

DQ11

BASIC LOGIC TEST PART 2
CZDQBD0

AH-8608D-MC
COPYRIGHT 74-78
FICHE 1 OF 1

JAN 1979
digital
MADE IN USA

This microfiche card contains a grid of frames on the left side, with the remainder of the card being blank. The frames contain technical data, likely test results or logic test outputs, organized in a structured format. The data is too small to read clearly but appears to be organized in columns and rows, possibly representing different test runs or parameters.

IDENTIFICATION

PRODUCT CODE: AC-8606D-MC
PRODUCT NAME: CZDQBD0 BL7 PRT 2
DATE: JUNE 1978
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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1974, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

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

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. CZDQA [REV] BASIS R/W TEST #1
2. CZDQB [REV] BASIC R/W TEST #2
3. CZDQC [REV] BASIC NPR AND INTERRUPT TEST
4. CZDQD [REV] RECEIVER TRANSMITTER EXERCISER TEST
5. CZDQE [REV] MISC. RX AND TX TESTS. PLUS BCC TESTS.
6. CZDQF [REV] CHARACTER DETECT TESTS.
7. CZDQH [REV] CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.

1. CZDQO [REV] ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE

1. CZDQG [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.

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 PROCEDURE

A. LOAD LOC. 200

B. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE
LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP
BY DQ11 TRIAL PROGRAM OR A PREVIOUSLY RUN DQ11 DIAGNOSTIC
THAT USED THE 'AUTO SIZING'.

****REFER TO SECTION 4.1 FOR SOFTWARE SWITCH REGISTER OPERATION
AND OPTIONS.****

NOTE:THE SOFTWARE SWITCH REGISTER IS LOCATED AT LOC.176
SOFTWARE DISPLAY REGISTER IS LOCATED AT LOC.174

C.THEN START

THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME
IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO
THE FOLLOWING:

'MAP OF DQ11 STATUS'
1400 160010
1402 152300
1404 160020
1406 150310

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

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

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

4.1 CONTROL SWITCH SETTINGS

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

CONTROL:

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

- 1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE "'NEW-'" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

SW 15	SET: HALT ON ERROR
SW 14	SET: LOOP ON CURRENT TEST
SW 13	SET: INHIBIT ERROR PRINT OUT
SW 12	SET: INHIBIT TYPE OUT/BELL ON ERROR.
SW 11	SET: INHIBIT ITERATIONS
SW 10	SET: ESCAPE TO NEXT TEST
SW 09	SET: LOOP WITH CURRENT DATA
SW 08	SET: CATCH ERROR AND LOOP ON IT
SW 07	SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.
SW 06	SET:
SW 05	SET:
SW 04	SET:
SW 03	SET:
SW 02	SET: LOCK ON SELECTED TEST
SW 01	SET: RESTART PROGRAM AT SELECTED TEST
SW 00	SET: RESELECT DQ11'S DESIRED ACTIVE.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DQ11'S DESIRED ACTIVE.
PLEASE NOTE THAT A MESSAGE IS TYPED
OUT FOR SWITCH REGISTER BEING EQUAL TO DQ11'S
ACTIVE. THIS MEANS IF THE SYSTEM HAS
FOUR DQ11S; BITS 00,01,02,03 WILL
BE SET IN LOC 'DQACTV'. USING THIS
SWITCH ALTERS THAT LOCATION; THEREFORE
IF FOUR DQ11S ARE IN THE SYSTEM
DO NOT SET SWITCHS GREATER THAN
SW 03 IN THE UP POSITION. THIS WOULD BE
A FATAL ERROR. DO NOT SELECT MORE ACTIVE
DQ11S THAN HAS BEEN GIVEN INFORMATION
ABOUT IN TRIAL PROGRAM.

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

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

SW 09 LOOP ON CURRENT DATA:
THIS SWITCH WILL ONLY WORK IF
CALL '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 OPERATOR IS REQUIRED TO TYPE A <^G> BEFORE DEPRESSING CONTINUE.
THE FOLLOWING WILL BE TYPED:
SWR=XXXXXX NEW (REFFR 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 AC-8606D-MC 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) 1500/000000000011111
MEANS THAT DQ11 NO. 00,01,02,03,04
WILL BE TESTED.

EXAMPLE:
(DQACTV) 1500/000000000010001
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 EXISTES

8.5.6 JUMPER ON END OF CABLE? ***NOTE:CZDQE ONLY***

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

8.5.7 ACTIVE ON FIRST NON-SYNC?

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

8.5.8 SET FOR ODD OR EVEN PARITY?

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

8.5.9 FINDING THE VECTOR.

THE PROGRAM SETS 'PRIMARY DONE', 'SECONDARY DONE', AND 'INTERUPT ENABLE' AND LOOKS FOR AN INTERUPT. IF IT INTERUPTS IT IS PICKED UP AND STORED AWAY. IF NO INTERUPT OCCURS 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

```
522 .ENABLE AMA
523
524 ;CZDQBD0/<377>/DQ11 STATIC LOGIC TEST-PART 2
525 ;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
526
527 ;REVISED 16-DEC-76 BY R. BLACK
528 ; A)SUPPORTS SOFTWARE SWITCH REGISTER
529 ; B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
530 ; BY <^G>.
531 ;STARTING PROCEDURE
532 ;LOAD PROGRAM
533 ;LOAD ADDRESS 000200
534 ;PRESS START
535 ;PROGRAM WILL TYPE "CZDQBD0/<377>/DQ11 STATIC LOGIC TEST-PART 2"
536 ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
537 ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
538 ;AND THEN RESUME TESTING
539
540
541 ;SWITCH REGISTER OPTIONS
542
543 100000 SW15=100000 ;-1,HALT ON ERROR
544 040000 SW14=40000 ;-1,LOOP ON CURRENT TEST
545 020000 SW13=20000 ;-1,INHIBIT ERROR TYPEOUT
546 010000 SW12=10000 ;=1,DELETE TYPEOUT/BELL ON ERROR.
547 004000 SW11=4000 ;-1,INHIBIT ITERATIONS
548 002000 SW10=2000 ;=1,ESCAPE TO NEXT TEST ON ERROR
549 001000 SW09=1000 ;-1,LOOP WITH CURRENT DATA
550 000400 SW08=400 ; 1,LOOP ON ERROR
551 000100 SW06=100
552 000040 SW05=40
553 000020 SW04=20
554 000010 SW03=10
555 000004 SW02=4 ;LOCK ON TEST SELECT
556 000002 SW01=2 ;RESTART PROGRAM AT SELECTED TEST
557 000001 SW00=1 ;RESELECT DQ11 DESIRED ACTIVE
558 ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

GENERAL DEFINITIONS AND EQUIVALENCIES

```
559
560
561           ;REGISTER DEFINITIONS
562
563           000000      R0=%0           ;GENERAL REGISTER
564           000001      R1=%1           ;GENERAL REGISTER
565           000002      R2=%2           ;GENERAL REGISTER
566           000003      R3=%3           ;GENERAL REGISTER
567           000004      R4=%4           ;GENERAL REGISTER
568           000005      R5=%5           ;GENERAL REGISTER
569           000006      SP=%6           ;PROCESSOR STACK POINTER
570           000007      PC=%7           ;PROGRAM COUNTER
571
572           ;LOCATION EQUIVALENCIES
573
574           177570      DSWR= 177570    ;HARDWARE SWITCH REGISTER LOC.
575           177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
576           177776      PS=177776     ;PROCESSOR STATUS WORD
577           001200      STACK=1200     ;START OF PROCESSOR STACK
578
579           ;INSTRUCTION DEFINITIONS
580
581           005746      PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
582           005726      POP1SP=5726    ;INCREMENT PROCESSOR STACK 1 WORD
583           010046      PUSHRO=10046    ;SAVE R0 ON STACK
584           012600      POPRO=12600    ;RESTORE R0 FROM STACK
585           024646      PUSH2SP=24646  ;DECREMENT STACK TWICE
586           022626      POP2SP=22626   ;INCREMENT STACK TWICE
587           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
588
589
590           100000      BIT15-100000
591           040000      BIT14=40000
592           020000      BIT13=20000
593           010000      BIT12=10000
594           004000      BIT11=4000
595           002000      BIT10=2000
596           001000      BIT9=1000
597           000400      BIT8=400
598           000200      BIT7=200
599           000100      BIT6=100
600           000040      BIT5=40
601           000020      BIT4=20
602           000010      BIT3=10
603           000004      BIT2=4
604           000002      BIT1=2
605           000001      BIT0 1
606
607
608           ;DQ11 OPTIONAL DEFINITIONS
609
610           002000      ABBIT 2000
611           004000      ACTBIT-4000
612           010000      BABIT=10000
613           020000      BBBIT-20000
614           040000      JUMBIT-40000
```

GENERAL DEFINITIONS AND EQUIVALENCIES

615 001000 ODDBIT=1000
616 100000 SYNBIT=100000

:DQ11 SECONDARY REGISTER DEFINATIONS

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

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```
640          :TRAPCATCAEP FOR ILLEGAL INTERRUPTS
641          000000          .=0
642          :STANDARD INTERRUPT VECTORS
643
644          000024          .=24
645 000024 011056          .PFAIL          :POWER FAIL HANDLER
646 000026 000340          340          :SERVICE AT LEVEL 7
647 000030 010526          .HLT          :ERROR HANDLER
648 000032 000340          340          :SERVICE AT LEVEL 7
649 000034 010474          .TRPSRV          :GENERAL HANDLER DISPATCH SERVICE
650 000036 000340          340          :SERVICE AT LEVEL 7
651          000046          .=46
652 000046 007254          LOGICAL          :ACT HOOKS
653          000052          .=52
654 000052 000000          .WORD 0
655          :THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
656          :TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
657          :FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
658          :NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
659          :EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
660          :TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
661          000056          .=56
662
663          000056          VECMAP:
664 000056 010120          1$: MOV R1,(R0)+          :START FILLING THE VECTOR AREA
665 000060 012721 000004          MOV #4,(R1)+          :WITH +2; IOT (4)
666 000064 022021          CMP (R0)+,(R1)+          :UPDATE THE POINTERS
667 000066 020127 001000          CMP R1,#1000          :IS ALL FLOATING VECTOR AREA DONE
668 000072 101771          BLOS 1$          :BR IF NOT ALL DONE
669 000074 012737 000146 000020          MOV #4$,@#20          :SET FOR IOT TRAP BY DQ11
670 000102 013737 001500 001244          MOV DQACTV,TEMP1          :GET THE ACTIVE DQ11 S
671 000110 006037 001244          2$: ROR TEMP1          :ARE YOU ACTIVE.. DQ11
672 000114 103023          BCC 5$          :IF CARRY CLEAR.. NO MORE DQ11S
673 000116 005037 177776          CLR PS          :CLEAR PS
674 000122 005722          TST (R2)+          :PUT POINTER TO STATUS TABLE
675 000124 012772 000340 177776          MOV #340,@-2(R2)          :TRY AND SET PRI/SEC DONE AND IE
676 000132 105200          INCB R0          :DELAY.....
677 000134 001376          BNE -2          :.....DELAY
678 000136 112712 000300          MOVB #300,(R2)          :NO INTERRUPT ASSUME 300 FIX IN TEST C
679 000142 005722          3$: TST (R2)+          :UPDATE POINTERS
680 000144 000761          BR 2$          :GO DO IT AGAIN
681 000146 051612          4$: BIS (SP),(R2)          :ENTERD BY IOT TRAP BY DQ11
682 000150 042712 000007          BIC #7,(R2)          :CLEAR UNWANTED BITS
683 000154 022626          CMP (SP)+,(SP)+          :POP IOT JUNK OFF STACK
684 000156 012716 000142          MOV #3$,(SP)          :SET RETURN PC ON STACK
685 000162 000002          RTI          :GC HOME
686 000164 000207          5$: RTS PC          :ALL SIZING IS DONE
687
688          :****SOFTWARE SWITCH REGISTER****
689          -174
690 000174 000000          DISPREG: 0          :SOFTWARE DISPLAY REGISTER
691 000176 000000          SWREG: 0          :SOFTWARE SWITCH REGISTER
692
693          :PROGRAM START
694
695          000200          .=200
```

```

696 000200 000137 001512          JMP      .START          ;GO TO START OF PROGRAM
697
698                                . =220
699 000220 012702 001400          CSRMAP: MOV     #1400,R2    ;CLEAR ALL STATUS TABLE
700 000224 005022                    CLR     (R2)+           ;DO CLEAR
701 000226 022702 001512          CMP     #1512,R2       ;ALL TABLE DONE
702 000232 001374                    BNE     .-6             ;BR IF MORE TO GO
703 000234 005037 001504          CLR     DQNUM          ;SET NUMBER OF DQ11S TO 0
704 000240 012702 001400          MOV     #1400,R2       ;SET TABLE POINTER
705 000244 012701 160000          MOV     #160000,R1     ;GET FIRST FLOATING ADDRESS
706 000250 012737 000614 000004   MOV     #5$,@#4        ;SET FOR TIME OUT TRAP--NO DEVICE--
707 000256 112761 000012 000005   1$:  MOV     #12,5(R1)    ;TRY AND SEL MISC REGISTER
708 000264 005061 000006          CLR     6(R1)         ;TRY AND CLEAR MISC REG
709 000270 012711 010000          MOV     #10000,(R1)    ;TRY AND SET RX ACTIVE
710 000274 022761 030000 000006   CMP     #30000,6(R1)  ;LOOK FOR SYNC 1 AND SYNC 2
711 000302 001071                    BNE     2$             ;THIS IS NOT A DQ11 IF I BRANCH
712 000304 010122                    MOV     R1,(R2)+       ;NOW THIS IS A DQ11 --STORE CSR
713 000306 052712 100000          BIS     #SYNBIT,(R2)  ;SET FOR TWO SYNC CHARS
714 000312 005011                    CLR     (R1)           ;CLEAR DQ ACTIVE BIT
715 000314 112761 000010 000005   MOV     #10,5(R1)     ;SEL CHAR DET REGISTER
716 000322 012761 177777 000006   MOV     #-1,6(R1)     ;WRITE INTO CHAR DET REG
717 000330 005761 000006          TST     6(R1)         ;WAS THE REGISTER WRITTEN?
718 000334 001402                    BEQ     .+6            ;APPARENTLY NO BB OPTION.
719 000336 052712 020000          BIS     #BBBIT,(R2)  ;SET FOR BB OPTION
720 000342 112761 000017 000005   MOV     #17,5(R1)     ;SEL POLYNO. REGISTER
721 000350 012761 177777 000006   MOV     #-1,6(R1)     ;WRITE POLYNO.REGISTER
722 000356 005761 000006          TST     6(R1)         ;WAS REG WRITTEN??
723 000362 001402                    BEQ     .+6            ;BR IF NO AB OPTION
724 000364 052712 002000          BIS     #ABBIT,(R2)  ;SET FOR AB OPTION
725 000370 012761 001400 000002   MOV     #1400,2(R1)   ;TRY TO SET .DTR. .RS.
726 000376 032761 001400 000002   BIT     #1400,2(R1)   ;DID ANY OF THEM SET
727 000404 001402                    BEQ     .+6            ;BR IF NO BA OPTION
728 000406 052712 010000          BIS     #BABIT,(R2)  ;SET FOR BA OPTION
729 000412 032761 030000 000002   BIT     #30000,2(R1) ;DID .CS. .CO. SET
730 000420 001402                    BEQ     .+6            ;BR IF NO JUMPER
731 000422 052712 040000          BIS     #JUMBIT,(R2) ;SET FOR JUMPER
732 000426 052712 004000          BIS     #ACTBIT,(R2) ;SET FOR ACTIVE ON FIRST NON-SYNC
733 000432 052712 001000          BIS     #ODDBIT,(R2) ;SET FOR ODD VRC.....
734 000436 005722                    TST     (R2)+         ;POP POINTER
735 000440 005011                    CLR     (R1)           ;CLEAR RCSR
736 000442 005061 000002          CLR     2(R1)         ;CLEAR TCSR
737 000446 005061 000002          CLR     2(R1)         ;CLEAR AGAIN
738 000452 005061 000004          CLR     4(R1)         ;CLEAR ERROR REG
739 000456 005061 000006          CLR     6(R1)         ;CLEAR SEC REG
740 000462 005237 001504          INC     DQNUM         ;UPDATE NUMBER OF DQ11S
741 000466 062701 000010 2$:  ADD     #10,R1         ;UPDATE CSR POINTER BY 10 (8)
742 000472 022701 164000          CMP     #164000,R1    ;HAVE ALL FLOATING ADDRESSES BEEN CHECKED??
743 000476 001267                    BNE     1$            ;BR IF NOT ALL DONE
744 000500 005037 001500          CLR     DQACTV        ;ZERO ACTIVE DQ11S
745 000504 005737 001504          TST     DQNUM         ;WERE ANY DQ11S FOUND
746 000510 001434                    BEQ     4$            ;HEY BUDDY. NO DQ11S FOUND IN SYSTEM
747 000512 013701 001504          MOV     DQNUM,R1      ;SAVE NUMBER OF DQ11S
748 000516 010137 001276          MOV     R1,SAVNUM     ;SAVE NUMBER FOR ACT11
749 000522 000241 3$:  CLC                    ;CLEAR CARRY
750 000524 006137 001500          ROL     DQACTV        ;ACTIVE ADDRESS
751 000530 005237 001500          INC     DQACTV        ;SET BIT 0

