
      CNVRT
    
```

B05

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 53
 DZD08C.P11 16-DEC-76 13:23 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

2607	010670	011050				XTSTN	
2608	010672	104402				TYPE	
2609	010674	011677				MERRPC	
2610	010676	104411				CONVRT	
2611	010700	011042				ERTABO	
2612	010702	104402				TYPE	
2613	010704	011332				MCRLF	
2614	010706	112737	177777	001312		MOV8	#-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	JSWR
2630	010760	100005				BPL	EXITER
2631	010762	010046				PUSHRO	
2632	010764	016600	000002			MOV	2(SP),R0
2633	010770	000000				HALT	
2634	010772	012600				POPPO	
2635	010774	104414				EXITER: CKSWR	
2636	010776	005237	001232			INC	ERRCNT
2637	011002	032777	000400	170170		BIT	#SW08,JSWR
2638	011010	001007				BNE	1\$
2639	011012	032777	002000	170160		BIT	#SW10,JSWR
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
2656	011064	000000				HALT	;SET UP FOR POWER UP TRAP
2657	011066	000777				BR	;HALT ON POWER DOWN NORMAL
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, SCNTG
2698 011230 005137 011152      .CNTLU: COM      @#RDSW
2699 011234 104402 011306      TYPE, SMSWR
2700 011248 104411 011274      CNVRT, SWREGC
2701 011244 104403 011315      INSTR, SMNEW
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      SCNTG: .ASCIZ <377>/†G/
2714 011306 051777 051127 020075      SMSWR: .ASCIZ <377>/SWR= /
2715 011314 000          SMNEW: .ASCIZ / NEW= /
2716 011315 040 047040 053505
2717 011322 020075 000
2718 011326          .EVEN

```

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 55
DZD08C.P11 16-DEC-76 13:23 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

2719	011326	020040	000077		MOM: .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	052122	040440	020124	
2725	011364	042524	052123	000040	
2726	011372	042777	042116	050040	MEPASS: .ASCIZ <377>/END PASS DZD08 /
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	011461	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

E05

DZD98 MACY11 27(1006) 22-DEC-76 11:05 PAGE 56
DZD98C.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					
2780					;BUFFERS FOR INPUT-OUTPUT
2781	011744	000000		INBUF:	0
2782		012006		.=.+40	
2783	012006	000000		TEMP:	0
2784		012050		.=.+40	
2785	012050	000000		MDATA:	0
2786		012112		.=.+40	

F05

DZDQB MACY11 27(1006) 22-DEC-76 11:05 PAGE 57
 DZDQBC.P11 16-DEC-76 13:23 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```

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 DHO: .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 DHO
2803 013000 013104 DT1
2804 013002 012264 EM3
2805 013004 012647 DHO
2806 013006 013062 DTO
2807 013010 012320 EM4
2808 013012 012647 DHO
2809 013014 013104 DT1
2810 013016 012357 EM5
2811 013020 012647 DHO
2812 013022 013104 DT1
2813 013024 012415 EM6
2814 013026 012647 DHO
2815 013030 013104 DT1
2816 013032 012451 EM7
2817 013034 012647 DHO
2818 013036 013104 DT1
2819 013040 012504 EM10
2820 013042 012716 DH1
2821 013044 013126 DT2
2822 013046 012550 EM11
2823 013050 012647 DHO
2824 013052 013104 DT1
2825 013054 012616 EM12
2826 013056 000000 0
2827 013060 000000 0
  
```

G05

DZD98 MACY11 27(1006) 22-DEC-76 11:05 PAGE 58
 DZD98C.P11 16-DEC-76 13:23 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

```

2828
2829                                     ;TABLE OF DATA POINTER FOR ERROR TYPEOUT
2830
2831 013062 000004          DTO: 4
2832 013064 003          .BYTE 3,7
2833 013066 001270          SAVR5
2834 013070 003          .BYTE 3,7
2835 013072 001266          SAVR4
2836 013074 006          .BYTE 6,4
2837 013076 001372          DQSEC
2838 013100 002          .BYTE 2,0
2839 013102 001264          SAVR3
2840 013104 000004          DT1: 4
2841 013106 006          .BYTE 6,4
2842 013110 001270          SAVR5
2843 013112 006          .BYTE 6,4
2844 013114 001266          SAVR4
2845 013116 006          .BYTE 6,4
2846 013120 001372          DQSEC
2847 013122 002          .BYTE 2,0
2848 013124 001262          SAVR2
2849 013126 000003          DT2: 3
2850 013130 006          .BYTE 6,4
2851 013132 001270          SAVR5
2852 013134 006          .BYTE 6,4
2853 013136 001266          SAVR4
2854 013140 006          .BYTE 6,4
2855 013142 001264          SAVR3
2856 00C001          .END
  
```


M05

DZD08 MACY11 27(1006) 22-DEC-76 11:05 PAGE 65
DZD08C.P11 16-DEC-76 13:23 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																	
	2601	2604#																		
TYPMSG 010662	922#	1019																		
VECMAP 000056	2516#	2547#	2556#																	
WRDCNT 010466	2616	2619#																		
WRKO.F 010726	2578	2580	2582#																	
XBX 010554	2286	2314#																		
XCSR 007300	2298	2323#																		
XERR 007322	1237	2767#																		
XHEAD 011704	2294	2320#																		
XPASS 007314	1242	2772#																		
XSTAT0 011732	2607	2648#																		
XTSTN 011050	2290	2317#																		
XVEC 007306	2697	2713#																		
SCNTG 011302	1#	1303	1304#	1363	1364#	1382	1383#	1406	1407#	1465	1467#	1498	1500#							
SE = 000037	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#																	
	2701	2716#																		
SMNEW 011315	2699	2714#																		
SMSWR 011306	1#	1299	1304#	1360	1364#	1379	1383#	1403	1407#	1462	1467#	1495	1500#							
SN = 000035	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#							
	1701#	2100#	2132#	2133#	2239#															
SN1 = 000035	1701#	2100#	2239#																	
SN2 = 000036	1701#	2100#	2239#																	
SN3 = 000026	1701#	2100#	2239#																	
SY = 000016	1#	1092#	1101	1103#	1105#	1107#	1109#	1111#	1113#	1115#	1117#	1119#	1121#							
	1123#	1125#	1127#	1129#																
	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#							

ADRTSX	1353#	1708	1745														
ADRTS1	1353#	1423															
COMMEN	1701#	2100#	2132#	2133#	2239#												
DOEND	1#	2271															
DOFRNT	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#																
SBEGIN	1#	1269															
SBUFFE	1#	2778															
SCATCH	1#	643															
SCLAVE	1#	1234															
SCONVR	1#	2503															
SEOP	1#	2271															
SGETFL	1#																
SGETPA	1#	1287															
SHEADE	1#	526															
SHLT	1#	2573															
SINSTR	1#	2382															
SINTNP	1#																
SMAINT	1#																
SMSG	1#	2719															
SPARAM	1#	2420															
SPFAIL	1#	2651															
SREG	1#	2479															
SSCOPE	1#	2326															
SSCOPI	1#	2350															
SSETFL	1#																
SSETVE	1#	901															
SSTART	1#	1189															
SSYMO	1#	543															
STRAPS	1#	1092															
STRPDE	1#	1101	1103	1105	1107	1109	1111	1113	1115	1117	1119	1121	1123	1125	1127		
STRPSR	1#	2560															
STSTN	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		
STYPE	1#	2358															
SVARIA	1#	1028															

C06

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.SEG/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)