```



```

752 000534 005301          DEC      R1          ;DEC NUMBER OF DQ11S
753 000536 001371          BNE      3$          ;BR IF MORE TO GO
754 000540 012737 000006 000004  MOV      #6,@#4      ;RESET TIME OUT VECTOR
755 000546 013737 001500 001502  MOV      DQACTV,SAVACT ;SAVE ACTIVE
756 000554 012737 000340 000022  MOV      #340,@#22   ;SET IOT TRAP PRIO: TO 7
757 000562 012702 001400          MOV      #1400,R2    ;SET TABLE POINTER
758 000566 012700 000300          MOV      #300,R0     ;SET VECTOR START
759 000572 012701 000302          MOV      #302,R1     ;SET VECTOR+2 START
760 000576 000137 000056          JMP      VECMAP      ;GO FIND THE VECTORS
761 000602 104402          4$:      TYPE          ;TYPE MESSAGE
762 000604 011417          MERR2          ;I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
763 000606 005000          CLR      R0          ;
764 000610 000000          HALT          ;HOW CAN I TEST NO DQ11S
765 000612 000776          BR      -2          ;DON'T LET OPR HIT CONT. SW
766 000614 012716 000466 5$:      MOV      #2$,(SP)   ;ENTERED BY TIME OUT TRAP
767 000620 000002          RTI              ;GO HOME.
768
769
770
771 001000 001000          .=1000
772 001006 005377 055103 050504  MTITLE: .ASCIZ <377><12>/CZDQBD0/<377>/DQ11 STATIC LOGIC TEST-PART 2/<377>
773 001014 042102 177460 050504
774 001014 030461 051440 040524
775 001022 044524 020103 047514
776 001030 044507 020103 042524
777 001036 052123 050055 051101
778 001044 020124 177462 000
779
780 001200          .-1200
781          ;INDIRECT POINTERS
782 001200 177570          SWR:      177570      ;SWITCH REGISTER POINTER
783 001202 177570          LIGHTS: 177570      ;DISPLAY REGISTER POINTER
784 001204 177560          TKCSR:   177560      ;TELETYPE KEYBOARD CONTROL REGISTER
785 001206 177562          TKDBR:   177562      ;TELETYPE KEYBOARD DATA BUFFER
786 001210 177564          TPCSR:   177564      ;TELEPRINTER CONTROL REGISTER
787 001212 177566          TPDBR:   177566      ;TELEPRINTER DATA BUFFER
788
789          ;PROGRAM CONTROL PARAMETERS
790
791 001214 000000          RETURN: 0          ;SCOPE ADDRESS FOR LOOP ON TEST
792 001216 000000          NEXT:   0          ;ADDRESS OF NEXT TEST TO BE EXECUTED
793 001220 000000          LOCK:   0          ;ADDRESS FOR LOCK ON CURRENT DATA
794 001222 000003          ICOUNT: 3        ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
795 001224 000000          LPCNT:  0          ;NUMBER OF ITERATIONS COMPLETED
796 001226 000000          TSTNO:  0          ;NUMBER OF TEST IN PROGRESS
797 001230 000000          PASCNT: 0          ;NUMBER OF PASSES COMPLETED
798 001232 000000          ERRCNT: 0          ;TOTAL NUMBER OF ERRORS
799 001234 000000          LSTERR: 0          ;PC OF LAST ERROR CALL
800
801          ;PROGRAM VARIABLES
802
803 001236 000000          CHAR1:  0
804 001240 000000          CHAR2:  0
805 001242 000000          CHAR3:  0
806 001244 000000          TEMP1:  0          ;TEMPORARY STORAGE
807 001246 000000          TEMP2:  0          ;TEMPORARY STORAGE

```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

808	001250	000000	TEMP3:	0	:TEMPORARY STORAGE
809	001252	000000	TEMP4:	0	:TEMPORARY STORAGE
810	001254	000000	TEMP5:	0	:TEMPORARY STORAGE
811	001256	000000	SAVR0:	0	:R0 STORAGE
812	001260	000000	SAVR1:	0	:R1 STORAGE
813	001262	000000	SAVR2:	0	:R2 STORAGE
814	001264	000000	SAVR3:	0	:R3 STORAGE
815	001266	000000	SAVR4:	0	:R4 STORAGE
816	001270	000000	SAVR5:	0	:R5 STORAGE
817	001272	000000	SAVSP:	0	:STACK POINTER STORAGE
818	001274	000000	SAVPC:	0	:PROGRAM COUNTER STORAGE
819	001276	000000	SAVNUM:	0	
820	001300	000001	CREAM:	.BLKW 1	
821	001302	000000	RUNFLG:	0	
822	001304	000000	RUN:	0	
823	001306	000000	RUNCNT:	0	

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
824
825
826
827 001310 000
828 001311 000
829 001312 000
830 001313 000
831 000000
832
833
834
835
836
837
838
839 001314
840 104400
841 001314 007330
842 104401
843 001316 007442
844 104402
845 001320 007462
846 104403
847 001322 007570
848 104404
849 001324 007706
850 104405
851 001326 007740
852 104406
853 001330 010154
854 104407
855 001332 010214
856 104410
857 001334 010246
858 104411
859 001336 010252
860 104412
861 001340 012114
862 104413
863 001342 012112
864 104414
865 001344 011154
866 104415
867 001346 011230
868
869
870
871
872
873
874 001350 000000
875 001352 000000
876 001354 000000
877 001356 000000
878 001360 000000
879 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
$Y=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
SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
.SCOPE1
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
SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
.SAV05
RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
.RES05
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 ISSUF MASTER CLEAR
.MSTCLR
MEMCLR=TRAP+13 ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
.MEMCLR
CKSWR=TRAP+14 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
.CKSWR
CNTLU=TRAP+15 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
.CNTLU

:*****
:*****

;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
DQRVEC: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
DQRLVL: 0 ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
DQTRVEC: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
DQTLVL: 0 ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
DQRCSR: 0 ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
DQRCSH: 0 ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER
```

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
880 001364 000000      DQTCR: 0          ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
881 001366 000000      DQERR: 0         ; POINTER TO DQ11 ERROR REGISTER
882 001370 000000      DQREG: 0        ; POINTER TO HIGH BYTE OF ERROR REGISTER
883 001372 000000      DQSEC: 0       ; POINTER TO DQ11 SECONDARY REGISTER
884 001374 000000      DQSECH: 0      ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
885
886
887
888                      ; DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
889
890                      . = 1400
891 001400 000001      DQCR00: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
892 001402 000001      DQST00: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
893 001404 000001      DQCR01: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
894 001406 000001      DQST01: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
895 001410 000001      DQCR02: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
896 001412 000001      DQST02: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
897 001414 000001      DQCR03: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
898 001416 000001      DQST03: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
899 001420 000001      DQCR04: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
900 001422 000001      DQST04: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
901 001424 000001      DQCR05: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
902 001426 000001      DQST05: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
903 001430 000001      DQCR06: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
904 001432 000001      DQST06: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
905 001434 000001      DQCR07: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
906 001436 000001      DQST07: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
907 001440 000001      DQCR10: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
908 001442 000001      DQST10: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
909 001444 000001      DQCR11: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
910 001446 000001      DQST11: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
911 001450 000001      DQCR12: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
912 001452 000001      DQST12: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
913 001454 000001      DQCR13: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
914 001456 000001      DQST13: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
915 001460 000001      DQCR14: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
916 001462 000001      DQST14: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
917 001464 000001      DQCR15: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
918 001466 000001      DQST15: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
919 001470 000001      DQCR16: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
920 001472 000001      DQST16: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
921 001474 000001      DQCR17: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
922 001476 000001      DQST17: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
923 001500 000001      DQACTV: .BLKW 1   ; HOLD ACTIVE BITS FOR TESTING
924 001502 000001      SAVACT: .BLKW 1   ; SAVE NUMBER OF ACTIVE DQ11S
925 001504 000001      DQNUM: .BLKW 1   ; OCTAL NUMBER OF TOTAL NUMBER OD DQ11S
926 001506 000001      DQCSR: .BLKW 1   ; CSR OF DQ11 UNDER TEST
927 001510 000001      DQSTAT: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
928
929                      ; PROGRAM INITIALIZATION
930                      ; LOCK OUT INTERRUPTS
931                      ; SET UP PROCESSOR STACK
932                      ; SET UP POWER FAIL VECTOR
933                      ; CLEAR PROGRAM CONTROL FLAGS AND COUNTS
934                      ; TYPE TITLE MESSAGE
935
```

```

936 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
937 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
938 001524 012737 011056 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
939 001532 013737 001504 001276 MOV DONUM,SAVNUM
940 001540 105037 001311 CLR STFLG ;CLEAR START FLAG
941 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
942 001550 105037 001312 CLR ERRFLG ;CLEAR ERROR FLAG
943 001554 005037 001302 CLR RUNFLG
944 001560 012737 001400 001300 MOV #1400,CREAM
945 001566 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
946 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
947 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
948 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
949 ;TESTING STARTS
950 001612 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
951 001620 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
952 001626 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
953 001632 013746 000004 MOV @#4,-(SP)
954 001636 012737 001656 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
955 001644 022777 177777 177326 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
956 001652 001402 BEQ 65$
957 001654 000407 BR 66$
958 001656 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
959 001660 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
960 001666 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
961 001674 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
962 001700 012637 000006 MOV (SP)+,@#6
963 001704 005737 000042 TST @#42 ;UNDER MONITOR
964 001710 001014 BNE 67$
965 ;:*****THE NEXT 4 LINES OF CODE MOVED TO SOLVE PR#2757 (JUNE 78)*****
966 001712 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED?
967 001716 001035 BNE 12$ ;IF YES, BR
968 001720 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
969 001724 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND INIT
970 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
971 001736 001001 BNE 67$
972 001740 104415 CNTLU
973 001742 105777 177232 67$: TSTB @SWR
974 001746 100402 BMI .+6
975 001750 004737 000220 JSR PC,CSRMAP
976 001754 104402 011704 TYPE ,XHEAD
977 001760 012737 001400 001244 MOV #1400,TEMP1
978 001766 017737 177252 001246 MOV @TEMP1,TEMP2
979 001774 001406 BEQ .+16
980 001776 104410 CONVRT
981 002000 011732 XSTATQ
982 002002 062737 000002 001244 ADD #2,TEMP1
983 002010 000766 BR .-22
984 002012 032777 000001 177160 12$: BIT #SW00,@SWR
985 002020 001424 BEQ 1$
986 002022 104402 TYPE
987 002024 011625 MNEW
988 002026 005000 CLR R0
989 002030 000000 HALT
990 002032 104414 LKSWR
991 002034 027737 177140 001502 CMP @SWR,SAVACT.

```

```

992 002042 101404      BLOS 11$
993 002044 104402      TYPE
994 002046 011466      MERR3
995 002050 000000      HALT
996 002052 000776      BR -2
997 002054 017737 177120 001500 11$: MOV @SWR,DQACTV
998 002062 013700 001500      MOV DQACTV,R0
999 002066 000000      HALT
1000 002070 104414      CKSWR
1001 002072 012700 000300 1$: MOV #300,R0
1002 002076 012701 000302      MOV #302,R1
1003 002102 010120      2$: MOV R1,(R0)+
1004 002104 005021      CLR (R1)+
1005 002106 022021      CMP (R0)+,(R1)+
1006 002110 022700 001000      CMP #1000,R0
1007 002114 001372      BNE 2$
1008
1009      ;TEST START AND RESTART
1010
1011 002116 012737 000340 177776 .BEGIN: MOV #340,PS      ;LOCK OUT INTERRUPTS
1012 002124 012706 001200      MOV #STACK,SP      ;SET UP STACK
1013 002130 005737 000042      TST @#42          ;IS PROGRAM UNDER MONITOR CONTROL
1014 002134 001040      BNE 3$
1015 002136 104414      CKSWR          ;CHECK FOR <^G>
1016 002140 032777 000004 177032      BIT #BIT2,@SWR    ;CHECK FOR LOCK ON TEST
1017 002146 001411      BEQ 1$
1018 002150 104402 011524      TYPE .MLOCK
1019 002154 012737 000240 007340      MOV #NOP,TTST
1020 002162 012737 000240 007342      MOV #NOP,TTST+2   ;SET UP TO LOCK
1021 002170 000406      BR 2$
1022 002172 013737 007436 007340 1$: MOV BRW,TTST
1023 002200 013737 007440 007342      MOV BRX,TTST+2    ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1024 002206 032777 000002 176764 2$: BIT #SW01,@SWR   ;IF SW01=1, GET STARTING PC
1025 002214 001410      BEQ 3$
1026 002216 104403      INSTR
1027 002220 011512      MTSTPC
1028 002222 104405      PARAM
1029 002224 002254      TST1
1030 002226 007046      TLAST
1031 002230 001214      #RETURN
1032 002232 001      .BYTE 1
1033 002233 001      .BYTE 1
1034 002234 000403      BR 4$
1035 002236 012737 002254 001214 3$: MOV #TST1,RETURN ;START AT TEST 1
1036 002244 104402 011414 4$: TYPE #R        ;TYPE R
1037 002250 000177 176740      JMP @RETURN      ;START TESTING
1038      : TEST 1
1039      :*****
1040 002254 012737 000001 001226 TST1: MOV #1,TSTNO
1041 002262 012737 002644 001214      MOV #TST2,RETURN
1042 002270 012737 002644 001216      MOV #TST2,NEXT
1043 002276 105737 001302      TSTB RUNFLG      ;IS THIS MY FIRST TIME HERE?
1044 002302 001010      BNE 1$          ;BR IF FLAG IS SET
1045 002304 012737 000001 001304      MOV #BIT0,RUN    ;SET RUN POINTER.
1046 002312 012737 000020 001306      MOV #16,RUNCNT   ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1047 002320 105137 001302      COMB RUNFLG     ;SET RUN FLAG
    
```

PROGRAM INITIALIZATION AND START UP.

```
1048 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1049 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1050 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1051 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
1052 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1053 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1054 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1055 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1056 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1057 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1058 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1059 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1060 002406 000746 BR 1$ ;KEEP ON TESTING.
1061 002410 104402 2$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1062 002412 011417 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1063 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!!
1064 002416 000776 BR .-2 ;STICK HERE ON CONT.
1065 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1066 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1067 002426 017737 176646 001506 MOV @CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1068 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1069 002442 017737 176632 001510 MOV @CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1070 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1071 002456 013737 001506 001360 MOV DQCSR,DQRCSR
1072 002464 013737 001510 001350 MOV DQSTAT,DQRVEC
1073 002472 042737 177007 001350 BIC #177007,DQRVEC
1074 002500 013737 001350 001352 MOV DQRVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1075 002506 062737 000002 001352 ADD #2,DQRLVL
1076 002514 013737 001352 001354 MOV DQRLVL,DQTVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1077 002522 062737 000002 001354 ADD #2,DQTVEC
1078 002530 013737 001354 001356 MOV DQTVEC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1079 002536 062737 000002 001356 ADD #2,DQTLVL
1080 002544 013737 001360 001362 MOV DQRCSR,DQRCSH
1081 002552 005237 001362 INC DQRCSH ;GENERATE ADDRESS OF HIGH BYTE
1082 002556 013737 001360 001364 MOV DQRCSR,DQTCSR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1083 002564 062737 000002 001364 ADD #2,DQTCSR
1084 002572 013737 001364 001366 MOV DQTCSR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1085 002600 062737 000002 001366 ADD #2,DQERR
1086 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1087 002614 005237 001370 INC DQREG
1088 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1089 002626 005237 001372 INC DQSEC
1090 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1091 002640 005237 001374 INC DQSECH
```

```
1092                                     :MEMORY EXTENSION WRITE ENABLE TEST
1093                                     :SET MEMORY EXTENSION WRITE ENABLE
1094                                     :ADDRESS BUS ADDRESS MEMORY
1095                                     :EXPECTED RESULTS
1096                                     :      MEMORY EXTENSION WRITE ENABLE IS CLEARED
1097
1098 002644                               OPT2X:
1099                                     : TEST 2
1100                                     :*****
1101 002644 012737 000002 001226          TST2:  MOV    #2,TSTNO
1102 002652 012737 002706 001216          MOV    #TST3,NEXT
1103 002660 112777 000020 176502          MOVB   #BIT4,@DQREG                :SET WRITE ENABLE,
1104                                     :SELECT RECEIVER BUS ADDRESS
1105                                     :MEMORY
1106 002666 105077 176500                CLR    @DQSEC                       :CLEAR MEMORY
1107 002672 132777 000020 176470          BITB   #BIT4,@DQREG                :WAS WRITE ENABLE CLEARED
1108 002700 001401                        BEQ    1$                          :BR IF WRITE ENABLE CLEARED
1109 002702 104012                        HLT    12                          :WRITE ENABLE NOT CLEARED
1110 002704 104400          1$:          SCOPE
```


1111
1112 :MEMORY EXTENTION BITS READ/WRITE TEST
1113 :READ MEMORY EXTENSION BITS
1114 :WITH WRITE ENABLE BITS CLEARED, ATTEMPT
1115 :TO CHANGE MEMORY EXTENSION BITS
1116 :VERITY THAT NO CHANGE OCCURS
1117

: TEST 3

1118 :*****
1119 :TST3: MOV #3,TSTNO
1120 002706 012737 000003 001226 MOV #TST4,NEXT
1121 002714 012737 002774 001216 MOV DQREG,R3 :SET FOR ERROR PRINTOUT
1122 002722 013703 001370 MOV #37,@DQREG :CLEAR MEM EXT AND WRITE ENABLE
1123 002726 142777 000037 176434 BICB #37,@DQREG :SAVE REGISTER
1124 002734 117705 176430 MOVB @DQREG,R5 :STORE REG
1125 002740 010500 MOV R5,R0 :CHANE CONTENTS
1126 002742 105100 COMB R0 :CLEAR UNWANTED BITS
1127 002744 142700 000237 BICB #237,R0 :WRITE REGISTER
1128 002750 110077 176414 MOVB R0,@DQREG :READ REGISTER
1129 002754 117704 176410 MOVB @DQREG,R4 :CLEAR UNWANTED BITS
1130 002760 142704 000037 BICB #37,R4 :DID IT CHANGE
1131 002764 020504 CMP R5,R4 :BR IF NO CHANGE OCCURED
1132 002766 001401 BEQ 1\$:REPORT ERROR
1133 002770 104000 HLT 0 :SCOPE THIS TEST
1134 002772 104400 1\$: SCOPE

1135
1136 :MEMORY EXTENSION READ/WRITE TEST
1137 :READ MEMORY EXTENSION BITS
1138 :WITH WRITE ENABLE SET, ATTEMPT TO CHANGE
1139 :MEMORY EXTENSION BITS
1140 :VERIFY THAT MEMORY EXTENSION BITS WERE CHANGED
1141

: TEST 4

1142 :*****
1143 :TST4: MOV #4,TSTNO
1144 002774 012737 000004 001226 MOV #TST5,NEXT
1145 003002 012737 003222 001216 MOV DQREG,R3 :SET FOR ERROR REPORT
1146 003010 013703 001370 MOV #37,@DQREG :CLEAR UNWANTED BITS.
1147 003014 142777 000037 176346 BICB #37,@DQREG :READ DQREG FOR DATA
1148 003022 117700 176342 MOVB @DQREG,R0 :DO A COMPLETE CHANGE OF DATA
1149 003026 105100 COMB R0 :CLEAR WRITE ENABLE AND GARBAGE
1150 003030 142700 000237 BICB #237,R0 :STORE DATA.
1151 003034 010005 MOV R0,R5 :SET WRITE ENABLE
1152 003036 152705 000020 BISB #BIT4,R5 :WRITE DQREG WITH NEW DATA
1153 003042 110577 176322 MOVB R5,@DQREG :REFERENCE SEL 6 (DQSEC)
1154 003046 005077 176320 CLR @DQSEC :READ DQREG
1155 003052 117704 176312 MOVB @DQREG,R4 :CLEAR UNWANTED BITS.
1156 003056 042704 000037 BIC #37,R4 :CLEAR WRITE ENABLE
1157 003062 042705 000020 BIC #20,R5 :ARE THEY EQUAL?
1158 003066 020504 CMP R5,R4 :BR IF GOOD
1159 003070 001401 BEQ 1\$:MEMORY EXTENSION READ/WRITE ERROR
1160 003072 104000 HLT 0 :SCOPE THE TEST.
1161 003074 104400 1\$: SCOPE

```

1162
1163 ;BUS ADDRESS EXTENSION MEMORY TEST
1164 ;LOAD EACH BUS ADDRESS MEMORY LOCATION WITH A DIFFERENT
1165 ;NUMBER (0-3)
1166 ;VERIFY THAT EACH LOCATION WAS ADDRESSED PROPERLY
1167
1168 003076 005003 CLR R3 ;FIRST MEMORY ADDRESS =0
1169 003100 052701 000020 BIS #BIT4,R1 ;SET WRITE ENABLE
1170 003104 112777 000020 176256 MOVB #BIT4,@DQREG ;SET MEMORY EXTENSION WRITE ENABLE
1171 003112 150377 176252 BISB R3,@DQREG ;SELECT LOCATION TO BE TESTED
1172 003116 110177 176246 MOVB R1,@DQREG ;LOAD DATA INTO MEMORY EXTENSION BITS
1173 003122 005077 176244 CLR @DQSEC ;REFERENCE SEL 6 (DQSEC)
1174 003126 062703 000002 ADD #2,R3 ;NEXT LOCATION TO BE TESTED
1175 003132 062705 000040 ADD #40,R5 ;DATA TO BE LOADED
1176 003136 005300 DEC R0 ;CONTINUE IF ALL LOCATIONS NOT LOADED
1177 003140 001355 BNE 1$ ;BR IF MORE TO GO
1178 003142 005005 CLR R5 ;INITIAL DATA = 0
1179 003144 012700 000004 MOV #4,R0 ;4 LOCATIONS TO BE TESTED
1180 003150 005003 CLR R3 ;FIRST LOCATION
1181 003152 142777 000037 176210 2$: BICB #37,@DQREG ;REINITIALIZE ADDRESS
1182 003160 150377 176204 BISB R3,@DQREG ;RESELECT SECONDARY ADDRESS.
1183 003164 117704 176200 MOVB @DQREG,R4 ;READ MEMORY EXTENSION BITS
1184 003170 142704 000037 BICB #37,R4 ;CLEAR UNWANTED BITS
1185 003174 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
1186 003176 001401 BEQ 3$
1187 003200 104001 HLT 1 ;BUS ADDRESS EXTENSION MEMORY ERROR
1188 003202 104401 3$: SCOP1 ;CHECK FOR LOOP ON CUREENT ADDRESS
1189 003204 062703 000002 ADD #2,R3 ;UPDATE SELECT ADDRESS
1190 003210 062705 000040 ADD #40,R5 ;UPDATE EXPECTED ATA
1191 003214 005300 DEC R0 ;CONTINUE IF NOT DONE
1192 003216 001355 BNE 2$ ;BR IF MORE TO CHECK.
1193 003220 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1194
1195 ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1196 ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1197 ;TO 0
1198 ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1199 ;COUNT MEMORY WAS SET TO 0
1200
1201 ; TEST 5
1202 ;*****
1203 003222 012737 000005 001226 TST5: MOV #5,TSTNO
1204 003230 012737 003336 001216 MOV #TST6,NEXT
1205 003236 012737 003306 001220 MOV #2$,LOCK
1206 003244 012705 000000 MOV #0,R5 ;DATA TO BE LOADED INTO MEMORY
1207 003250 013703 001372 MOV DQSEC,R3 ;ADDRESS OF SECONDARY REGISTER
1208 003254 005002 CLR R2 ;FIRST MEMORY LOCATION TO BF TESTED
1209 003256 012700 000010 MOV #10,R0 ;# OF MEMORY LOCATIONS TO BE TESTED
1210 003262 110277 176102 1$: MOVB R2,@DQREG ;SELECT MEMORY LOCATION TO BE TESTED
1211 003266 010577 176100 MOV R5,@DQSEC ;LOAD DATA INTO MEMORY
1212 003272 005202 INC R2 ;GO TO NEXT LOCATION
1213 003274 005300 DEC R0 ;MORE TO GO??
1214 003276 001371 BNE 1$ ;BR IF NOT DONE
1215 003300 005002 CLR R2 ;SET UP TO TEST
1216 003302 012700 000010 MOV #10,R0 ;10 (OCTAL) LOCATIONS
1217 003306 110277 176056 2$: MOVB R2,@DQREG ;SELECT LOCATION TO BE TESTED
    
```

BUS ADDRESS AND CHARACTER COUNT MEMORY TESTS

```
1218 003312 017704 176054      MOV    @DQSEC,R4      ;READ DATA
1219 003316 020504      CMP    R5,R4         ;COMPARE TO EXPECTED DATA
1220 003320 001401      BEQ    3$           ;BR IF GOOD
1221 003322 104002      HLT    2             ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1222 003324 104401      3$:  SCOP1          ;CHECK FOR FREEZE ON CURRENT LOCATION
1223 003326 005202      INC    R2           ;GO TO NEXT LOCATION
1224 003330 005300      DEC    R0           ;DONE YET?
1225 003332 001365      BNE    2$           ;BR IF FURTHER CHECKING NEEDED
1226 003334 104400      4$:  SCOPE
```

```
1227
1228      ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1229      ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1230      ;TO 177777
1231      ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1232      ;COUNT MEMORY WAS SET TO 177777
1233
```

TEST 6

```
1234 ;*****
1235 ;*****
1236 003336 012737 000006 001226 TST6: MOV    #6,TSTNO
1237 003344 012737 003452 001216      MOV    #TST7,NEXT
1238 003352 012737 003422 001220      MOV    #2$,LOCK
1239 003360 012705 177777      MOV    #177777,R5      ;DATA TO BE LOADED INTO MEMORY
1240 003364 013703 001372      MOV    DQSEC,R3       ;ADDRESS OF SECONDARY REGISTER
1241 003370 005002      CLR    R2            ;FIRST MEMORY LOCATION TO BE TESTED
1242 003372 012700 000010      MOV    #10,R0        ;# OF MEMORY LOCATIONS TO BE TESTED
1243 003376 110277 175766      1$:  MOVB   R2,@DQREG   ;SELECT MEMORY LOCATION TO BE TESTED
1244 003402 010577 175764      MOV    R5,@DQSEC     ;LOAD DATA INTO MEMORY
1245 003406 005202      INC    R2            ;GO TO NEXT LOCATION
1246 003410 005300      DEC    R0            ;MORE TO GO??
1247 003412 001371      BNE    1$           ;BR IF NOT DONE
1248 003414 005002      CLR    R2            ;SET UP TO TEST
1249 003416 012700 000010      MOV    #10,R0        ;10 (OCTAL) LOCATIONS
1250 003422 110277 175742      2$:  MOVB   R2,@DQREG   ;SELECT LOCATION TO BE TESTED
1251 003426 017704 175740      MOV    @DQSEC,R4     ;READ DATA
1252 003432 020504      CMP    R5,R4         ;COMPARE TO EXPECTED DATA
1253 003434 001401      BEQ    3$           ;BR IF GOOD
1254 003436 104002      HLT    2             ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1255 003440 104401      3$:  SCOP1          ;CHECK FOR FREEZE ON CURRENT LOCATION
1256 003442 005202      INC    R2           ;GO TO NEXT LOCATION
1257 003444 005300      DEC    R0           ;DONE YET?
1258 003446 001365      BNE    2$           ;BR IF FURTHER CHECKING NEEDED
1259 003450 104400      4$:  SCOPE
```

```
1260 ;BUS ADDRESS AND CHARACTER COUNT MEMORY TEST
1261 ;SET EACH LOCATION IN BUS ADDRESS AND CHARACTER COUNT MEMORY
1262 ;TO 125252
1263 ;VERIFY THAT EACH LOCATION IN BUS ADDRESS AND CHARACTER
1264 ;COUNT MEMORY WAS SET TO 125252
1265
```

TEST 7

```
1266 ;*****
1267 ;*****
1268
1269 003452 012737 000007 001226 TST7: MOV    #7,TSTNO
1270 003460 012737 003566 001216      MOV    #TST10,NEXT
1271 003466 012737 003536 001220      MOV    #2$,LOCK
1272 003474 012705 125252      MOV    #125252,R5     ;DATA TO BE LOADED INTO MEMORY
1273 003500 013703 001372      MOV    DQSEC,R3       ;ADDRESS OF SECONDARY REGISTER
```

BUS ADDRESS AND CHARACTER COUNT MEMORY TESTS

```
1274 003504 005002          CLR      R2          ;FIRST MEMORY LOCATION TO BE TESTED
1275 003506 012700 000010    MOV      #10,R0      ;# OF MEMORY LOCATIONS TO BE TESTED
1276 003512 110277 175652    1$:     MOVB     R2,@DQREG ;SELECT MEMORY LOCATION TO BE TESTED
1277 003516 010577 175650    MOV      R5,@DQSEC   ;LOAD DATA INTO MEMORY
1278 003522 005202          INC      R2          ;GO TO NEXT LOCATION
1279 003524 005300          DEC      R0          ;MORE TO GO??
1280 003526 001371          BNE     1$          ;BR IF NOT DONE
1281 003530 005002          CLR      R2          ;SET UP TO TEST
1282 003532 012700 000010    MOV      #10,R0      ;10 (OCTAL) LOCATIONS
1283 003536 110277 175626    2$:     MOVB     R2,@DQREG ;SELECT LOCATION TO BE TESTED
1284 003542 017704 175624    MOV      @DQSEC,R4   ;READ DATA
1285 003546 020504          CMP      R5,R4       ;COMPARE TO EXPECTED DATA
1286 003550 001401          BEQ     3$          ;BR IF GOOD
1287 003552 104002          HLT     2           ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1288 003554 104401          3$:     SCOP1        ;CHECK FOR FREEZE ON CURRENT LOCATION
1289 003556 005202          INC      R2          ;GO TO NEXT LOCATION
1290 003560 005300          DEC      R0          ;DONE YET?
1291 003562 001365          BNE     2$          ;BR IF FURTHER CHECKING NEEDED
1292 003564 104400          4$:     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

:*****

```
1301
1302 003566 012737 000010 001226 TST10: MOV      #10,TSTNO
1303 003574 012737 003702 001216    MOV      #TST11,NEXT
1304 003602 012737 003652 001220    MOV      #2$,LOCK
1305 003610 012705 052525          MOV      #52525,R5   ;DATA TO BE LOADED INTO MEMORY
1306 003614 013703 001372          MOV      DQSEC,R3    ;ADDRESS OF SECONDARY REGISTER
1307 003620 005002          CLR      R2          ;FIRST MEMORY LOCATION TO BE TESTED
1308 003622 012700 000010    MOV      #10,R0      ;# OF MEMORY LOCATIONS TO BE TESTED
1309 003626 110277 175536    1$:     MOVB     R2,@DQREG ;SELECT MEMORY LOCATION TO BE TESTED
1310 003632 010577 175534    MOV      R5,@DQSEC   ;LOAD DATA INTO MEMORY
1311 003636 005202          INC      R2          ;GO TO NEXT LOCATION
1312 003640 005300          DEC      R0          ;MORE TO GO??
1313 003642 001371          BNE     1$          ;BR IF NOT DONE
1314 003644 005002          CLR      R2          ;SET UP TO TEST
1315 003646 012700 000010    MOV      #10,R0      ;10 (OCTAL) LOCATIONS
1316 003652 110277 175512    2$:     MOVB     R2,@DQREG ;SELECT LOCATION TO BE TESTED
1317 003656 017704 175510    MOV      @DQSEC,R4   ;READ DATA
1318 003662 020504          CMP      R5,R4       ;COMPARE TO EXPECTED DATA
1319 003664 001401          BEQ     3$          ;BR IF GOOD
1320 003666 104002          HLT     2           ;CHAR COUNT-BUS ADDRESS MEMORY ERROR
1321 003670 104401          3$:     SCOP1        ;CHECK FOR FREEZE ON CURRENT LOCATION
1322 003672 005202          INC      R2          ;GO TO NEXT LOCATION
1323 003674 005300          DEC      R0          ;DONE YET?
1324 003676 001365          BNE     2$          ;BR IF FURTHER CHECKING NEEDED
1325 003700 104400          4$:     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
```

1326
1327
1328
1329

BUS ADDRESS MEMORY EXTENSION DATA TESTS

```
1330  
1331 ; TEST 11  
1332 :*****  
1333 003702 012737 000011 001226 TST11: MOV #11,TSTNO  
1334 003710 012737 004052 001216 MOV #TST12,NEXT  
1335 003716 012737 004006 001220 MOV #2$,LOCK  
1336 003724 012705 000040 MOV #40,R5  
1337 003730 012700 000004 MOV #4,R0  
1338 003734 005003 CLR R3 ;4 LOCATIONS WILL BE ADDRESSED  
1339 ;FIRST MEMORY ADDRESS =0  
1340 003736 010501 1$: MOV R5,R1 ;(RECEIVER PRIMARY BUS ADDRESS)  
1341 003740 050301 BIS R3,R1 ;LOAD 'DATA'  
1342 003742 052701 000020 BIS #BIT4,R1 ;LOAD SECONDARY ADDRESS  
1343 003746 112777 000020 175414 MOVB #BIT4,@DQREG ;SET WRITE ENABLE.  
1344 003754 150377 175410 BISB R3,@DQREG ;SET MEMORY EXTENSION WRITE ENABLE  
1345 003760 110177 175404 MOVB R1,@DQREG ;SELECT LOCATION TO BE TESTED  
1346 003764 005077 175402 CLR @DQSEC ;LOAD DATA INTO MEMORY EXTENSION BITS  
1347 003770 062703 000002 ADD #2,R3 ;REFERENCE SEL 6 (DQSEC)  
1348 003774 005300 DEC R0 ;NEXT LOCATION TO BE TESTED  
1349 003776 001357 BNE 1$ ;CONTINUE IF ALL LOCATIONS NOT LOADED  
1350 004000 012700 000004 MOV #4,R0 ;4 LOCATIONS TO BE TESTED  
1351 004004 005003 CLR R3 ;FIRST LOCATION  
1352 004006 142777 000037 175354 2$: BICB #37,@DQREG ;REINITIALIZE ADDRESS  
1353 004014 150377 175350 BISB R3,@DQREG ;RESELECT SECONDARY ADDRESS  
1354 004020 117704 175344 MOVB @DQREG,R4 ;READ DQREG  
1355 004024 142704 000037 BICB #37,R4 ;CLEAR UNWANTED BITS  
1356 004030 020504 CMP R5,R4 ;IS DATA CORRECT?  
1357 004032 001401 BEQ 3$ ;BR IF OK.  
1358 004034 104003 HLT 3 ;BUS ADDRESS EXTENSION MEMORY ERROR  
1359 004036 104401 3$: SCOP1 ;FREEZE ON CURRENT DATA? (SW09=1)  
1360 004040 062703 000002 ADD #2,R3 ;UPDATE POINTER TO NEXT ADDRESS  
1361 004044 005300 DEC R0 ;ALL ADDRESS DONE?  
1362 004046 001357 BNE 2$ ;BR IF NOT DONE.  
1363 004050 104400 4$: SCOPE ;SCOPE THE TEST.  
1364  
1365 ;BUS ADDRESS MEMORY EXTENSION DATA TEST  
1366 ;LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 100  
1367 ;VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA  
1368  
1369 ; TEST 12  
1370 :*****  
1371 004052 012737 000012 001226 TST12: MOV #12,TSTNO  
1372 004060 012737 004222 001216 MOV #TST13,NEXT  
1373 004066 012737 004156 001220 MOV #2$,LOCK  
1374 004074 012705 000100 MOV #100,R5  
1375 004100 012700 000004 MOV #4,R0  
1376 004104 005003 CLR R3 ;4 LOCATIONS WILL BE ADDRESSED  
1377 ;FIRST MEMORY ADDRESS =0  
1378 004106 010501 1$: MOV R5,R1 ;(RECEIVER PRIMARY BUS ADDRESS)  
1379 004110 050301 BIS R3,R1 ;LOAD 'DATA'  
1380 004112 052701 000020 BIS #BIT4,R1 ;LOAD SECONDARY ADDRESS  
1381 004116 112777 000020 175244 MOVB #BIT4,@DQREG ;SET WRITE ENABLE.  
1382 004124 150377 175240 BISB R3,@DQREG ;SET MEMORY EXTENSION WRITE ENABLE  
1383 004130 110177 175234 MOVB R1,@DQREG ;SELECT LOCATION TO BE TESTED  
1384 004134 005077 175232 CLR @DQSEC ;LOAD DATA INTO MEMORY EXTENSION BITS  
1385 004140 062703 000002 ADD #2,R3 ;REFERENCE SEL 6 (DQSEC)  
 ;NEXT LOCATION TO BE TESTED
```

```

1386 004144 005300          DEC      R0          ;CONTINUE IF ALL LOCATIONS NOT LOADED
1387 004146 001357          BNE     1$          ;
1388 004150 012700 000004    MOV     #4,R0      ;4 LOCATIONS TO BE TESTED
1389 004154 005003          CLR     R3         ;FIRST LOCATION
1390 004156 142777 000037 175204 2$: BICB   #37,@DQREG ;REINITIALIZE ADDRESS
1391 004164 150377 175200    BISB   R3,@DQREG  ;RESELECT SECONDARY ADDRESS
1392 004170 117704 175174    MOVB   @DQREG,R4  ;READ DQREG
1393 004174 142704 000037    BICB   #37,R4     ;CLEAR UNWANTED BITS
1394 004200 020504          CMP     R5,R4     ;IS DATA CORRECT?
1395 004202 001401          BEQ    3$         ;BR IF OK.
1396 004204 104003          HLT    3          ;BUS ADDRESS EXTENSION MEMORY ERROR
1397 004206 104401          SCOPE 3$        ;FREEZE ON CURRENT DATA? (SW09=1)
1398 004210 062703 000002    ADD    #2,R3     ;UPDATE POINTER TO NEXT ADDRESS
1399 004214 005300          DEC    R0         ;ALL ADDRESS DONE?
1400 004216 001357          BNE    2$        ;BR IF NOT DONE.
1401 004220 104400          SCOPE 4$        ;SCOPE THE TEST.
1402
1403          ;BUS ADDRESS MEMORY EXTENSION DATA TEST
1404          ;LOAD EACH LOCATION IN BUS ADDRESS EXTENTION MEMORY WITH 140
1405          ;VERIFY THAT EACH ADDRESS WAS LOADED WITH THE CORRECT DATA
1406
1407          ; TEST 13
1408          ;*****
1409 004222 012737 000013 001226 TST13: MOV   #13,TSTNO
1410 004230 012737 004372 001216    MOV   #CHKBB1,NEXT
1411 004236 012737 004326 001220    MOV   #2$,LOCK
1412 004244 012705 000140          MOV   #140,R5
1413 004250 012700 000004          MOV   #4,R0      ;4 LOCATIONS WILL BE ADDRESSED
1414 004254 005003          CLR   R3         ;FIRST MEMORY ADDRESS =0
1415          ;(RECEIVER PRIMARY BUS ADDRESS)
1416 004256 010501          1$: MOV   R5,R1    ;LOAD 'DATA'
1417 004260 050301          BIS   R3,R1     ;LOAD SECONDARY ADDRESS
1418 004264 052701 000020    BIS   #BIT4,R1  ;SET WRITE ENABLE.
1419 004266 112777 000020 175074    MOVB  #BIT4,@DQREG ;SET MEMORY EXTENSION WRITE ENABLE
1420 004274 150377 175070    BISB  R3,@DQREG  ;SELECT LOCATION TO BE TESTED
1421 004300 110177 175064    MOVB  R1,@DQREG  ;LOAD DATA INTO MEMORY EXTENSION BITS
1422 004304 005077 175062    CLR   @DQSEC    ;REFERENCE SEL 6 (DQSEC)
1423 004310 062703 000002    ADD   #2,R3     ;NEXT LOCATION TO BE TESTED
1424 004314 005300          DEC   R0        ;CONTINUE IF ALL LOCATIONS NOT LOADED
1425 004316 001357          BNE   1$        ;
1426 004320 012700 000004    MOV   #4,R0     ;4 LOCATIONS TO BE TESTED
1427 004324 005003          CLR   R3         ;FIRST LOCATION
1428 004326 142777 000037 175034 2$: BICB   #37,@DQREG ;REINITIALIZE ADDRESS
1429 004334 150377 175030    BISB  R3,@DQREG  ;RESELECT SECONDARY ADDRESS
1430 004340 117704 175024    MOVB  @DQREG,R4  ;READ DQREG
1431 004344 142704 000037    BICB  #37,R4     ;CLEAR UNWANTED BITS
1432 004350 020504          CMP   R5,R4     ;IS DATA CORRECT?
1433 004352 001401          BEQ   3$        ;BR IF OK.
1434 004354 104003          HLT   3         ;BUS ADDRESS EXTENSION MEMORY ERROR
1435 004356 104401          SCOPE 3$        ;FREEZE ON CURRENT DATA? (SW09 1)
1436 004360 062703 000002    ADD   #2,R3     ;UPDATE POINTER TO NEXT ADDRESS
1437 004364 005300          DEC   R0        ;ALL ADDRESS DONE?
1438 004366 001357          BNE   2$        ;BR IF NOT DONE.
1439 004370 104400          SCOPE 4$        ;SCOPE THE TEST.
1440
1441          ;IF CHARACTER DETECT OPTION IS INSTALLED
    
```

CHARACTER AND SEQUENCE MEMORY TESTS

1442 :TESTS 14 THRU 25 WILL BE EXECUTED

1443
1444 004372 032737 020000 001510 CHKBB1: BIT #BBBIT,DQSTAT
1445 004400 001002 BNE TST14
1446 004402 000137 006206 JMP TST26

1447
1448 :CHARACTER MEMORY ADDRESSING TEST
1449 :LOAD EACH LOCATION IN CHARACTER MEMORY WITH ITS
1450 :ADDRESS (DUPLICATED EVERY 4 BITS)
1451 :VERIFY THAT EACH LOCATION WAS ADDRESSED
1452

1453 : TEST 14

1454 :*****
1455 004406 012737 000014 001226 TST14: MOV #14,TSTNO
1456 004414 012737 004546 001216 MOV #TST15,NEXT
1457 004422 012737 004512 001220 MOV #2\$,LOCK
1458 004430 005005 CLR R5 ;CLEAR DATA TO BE LOADED
1459 004432 012700 000020 MOV #16.,R0 ;16(DECIMAL) REGISTERS WILL BE TESTED
1460 004436 005002 CLR R2 ;FIRST ADDRESS =0
1461 004440 012703 000010 MOV #10,R3 ;CHARACTER MEMORY WILL BE SELECTED
1462 004444 112777 000010 174716 MOVB #10,@DQREG ;SELECT CHARACTER MEMORY
1463 004452 110277 174704 1\$: MOVB R2,@DQRCSH ;SELECT LOCATION IN CHARACTER MEMORY
1464 004456 010577 174710 MOV R5,@DQSEC ;LOAD DATA
1465 004462 005202 INC R2 ;UPDATE ADDRESS
1466 004464 062705 010421 ADD #10421,R5 ;UPDATE DATA
1467 004470 005300 DEC R0 ;CONTINUE IF NOT DONE
1468 004472 001367 BNE 1\$;
1469 004474 005005 CLR R5 ;CLEAR EXPECTED DATA
1470 004476 012700 000020 MOV #16.,R0 ;16 (DECIMAL) LOCATIONS WILL BE TESTED
1471 004502 005002 CLR R2 ;FIRST LOCATION =0
1472 004504 112777 000010 174656 MOVB #10,@DQREG ;SELECT CHARACTER REGISTER
1473 004512 110277 174644 2\$: MOVB R2,@DQRCSH ;SELECT ADDRESS TO BE TESTED
1474 004516 017704 174650 MOV @DQSEC,R4 ;READ ADDRESS TO BE TESTED
1475 004522 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
1476 004524 001401 BEQ 3\$;BR IF GOOD.
1477 004526 104004 HLT 4 ;CHARACTER MEMORY ADDRESS ERROR
1478 004530 104401 3\$: SCOP1 ;CHECK FOR LOOP ON CUREENT ADDRESS
1479 004532 005202 INC R2 ;UPDATE ADDRESS TO BE TESTED
1480 004534 062705 010421 ADD #10421,R5 ;UPDATE EXPECTED DATA
1481 004540 005300 DEC R0 ;CONTINUE IF NOT DONE
1482 004542 001363 BNE 2\$;
1483 004544 104400 4\$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1484

1485 :SEQUENCE MEMORY ADDRESSING TEST
1486 :LOAD EACH LOCATION IN SEQUENCE MEMORY WITH ITS
1487 :ADDRESS (DUPLICATED EVERY 4 BITS)
1488 :VERIFY THAT EACH LOCATION WAS ADDRESSED
1489

1490 : TEST 15

1491 :*****
1492 004546 012737 000015 001226 TST15: MOV #15,TSTNO
1493 004554 012737 004706 001216 MOV #TST16,NEXT
1494 004562 012737 004652 001220 MOV #2\$,LOCK
1495 004570 005005 CLR R5 ;CLEAR DATA TO BE LOADED
1496 004572 012700 000020 MOV #16.,R0 ;16(DECIMAL) REGISTERS WILL BE TESTED
1497 004576 005002 CLR R2 ;FIRST ADDRESS =0

CHARACTER AND SEQUENCE MEMORY TESTS

```
1498 004600 012703 000014      MOV      #14,R3      ;SEQUENCE MEMORY WILL BE SELECTED
1499 004604 112777 000014 174556  MOVB     #14,@DQREG  ;SELECT SEQUENCE MEMORY
1500 004612 110277 174544      1$:  MOVB     R2,@DQRCSH ;SELECT LOCATION IN SEQUENCE MEMORY
1501 004616 010577 174550      MOV      R5,@DQSEC  ;LOAD DATA
1502 004622 005202          INC      R2         ;UPDATE ADDRESS
1503 004624 062705 010421      ADD     #10421,R5   ;UPDATE DATA
1504 004630 005300          DEC     R0         ;CONTINUE IF NOT DONE
1505 004632 001367          BNE     1$         ;
1506 004634 005005          CLR     R5         ;CLEAR EXPECTED DATA
1507 004636 012700 000020      MOV     #16.,R0    ;16 (DECIMAL) LOCATIONS WILL BE TESTED
1508 004642 005002          CLR     R2         ;FIRST LOCATION =0
1509 004644 112777 000014 174516  MOVB     #14,@DQREG ;SELECT SEQUENCE REGISTER
1510 004652 110277 174504      2$:  MOVB     R2,@DQRCSH ;SELECT ADDRESS TO BE TESTED
1511 004656 017704 174510      MOV     @DQSEC,R4  ;READ ADDRESS TO BE TESTED
1512 004662 020504          CMP     R5,R4     ;ARE EXPECTED AND RECEIVED DATA THE SAME
1513 004664 001401          BEQ     3$         ;BR IF GOOD
1514 004666 104005          HLT     5         ;SEQUENCE MEMORY ADDRESS ERROR
1515 004670 104401      3$:  SCOP1          ;CHECK FOR LOOP ON CUREENT ADDRESS
1516 004672 005202          INC     R2         ;UPDATE ADDRESS TO BE TESTED
1517 004674 062705 010421      ADD     #10421,R5  ;UPDATE EXPECTED DATA
1518 004700 005300          DEC     R0         ;CONTINUE IF NOT DONE
1519 004702 001363          BNE     2$         ;
1520 004704 104400      4$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
```

```
1521
1522          ;CHARACTER MEMORY TEST
1523          ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 000000
1524          ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
1525          ;LOADED WITH 000000
```

```
1526
1527          ; TEST 16
1528          ;*****
```

```
1529 004706 012737 000016 001226  TST16: MOV     #16,TSTNO
1530 004714 012737 005036 001216  MOV     #TST17,NEXT
1531 004722 012737 005006 001220  MOV     #2$,LOCK
1532 004730 012705 000000          MOV     #000000,R5 ;LOAD THE DATA INTO R5
1533 004734 012700 000020          MOV     #16.,R0    ;SET COUNT FOR 16
1534 004740 005002          CLR     R2         ;CLEAR REGISTER POINTER
1535 004742 012703 000010          MOV     #10,R3     ;GET REGISTER IN R3
1536 004746 112777 000010 174414  MOVB     #10,@DQREG ;SELECT THE REGISTER
1537 004754 110277 174402      1$:  MOVB     R2,@DQRCSH ;SELECT THE CHARACTER DET REG
1538 004760 010577 174406      MOV     R5,@DQSEC  ;LOAD THE DATA
1539 004764 005202          INC     R2         ;UPDATE THE POINTER
1540 004766 005300          DEC     R0         ;ALL DONE YET?
1541 004770 001371          BNE     1$         ;BR IF NOT DONE
1542 004772 012700 000020          MOV     #16.,R0    ;SET FOR 16 REGISTERS TO BE TESTED
1543 004776 005002          CLR     R2         ;ZERO REG POINTER
1544 005000 112777 000010 174362  MOVB     #10,@DQREG ;SELECT THE REGISTER
1545 005006 110277 174350      2$:  MOVB     R2,@DQRCSH ;SELECT THE CHARACTER DET REGISTER
1546 005012 017704 174354      MOV     @DQSEC,R4  ;READ THE DATA
1547 005016 020504          CMP     R5,R4     ;WAS IT LOADED CORRECTLY?
1548 005020 001401          BEQ     3$         ;BR IF DATA OK
1549 005022 104006          HLT     6         ;INCORRECT DATA REPORT ERROR
1550 005024 104401      3$:  SCOP1          ;CHECK FOR FREEZE ON DATA
1551 005026 005202          INC     R2         ;UPDATE POINTER
1552 005030 005300          DEC     R0         ;CHECK FOR ALL DONE
1553 005032 001365          BNF     2$         ;BR IF NOT DONE
```


CHARACTER AND SEQUENCE MEMORY TESTS

```
1554 005034 104400 4$: SCOPE ;SCOPE THIS TEST
1555
1556 ;CHARACTER MEMORY TEST
1557 ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 177777
1558 ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
1559 ;LOADED WITH 177777
1560
1561 ; TEST 17
1562 ;*****
1563 005036 012737 000017 001226 TST17: MOV #17,TSTNO
1564 005044 012737 005166 001216 MOV #TST20,NEXT
1565 005052 012737 005136 001220 MOV #2$,LOCK
1566 005060 012705 177777 MOV #177777,R5 ;LOAD THE DATA INTO R5
1567 005064 012700 000020 MOV #16.,R0 ;SET COUNT FOR 16
1568 005070 005002 CLR R2 ;CLEAR REGISTER POINTER
1569 005072 012703 000010 MOV #10,R3 ;GET REGISTER IN R3
1570 005076 112777 000010 174264 MOVB #10,@DQREG ;SELECT THE REGISTER
1571 005104 110277 174252 1$: MOVB R2,@DQRCSH ;SELECT THE CHARACTER DET REG
1572 005110 010577 174256 MOV R5,@DQSEC ;LOAD THE DATA
1573 005114 005202 INC R2 ;UPDATE THE POINTER
1574 005116 005300 DEC R0 ;ALL DONE YET?
1575 005120 001371 BNE 1$ ;BR IF NOT DONE
1576 005122 012700 000020 MOV #16.,R0 ;SET FOR 16 REGISTERS TO BE TESTED
1577 005126 005002 CLR R2 ;ZERO REG POINTER
1578 005130 112777 000010 174232 MOVB #10,@DQREG ;SELECT THE REGISTER
1579 005136 110277 174220 2$: MOVB R2,@DQRCSH ;SELECT THE CHARACTER DET REGISTER
1580 005142 017704 174224 MOV @DQSEC,R4 ;READ THE DATA
1581 005146 020504 CMP R5,R4 ;WAS IT LOADED CORRECTLY?
1582 005150 001401 BEQ 3$ ;BR IF DATA OK
1583 005152 104006 HLT 6 ;INCORRECT DATA REPORT ERROR
1584 005154 104401 3$: SCOP1 ;CHECK FOR FREEZE ON DATA
1585 005156 005202 INC R2 ;UPDATE POINTER
1586 005160 005300 DEC R0 ;CHECK FOR ALL DONE
1587 005162 001365 BNE 2$ ;BR IF NOT DONE
1588 005164 104400 4$: SCOPE ;SCOPE THIS TEST
1589
1590 ;CHARACTER MEMORY TEST
1591 ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 125252
1592 ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS
1593 ;LOADED WITH 125252
1594
1595 ; TEST 20
1596 ;*****
1597 005166 012737 000020 001226 TST20: MOV #20,TSTNO
1598 005174 012737 005316 001216 MOV #TST21,NEXT
1599 005202 012737 005266 001220 MOV #2$,LOCK
1600 005210 012705 125252 MOV #125252,R5 ;LOAD THE DATA INTO R5
1601 005214 012700 000020 MOV #16.,R0 ;SET COUNT FOR 16
1602 005220 005002 CLR R2 ;CLEAR REGISTER POINTER
1603 005222 012703 000010 MOV #10,R3 ;GET REGISTER IN R3
1604 005226 112777 000010 174134 MOVB #10,@DQREG ;SELECT THE REGISTER
1605 005234 110277 174122 1$: MOVB R2,@DQRCSH ;SELECT THE CHARACTER DET REG
1606 005240 010577 174126 MOV R5,@DQSEC ;LOAD THE DATA
1607 005244 005202 INC R2 ;UPDATE THE POINTER
1608 005246 005300 DEC R0 ;ALL DONE YET?
1609 005250 001371 BNE 1$ ;BR IF NOT DONE
```

```
1610 005252 012700 000020          MOV    #16.,R0          ;SET FOR 16 REGISTERS TO BE TESTED
1611 005256 005002          CLR    R2              ;ZERO REG POINTER
1612 005260 112777 000010 174102    MOVB   #10,@DQREG      ;SELECT THE REGISTER
1613 005266 110277 174070 2$:    MOVB   R2,@DQRC5H      ;SELECT THE CHARACTER DET REGISTER
1614 005272 017704 174074          MOV    @DQSEC,R4       ;READ THE DATA
1615 005276 020504          CMP    R5,R4          ;WAS IT LOADED CORRECTLY?
1616 005300 001401          BEQ    3$             ;BR IF DATA OK
1617 005302 104006          HLT    6              ;INCORRECT DATA REPORT ERROR
1618 005304 104401 3$:    SCOP1          ;CHECK FOR FREEZE ON DATA
1619 005306 005202          INC    R2             ;UPDATE POINTER
1620 005310 005300          DEC    R0             ;CHECK FOR ALL DONE
1621 005312 001365          BNE    2$            ;BR IF NOT DONE
1622 005314 104400 4$:    SCOPE          ;SCOPE THIS TEST
```

```
1623  
1624          ;CHARACTER MEMORY TEST  
1625          ;LOAD EACH LOCATION IN CHARACTER MEMORY WITH 52525  
1626          ;VERIFY THAT EACH LOCATION IN CHARACTER MEMORY WAS  
1627          ;LOADED WITH 52525
```

```
1628  
1629          ; TEST 21  
1630          ;*****
```

```
1631 005316 012737 000021 001226  TST21: MOV    #21,TSTNO  
1632 005324 012737 005446 001216    MOV    #TST22,NEXT  
1633 005332 012737 005416 001220    MOV    #2$,LOCK  
1634 005340 012705 052525          MOV    #52525,R5      ;LOAD THE DATA INTO R5  
1635 005344 012700 000020          MOV    #16.,R0       ;SET COUNT FOR 16  
1636 005350 005002          CLR    R2             ;CLEAR REGISTER POINTER  
1637 005352 012703 000010          MOV    #10,R3        ;GET REGISTER IN R3  
1638 005356 112777 000010 174004    MOVB   #10,@DQREG      ;SELECT THE REGISTER  
1639 005364 110277 173772 1$:    MOVB   R2,@DQRC5H      ;SELECT THE CHARACTER DET REG  
1640 005370 010577 173776          MOV    R5,@DQSEC     ;LOAD THE DATA  
1641 005374 005202          INC    R2             ;UPDATE THE POINTER  
1642 005376 005300          DEC    R0             ;ALL DONE YET?  
1643 005400 001371          BNE    1$            ;BR IF NOT DONE  
1644 005402 012700 000020          MOV    #16.,R0       ;SET FOR 16 REGISTERS TO BE TESTED  
1645 005406 005002          CLR    R2             ;ZERO REG POINTER  
1646 005410 112777 000010 173752    MOVB   #10,@DQREG      ;SELECT THE REGISTER  
1647 005416 110277 173740 2$:    MOVB   R2,@DQRC5H      ;SELECT THE CHARACTER DET REGISTER  
1648 005422 017704 173744          MOV    @DQSEC,R4     ;READ THE DATA  
1649 005426 020504          CMP    R5,R4          ;WAS IT LOADED CORRECTLY?  
1650 005430 001401          BEQ    3$             ;BR IF DATA OK  
1651 005432 104006          HLT    6              ;INCORRECT DATA REPORT ERROR  
1652 005434 104401 3$:    SCOP1          ;CHECK FOR FREEZE ON DATA  
1653 005436 005202          INC    R2             ;UPDATE POINTER  
1654 005440 005300          DEC    R0             ;CHECK FOR ALL DONE  
1655 005442 001365          BNE    2$            ;BR IF NOT DONE  
1656 005444 104400 4$:    SCOPE          ;SCOPE THIS TEST
```

```
1657  
1658          ;SEQUENCE MEMORY TEST  
1659          ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 000000  
1660          ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS  
1661          ;LOADED WITH 000000
```

```
1662  
1663          ; TEST 22  
1664          ;*****
```

```
1665 005446 012737 000022 001226  TST22: MOV    #22,TSTNO
```

```

1666 005454 012737 005576 001216      MOV      #TST23,NEXT
1667 005462 012737 005546 001220      MOV      #2$,LOCK
1668 005470 012705 000000      MOV      #000000,R5      ;LOAD THE DATA INTO R5
1669 005474 012700 000020      MOV      #16.,R0        ;SET COUNT FOR 16
1670 005500 005002      CLR      R2             ;CLEAR REGISTER POINTER
1671 005502 012703 000014      MOV      #14,R3         ;GET REGISTER IN R3
1672 005506 112777 000014 173654      MOV      #14,@DQREG     ;SELECT THE REGISTER
1673 005514 110277 173642 1$:      MOV      R2,@DQRCSH    ;SELECT THE CHARACTER DET REG
1674 005520 010577 173646      MOV      R5,@DQSEC     ;LOAD THE DATA
1675 005524 005202      INC      R2             ;UPDATE THE POINTER
1676 005526 005300      DEC      R0             ;ALL DONE YET?
1677 005530 001371      BNE      1$            ;BR IF NOT DONE
1678 005532 012700 000020      MOV      #16.,R0        ;SET FOR 16 REGISTERS TO BE TESTED
1679 005536 005002      CLR      R2             ;ZERO REG POINTER
1680 005540 112777 000014 173622      MOV      #14,@DQREG     ;SELECT THE REGISTER
1681 005546 110277 173610 2$:      MOV      R2,@DQRCSH    ;SELECT THE CHARACTER DET REGISTER
1682 005552 017704 173614      MOV      @DQSEC,R4     ;READ THE DATA
1683 005556 020504      CMP      R5,R4         ;WAS IT LOADED CORRECTLY?
1684 005560 001401      BEQ      3$            ;BR IF DATA OK
1685 005562 104007      HLT      7              ;INCORRECT DATA REPORT ERROR
1686 005564 104401 3$:      SCOPE1                ;CHECK FOR FREEZE ON DATA
1687 005566 005202      INC      R2             ;UPDATE POINTER
1688 005570 005300      DEC      R0             ;CHECK FOR ALL DONE
1689 005572 001365      BNE      2$            ;BR IF NOT DONE
1690 005574 104400 4$:      SCOPE                  ;SCOPE THIS TEST
1691
1692      ;SEQUENCE MEMORY TEST
1693      ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 177777
1694      ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
1695      ;LOADED WITH 177777
1696
1697      ; TEST 23
1698      ;*****
1699 005576 012737 000023 001226 1ST23: MOV      #23,TSTNC
1700 005604 012737 005726 001216      MOV      #TST24,NEXT
1701 005612 012737 005676 001220      MOV      #2$,LOCK
1702 005620 012705 177777      MOV      #177777,R5     ;LOAD THE DATA INTO R5
1703 005624 012700 000020      MOV      #16.,R0        ;SET COUNT FOR 16
1704 005630 005002      CLR      R2             ;CLEAR REGISTER POINTER
1705 005632 012703 000014      MOV      #14,R3         ;GET REGISTER IN R3
1706 005636 112777 000014 173524      MOV      #14,@DQREG     ;SELECT THE REGISTER
1707 005644 110277 173512 1$:      MOV      R2,@DQRCSH    ;SELECT THE CHARACTER DET REG
1708 005650 010577 173516      MOV      R5,@DQSEC     ;LOAD THE DATA
1709 005654 005202      INC      R2             ;UPDATE THE POINTER
1710 005656 005300      DEC      R0             ;ALL DONE YET?
1711 005660 001371      BNE      1$            ;BR IF NOT DONE
1712 005662 012700 000020      MOV      #16.,R0        ;SET FOR 16 REGISTERS TO BE TESTED
1713 005666 005002      CLR      R2             ;ZERO REG POINTER
1714 005670 112777 000014 173472      MOV      #14,@DQREG     ;SELECT THE REGISTER
1715 005676 110277 173460 2$:      MOV      R2,@DQRCSH    ;SELECT THE CHARACTER DET REGISTER
1716 005702 017704 173464      MOV      @DQSEC,R4     ;READ THE DATA
1717 005706 020504      CMP      R5,R4         ;WAS IT LOADED CORRECTLY?
1718 005710 001401      BEQ      3$            ;BR IF DATA OK
1719 005712 104007      HLT      7              ;INCORRECT DATA REPORT ERROR
1720 005714 104401 3$:      SCOPE1                ;CHECK FOR FREEZE ON DATA
1721 005716 005202      INC      R2             ;UPDATE POINTER

```

```
1722 005720 005300          DEC    R0          ;CHECK FOR ALL DONE
1723 005722 001365          BNE    2$          ;BR IF NOT DONE
1724 005724 104400          4$:    SCOPE          ;SCOPE THIS TEST
1725
1726          ;SEQUENCE MEMORY TEST
1727          ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 125252
1728          ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
1729          ;LOADED WITH 125252
1730
1731          : TEST 24
1732          :*****
1733 005726 012737 000024 001226 TST24: MOV    #24,TSTNO
1734 005734 012737 006056 001216      MOV    #TST25,NEXT
1735 005742 012737 006026 001220      MOV    #2$,LOCK
1736 005750 012705 125252          MOV    #125252,R5      ;LOAD THE DATA INTO R5
1737 005754 012700 000020          MOV    #16.,R0        ;SET COUNT FOR 16
1738 005760 005002          CLR    R2            ;CLEAR REGISTER POINTER
1739 005762 012703 000014          MOV    #14,R3        ;GET REGISTER IN R3
1740 005766 112777 000014 173374      MOVB  #14,@DQREG     ;SELECT THE REGISTER
1741 005774 110277 173362          1$:    MOVB  R2,@DQRC5H ;SELECT THE CHARACTER DET REG
1742 006000 010577 173366          MOV    R5,@DQSEC     ;LOAD THE DATA
1743 006004 005202          INC    R2            ;UPDATE THE POINTER
1744 006006 005300          DEC    R0            ;ALL DONE YET?
1745 006010 001371          BNE    1$            ;BR IF NOT DONE
1746 006012 012700 000020          MOV    #16.,R0      ;SET FOR 16 REGISTERS TO BE TESTED
1747 006016 005002          CLR    R2            ;ZERO REG POINTER
1748 006020 112777 000014 173342      MOVB  #14,@DQREG     ;SELECT THE REGISTER
1749 006026 110277 173330          2$:    MOVB  R2,@DQRC5H ;SELECT THE CHARACTER DET REGISTER
1750 006032 017704 173334          MOV    @DQSEC,R4     ;READ THE DATA
1751 006036 020504          CMP    R5,R4        ;WAS IT LOADED CORRECTLY?
1752 006040 001401          BEQ    3$            ;BR IF DATA OK
1753 006042 104007          HLT    7            ;INCORRECT DATA REPORT ERROR
1754 006044 104401          3$:    SCOP1          ;CHECK FOR FREEZE ON DATA
1755 006046 005202          INC    R2            ;UPDATE POINTER
1756 006050 005300          DEC    R0            ;CHECK FOR ALL DONE
1757 006052 001365          BNE    2$            ;BR IF NOT DONE
1758 006054 104400          4$:    SCOPE          ;SCOPE THIS TEST
1759
1760          ;SEQUENCE MEMORY TEST
1761          ;LOAD EACH LOCATION IN SEQUENCE MEMORY WITH 52525
1762          ;VERIFY THAT EACH LOCATION IN SEQUENCE MEMORY WAS
1763          ;LOADED WITH 52525
1764
1765          : TEST 25
1766          :*****
1767 006056 012737 000025 001226 TST25: MOV    #25,TSTNO
1768 006064 012737 006206 001216      MOV    #TST26,NEXT
1769 006072 012737 006156 001220      MOV    #2$,LOCK
1770 006100 012705 052525          MOV    #52525,R5     ;LOAD THE DATA INTO R5
1771 006104 012700 000020          MOV    #16.,R0      ;SET COUNT FOR 16
1772 006110 005002          CLR    R2            ;CLEAR REGISTER POINTER
1773 006112 012703 000014          MOV    #14,R3        ;GET REGISTER IN R3
1774 006116 112777 000014 173244      MOVB  #14,@DQREG     ;SELECT THE REGISTER
1775 006124 110277 173232          1$:    MOVB  R2,@DQRC5H ;SELECT THE CHARACTER DET REG
1776 006130 010577 173236          MOV    R5,@DQSEC     ;LOAD THE DATA
1777 006134 005202          INC    R2            ;UPDATE THE POINTER
```



```

1834 006344 017704 173010          MOV    @DQPCSR,R4          ;(R4)=ACTUAL 07400 IN
1835                                     ;RECEIVER CONTROL REGISTER
1836 006350 001401          BEQ    1$                ;BR IF ALL OK.
1837 006352 104000          HLT                                     ;PRIMARY REGISTER MASTER CLEAR ERROR
1838 006354 104400          1$:   SCOPE
1839
1840                                     ;IF DATASET CONTROL OPTION IS INSTALLED
1841                                     ;TEST 30 WILL BE EXECUTED
1842
1843 006356 032737 010000 001510  CHKBA1: BIT    #BABIT,DQSTAT
1844 006364 001436          BEQ    TST31
1845
1846
1847                                     ;TRANSMITTER CONTROL REGISTER MASTER CLEAR TEST
1848                                     ;SET ALL READ/WRITE BITS IN TRANSMITTER CONTROL REGISTER
1849                                     ;ISSUE MASTER CLEAR
1850                                     ;VERIFY THAT TRANSMITTER CONTROL WAS CLEARED
1851
1852                                     : TEST 30
1853                                     :*****
1854 006366 012737 000030 001226  TST30: MOV    #30,TSTNO
1855 006374 012737 006460 001216          MOV    #CHKBA2,NEXT
1856 006402 012737 000340 177776          MOV    #340,PS          ;LOCK OUT INTERRUPTS
1857 006410 013703 001364          MOV    DQTCR,R3        ;BUS ADDRESS OF
1858                                     ;TRANSMITTER CONTROL REGISTER
1859 006414 012777 101772 172742          MOV    #101772,@DQTCR ;LOAD TRANSMITTER CONTROL REGISTER
1860 006422 112777 000012 172740          MOVB  #12,@DQREG       ;SELECT MISCELLANEOUS REGISTER
1861 006430 052777 000040 172734          BIS   #BIT5,@DQSEC    ;ISSUE MASTER CLEAR
1862 006436 005005          CLR    R5              ;(R5)=EXPECTED 101772 IN
1863                                     ;TRANSMITTER CONTROL REGISTER, 0
1864 006440 017704 172720          MOV    @DQTCR,R4      ;(R4)=ACTUAL 101772 IN
1865                                     ;TRANSMITTER CONTROL REGISTER
1866 006444 042704 100000          BIC   #100000,R4      ;100000 OFF UNWANTED BITS.
1867 006450 020504          CMP   R5,R4           ;EXPECTED=ACTUAL?
1868 006452 001401          BEQ    1$              ;BR IF ALL OK.
1869 006454 104000          HLT                                     ;PRIMARY REGISTER MASTER CLEAR ERROR
1870 006456 104400          1$:   SCOPE
1871 006460 000435          CHKBA2: BR    TST32
1872
1873                                     ;IF DATASET CONTROL OPTION IS NOT INSTALLED
1874                                     ;TEST 31 WILL BE EXECUTED
1875
1876
1877                                     ;TRANSMITTER CONTROL REGISTER MASTER CLEAR TEST
1878                                     ;SET ALL READ/WRITE BITS IN TRANSMITTER CONTROL REGISTER
1879                                     ;ISSUE MASTER CLEAR
1880                                     ;VERIFY THAT TRANSMITTER CONTROL WAS CLEARED
1881
1882                                     : TEST 31
1883                                     :*****
1884 006462 012737 000031 001226  TST31: MOV    #31,ISTNO
1885 006470 012737 006554 001216          MOV    #TST32,NEXT
1886 006476 012737 000340 177776          MOV    #340,PS          ;LOCK OUT INTERRUPTS
1887 006504 013703 001364          MOV    DQTCR,R3        ;BUS ADDRESS OF
1888                                     ;TRANSMITTER CONTROL REGISTER
1889 006510 012777 101772 172646          MOV    #101772,@DQTCR ;LOAD TRANSMITTER CONTROL REGISTER
  
```



```
1946
1947 006730 017704 172436      MOV      @DQSEC,R4      ;SYNC REGISTER, 0
1948                                ;(R4)=ACTUAL 177777 IN
1949 006734 001401      BEQ      1$            ;SYNC REGISTER
1950 006736 104000      HLT                                ;BR IF ALL OK.
1951 006740 104400      1$:      SCOPE        ;SECONDARY REGISTER MASTER CLEAR ERROR
1952
1953                                ;MISCELLANEOUS REGISTER MASTER CLEAR TEST
1954                                ;SET ALL READ/WRITE BITS IN MISCELLANEOUS REGISTER
1955                                ;ISSUE MASTER CLEAR
1956                                ;VERIFY THAT MISCELLANEOUS WAS CLEARED
1957
1958                                ; TEST 34
1959                                ;*****
1960 006742 012737 000034 001226  TST34:  MOV      #34,TSTNO
1961 006750 012737 007036 001216      MOV      #CHKAB1,NEXT
1962 006756 012737 000340 177776      MOV      #340,PS
1963 006764 012703 000012      MOV      #12,R3      ;LOCK OUT INTERRUPTS
1964                                ;GET ADDRESS OF SECONDARY
1965 006770 110377 172374      MOVVB   R3,@DQREG    ;REGISTER TO BE TESTED
1966                                ;SELECT CERE
1967 006774 012777 007717 172370      MOV      #7717,@DQSEC ;SECONDARY REGISTER
1968 007002 112777 000012 172360      MOVVB   #12,@DQREG   ;LOAD MISCELLANEOUS REGISTER
1969 007010 052777 000040 172354      BIS     #BIT5,@DQSEC ;SELECT MISCELLANEOUS REGISTER
1970 007016 110377 172346      MOVVB   R3,@DQREG   ;ISSUE MASTER CLEAR
1971 007022 005005      CLR     R5          ;SELECT MISCELLANEOUS SECONDARY REGISTER
1972                                ;(R5)=EXPECTED 7717 IN
1973 007024 017704 172342      MOV      @DQSEC,R4  ;MISCELLANEOUS REGISTER, 0
1974                                ;(R4)=ACTUAL 7717 IN
1975 007030 001401      BEQ      1$            ;MISCELLANEOUS REGISTER
1976 007032 104000      HLT                                ;BR IF ALL OK.
1977 007034 104400      1$:      SCOPE        ;SECONDARY REGISTER MASTER CLEAR ERROR
1978
1979                                ;IF BCC OPTION IS INSTALLED
1980                                ;TEST 35 WILL BE EXECUTED
1981
1982 007036 032737 002000 001510  (CHKAB1: BIT     #ABBIT,DQSTAT
1983 007044 001436      BEQ     .EOP
1984
1985                                ;BCC POLYNOMIAL REGISTER MASTER CLEAR TEST
1986                                ;SET ALL READ/WRITE BITS IN BCC POLYNOMIAL REGISTER
1987                                ;ISSUE MASTER CLEAR
1988                                ;VERIFY THAT BCC POLYNOMIAL WAS CLEARED
1989
1990                                ; TEST 35
1991                                ;*****
1992 007046 012737 000035 001226  TST35:  MOV      #35,TSTNO
1993 007054 012737 007142 001216      MOV      #.EOP,NEXT
1994 007062 012737 000340 177776      MOV      #340,PS
1995 007070 012703 000017      MOV      #17,R3      ;LOCK OUT INTERRUPTS
1996                                ;GET ADDRESS OF SECONDARY
1997 007074 110377 172270      MOVVB   R3,@DQREG    ;REGISTER TO BE TESTED
1998                                ;SELECT CERE
1999 007100 012777 177777 172264      MOV      #177777,@DQSEC ;SECONDARY REGISTER
2000 007106 112777 000012 172254      MOVVB   #?,@DQREG   ;LOAD BCC POLYNOMIAL REGISTER
2001 007114 052777 000040 172250      BIS     #BIT5,@DQSEC ;SELECT MISCELLANEOUS REGISTER
                                ;ISSUE MASTER CLEAR
```


2002	007122	110377	172242	MOVB	R3,@DQREG	:SELECT BCC POLYNOMIAL SECONDARY REGISTER
2003	007126	005005		CLR	R5	:(R5)=EXPECTED 177777 IN
2004						:BCC POLYNOMIAL REGISTER, 0
2005	007130	017704	172236	MOV	@DQSEC,R4	:(R4)=ACTUAL 177777 IN
2006						:BCC POLYNOMIAL REGISTER
2007	007134	001401		BEQ	1\$:BR IF ALL OK.
2008	007136	104060		HLT		:SECONDARY REGISTER MASTER CLEAR ERROR
2009	007140	104400	1\$.	SCOPE		

```

2010
2011
2012      :END OF PASS
2013      :TYPE NAME OF TEST
2014      :UPDATE PASS COUNT
2015      :CHECK FOR EXIT TO ACT-11
2016      :RESTART TEST
2017 007142 005037 001234      .EOP: CLR      LSTERR      :CLEAR LAST ERROR PC
2018 007146 005037 001312      CLR      ERRFLG      :CLEAR ERROR FLAG
2019 007152 005237 001230      INC      PASCNT      :UPDATE PASS COUNT
2020 007156 104402
2021 007160 011372      MEPASS
2022 007162 104402      TYPE
2023 007164 011553      MCSRX
2024 007166 104411      CNVRT
2025 007170 007300      XCSR
2026 007172 104402      TYPE
2027 007174 011561      MVECX
2028 007176 104411      CNVRT
2029 007200 007306      XVEC
2030 007202 104402      TYPE
2031 007204 011567      MPASSX
2032 007206 104411      CNVRT
2033 007210 007314      XPASS
2034 007212 104402      TYPE
2035 007214 011600      MERRX
2036 007216 104411      CNVRT
2037 007220 007322      XERR
2038 007222 013777 001230 171752      MOV      PASCNT,@LIGHTS :DISPLAY PASS COUNT
2039 007230 005337 001276      DEC      SAVNUM
2040 007234 001013      BNE      RESTRT
2041 007236 013737 001504 001276      MOV      DQNUM,SAVNUM
2042 007244 013701 000042      MOV      @#42,R1
2043 007250 001405      BEQ      RESTRT      :CHECK FOR ACT-11 OR DDP
2044 007252 000005      RESET      :IF NOT, CONTINUE TESTING
2045 007254
2046 007254 004711      LOGICAL: JSR      PC,(R1)
2047 007256 000240      NOP
2048 007260 000240      NOP
2049 007262 000240      NOP
2050 007264 104414      RESTRT: CKSWR
2051 007266 012737 002254 001214      MOV      #TST1,RETURN
2052 007274 000137 002254      JMP      TST1
2053 007300 000001      XCSR: 1
2054 007302 006 002      .BYTE 6.2
2055 007304 001360      DQRCSR
2056 007306 000001      XVEC: 1
2057 007310 003 002      .BYTE 3.2
2058 007312 001350      DQRVEC
2059 007314 000001      XPASS: 1
2060 007316 006 002      .BYTE 6.2
2061 007320 001230      PASCNT
2062 007322 000001      XERR: 1
2063 007324 006 002      .BYTE 6.2
2064 007326 001232      ERRCNT
2065

```

```

2066                                     ;SCOPE LOOP AND INTERATION HANDLER
2067
2068 007330 104414 .SCOPE: CKSWR
2069 007332 032777 040000 171640 BIT #BIT14,@SWR
2070 007340 001407 TTST: BEQ 1$
2071 007342 000432 BR 3$
2072 007344 105777 171634 TSTB @TKCSR
2073 007350 100027 BPL 3$
2074 007352 017700 171630 MOV @TKDBR,R0
2075 007356 000412 BR 2$
2076 007360 032777 004000 171612 1$: BIT #SW11,@SWR
2077 007366 001006 BNE 2$
2078 007370 005237 001224 INC LPCNT
2079 007374 023737 001224 001222 CMP LPCNT,I COUNT
2080 007402 001012 BNE 3$
2081 007404 105037 001312 2$: CLRB ERRFLG
2082 007410 005037 001224 CLR LPCNT
2083 007414 012737 002000 001222 MOV #2000,I COUNT
2084 007422 013737 001216 001214 MOV NEXT,RETURN
2085 007430 013716 001214 3$: MOV RETURN,(SP)
2086 007434 000002 RTI
2087 007436 001407 BRW: 1407
2088 007440 000432 BRX: 432
2089
2090                                     ;CHECK FOR FREEZE ON CURRENT DATA
2091
2092 007442 104414 .SCOPE1: CKSWR
2093 007444 032777 001000 171526 BIT #SW09,@SWR
2094 007452 001402 BEQ 1$
2095 007454 013716 001220 MOV LOCK,(SP)
2096 007460 000002 1$: RTI
2097
2098                                     ;TELETYPE OUTPUT ROUTINE
2099
2100 .TYPE: MOV R5,-(SP)
2101 007464 017605 MOV @2(SP),R5
2102 007470 062766 000002 000002 ADD #2,2(SP)
2103 007476 005737 011152 1$: TST @#RDSW
2104 007502 001004 BNE 300$
2105 007504 032777 010000 171466 BIT #SW12,@SWR
2106 007512 001024 BNE 3$
2107 007514 105715 300$: TSTB (R5)
2108 007516 100014 BPL 2$
2109 007520 105777 171464 TSTB @TPCSR
2110 007524 100375 BPL .-4
2111 007526 012777 000015 171456 MOV #15,@TPDBR
2112 007534 105777 171450 TSTB @TPCSR
2113 007540 100375 BPL .-4
2114 007542 012777 000012 171442 MOV #12,@TPDBR
2115 007550 105777 171434 2$: TSTB @TPCSR
2116 007554 100375 BPL 2$
2117 007556 112577 171430 MOVB (R5)+,@TPDBR
2118 007562 001345 BNE 1$
2119 007564 012605 3$: MOV (SP)+,R5
2120 007566 000002 RTI
2121

```

```

2122                                     ;ASCII STRING INPUT ROUTINE
2123
2124 007570 010346      .INSTR: MOV     R3,-(SP)
2125 007572 010446      MOV     R4,-(SP)
2126 007574 017637 000004 007612      MOV     @4(SP),.MSG
2127 007602 062766 000002 000004      ADD     #2,4(SP)
2128 007610 104402      .INST1: TYPE
2129 007612 000000      .MSG: 0
2130 007614 012704 011744      MOV     #INBUF,R4
2131 007620 012703 000007      MOV     #7,R3
2132 007624 105777 171354      1$:   TSTB   @TKCSR
2133 007630 100375      BPL    1$
2134 007632 117714 171350      MOVB   @TKDBR,(R4)
2135 007636 142714 000200      BICB   #200,(R4)
2136 007642 121427 000025      CMPB   (R4),#25
2137 007646 001003      BNE    200$
2138 007650 104402 011332      TYPE ,MCRLF
2139 007654 000755      BR     .INST1
2140 007656 122427 000015      200$: CMPB   (R4)+,#15
2141 007662 001423      BEQ    INSTR2
2142 007664 117777 171316 171320      MOVB   @TKDBR,@TPDBR
2143 007672 105777 171312      2$:   TSTB   @TPCSR
2144 007676 100375      BPL    2$
2145 007700 005303      DEC    R3
2146 007702 001350      BNE    1$
2147 007704 000402      BR     .INSTG
2148 007706 010346      .INSTE: MOV    R3,-(SP)
2149 007710 010446      MOV    R4,-(SP)
2150 007712 104402      .!INSTG: TYPE
2151 007714 011326      MQM
2152 007716 005737 011152      TST    @#RDSW
2153 007722 001402      BEQ    400$
2154 007724 104402 011332      TYPE ,MCRLF
2155 007730 000727      400$: BR     .INST1
2156 007732 012604      INSTR2: MOV   (SP)+,R4
2157 007734 012603      MOV   (SP)+,R3
2158 007736 000002      RTI
2159
2160                                     ;CONVERT ASCII STRING TO OCTAL
2161
2162 007740 010546      .PARAM: MOV    R5,-(SP)
2163 007742 010446      MOV    R4,-(SP)
2164 007744 016605 000004      MOV    4(SP),R5
2165 007750 012537 010144      MOV    (R5)+,LOLIM
2166 007754 012537 010146      MOV    (R5)+,HILIM
2167 007760 012537 010150      MOV    (R5)+,DEVADR
2168 007764 112537 010152      MOVB   (R5)+,LOBITS
2169 007770 112537 010153      MOVB   (R5)+,ADRCNT
2170 007774 010566 000004      MOV    R5,4(SP)
2171 010000 005005      PARAM1: CLR   R5
2172 010002 012704 011744      MOV    #INBUF,R4
2173 010006 122714 000015      CMPB   #15,(R4)
2174 010012 001420      BEQ    PARERR
2175 010014 121427 000060      1$:   CMPB   (R4),#60
2176 010020 002415      BLT    PARERR
2177 010022 121427 000067      CMPB   (R4),#67
  
```

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

```
2178 010026 003012          BGT      PARERR
2179 010030 142714 000060    BICB    #60,(R4)
2180 010034 152405          BISB    (R4)+,R5
2181 010036 122714 000015    CMPB    #15,(R4)
2182 010042 001414          BEQ     LIMITS
2183 010044 006305          ASL     R5
2184 010046 006305          ASL     R5
2185 010050 006305          ASL     R5
2186 010052 000760          BR      1$
2187 010054 122714 000015    PARERR: CMPB    #15,(R4)          ;IS FIRST CHARACTER A <CR>
2188 010060 001003          BNE     120$
2189 010062 005737 011152    TST     @#RDSW          ;IS CKSWR ROUTINE BEING USED
2190 010066 001023          BNE     PARTI
2191 010070 104404          120$:  INSTER
2192 010072 000742          BR      PARAM1
2193
2194          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2195
2196 010074 020537 010146    LIMITS: CMP     R5,HILIM
2197 010100 101365          BHI     PARERR
2198 010102 020537 010144    CMP     R5,LOLIM
2199 010106 103762          BLO     PARERR
2200 010110 133705 010152    BITB    LOBITS,R5
2201 010114 001357          BNE     PARERR
2202
2203          ;STORE NUMBER AT SPECIFIED ADDRESS
2204
2205 010116 013704 010150          MOV     DEVADR,R4
2206 010122 010524          1$:  MOV     R5,(R4)+
2207 010124 062705 000002    ADD     #2,R5
2208 010130 105337 010153    DECB    ADCRNT
2209 010134 001372          BNE     1$
2210 010136 012604          PARTI. MOV     (SP)+,R4
2211 010140 012605          MOV     (SP)+,R5
2212 010142 000002          RTI
2213 010144 000000          LOLIM: 0
2214 010146 000000          HILIM: 0
2215 010150 000000          DEVADR: 0
2216 010152 000000          LOBITS: 0
2217          ADCRNT-LOBITS+1
2218
2219          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2220
2221 010154 016637 000004 001274 .SAV05: MOV     4(SP),SAVPC
2222
2223          ;SAVE R0-R5
2224
2225 010162 010537 001270          SV05: MOV     R5,SAVR5
2226 010166 010437 001266          MOV     R4,SAVR4
2227 010172 010337 001264          MOV     R3,SAVR3
2228 010176 010237 001262          MOV     R2,SAVR2
2229 010202 010137 001260          MOV     R1,SAVR1
2230 010206 010037 001256          MOV     R0,SAVR0
2231 010212 000002          RTI
2232
2233          ;RESTORE R0-R5
```

```

2234
2235 010214 013700 001256 .RES05: MOV SAVR0,R0
2236 010220 013701 001260 MOV SAVR1,R1
2237 010224 013702 001262 MOV SAVR2,R2
2238 010230 013703 001264 MOV SAVR3,R3
2239 010234 013704 001266 MOV SAVR4,R4
2240 010240 013705 001270 MOV SAVR5,R5
2241 010244 000002 RTI
2242
2243 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2244
2245 010246 104402 .CONVR: TYPE
2246 010250 011332 MCRLF
2247 010252 010046 .CNVRT: MOV R0,-(SP)
2248 010254 010146 MOV R1,-(SP)
2249 010256 010346 MOV R3,-(SP)
2250 010260 010446 MOV R4,-(SP)
2251 010262 010546 MOV R5,-(SP)
2252 010264 017601 000012 MOV @12(SP),R1
2253 010270 013737 012006 001250 MOV TEMP,TEMP3
2254 010276 062766 000002 000012 ADD #2,12(SP)
2255 010304 012137 010466 MOV (R1)+,WRDCNT
2256 010310 112137 010470 1$: MOV (R1)+,CHRCNT
2257 010314 112137 010471 MOV (R1)+,SPACNT
2258 010320 013137 010472 MOV @ (R1)+,BINWRD
2259 010324 013704 010472 2$: MOV BINWRD,R4
2260 010330 113705 010470 MOV CHRCNT,R5
2261 010334 012700 012006 MOV #TEMP,R0
2262 010340 010403 3$: MOV R4,R3
2263 010342 042703 177770 BIC #177770,R3
2264 010346 062703 000060 ADD #060,R3
2265 010352 110320 MOV R3,(R0)+
2266 010354 000241 CLC
2267 010356 006004 ROR R4
2268 010360 000241 CLC
2269 010362 006004 ROR R4
2270 010364 000241 CLC
2271 010366 006004 ROR R4
2272 010370 005305 DEC R5
2273 010372 001362 BNE 3$
2274 010374 012703 012050 MOV #MDATA,R3
2275 010400 114023 4$: MOV -(R0),(R3)+
2276 010402 105337 010470 DECB CHRCNT
2277 010406 001374 BNE 4$
2278 010410 105737 010471 TSTB SPACNT
2279 010414 001405 BEQ 6$
2280 010416 112723 000040 5$: MOV #040,(R3)+
2281 010422 105337 010471 DECB SPACNT
2282 010426 001373 BNE 5$
2283 010430 105013 6$: CLRB (R3)
2284 010432 104402 TYPE
2285 010434 012050 MDATA
2286 010436 005337 010466 DEC WRDCNT
2287 010442 001322 BNE 1$
2288 010444 013737 001250 012006 MOV TEMP3,TEMP
2289 010452 012605 MOV (SP)+,R5
  
```

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

2290	010454	012604				MOV	(SP)+,R4		
2291	010456	012603				MOV	(SP)+,R3		
2292	010460	012601				MOV	(SP)+,R1		
2293	010462	012600				MOV	(SP)+,R0		
2294	010464	000002				RTI			
2295	010466	000000				WRDCNT:	0		
2296	010470	000000				CHRCNT:	0		
2297		010471				SPACNT=	CHRCNT+1		
2298	010472	000000				BINWRD:	0		
2299								:TRAP DISPATCH SERVICE	
2300								:ARGUMENT OF TRAP IS EXTRACTED	
2301								:AND USED AS OFFSET TO OBTAIN POINTER	
2302								:TO SELECTED SUBROUTINE	
2303									
2304	010474	011646				.TRPSR:	MOV	(SP),-(SP)	:GET PC OF RETURN
2305	010476	162716	000002				SUB	#2,(SP)	:PC OF TRAP
2306	010502	017616	000000				MOV	@(SP),(SP)	:GET TRP
2307	010506	006316				TRPOK:	ASL	(SP)	:MULTIPLY TRAP ARG BY 2
2308	010510	042716	177001				BIC	#177001,(SP)	:CLEAR UNWANTED BITS
2309	010514	062716	001314				ADD	#.TRPTAB,(SP)	:POINTER TO SUBROUTINE ADDRESS
2310	010520	017616	000000				MOV	@(SP),(SP)	:SUBROUTINE ADDRESS
2311	010524	000136					JMP	@(SP)+	:GC TO SUBROUTINE
2312									
2313									
2314								:ERROR HANDLER	
2315	010526	104414				.HLT:	CKSWR		
2316	010530	032777	010000	170442			BIT	#SW12,@SWR	
2317	010536	001406					BEQ	XBX	
2318	010540	105777	170444				TSTB	@TPCSR	
2319	010544	100003					BPL	XBX	
2320	010546	112777	000207	170436			MOVB	#207,@TPDBR	
2321	010554	032777	020000	170416		XBX:	BIT	#SW13,@SWR	
2322	010562	001074					BNE	HALTS	
2323	010564	021637	001234				CMP	(SP),LSTERR	
2324	010570	001404					BEQ	1\$	
2325	010572	011637	001234				MOV	(SP),LSTERR	
2326	010576	105037	001312				CLRB	ERRFLG	
2327	010602	104406				1\$:	SAV05		
2328	010604	011605					MOV	(SP),R5	
2329	010606	162705	000002				SUB	#2,R5	
2330	010612	011504					MOV	(R5),R4	
2331	010614	006304					ASL	R4	
2332	010616	061504					ADD	(R5),R4	
2333	010620	006304					ASL	R4	
2334	010622	042704	177001				BIC	#177001,R4	
2335	010626	062704	012760				ADD	#.ERRTAB,R4	
2336	010632	012437	010724				MOV	(R4)+,ERRMSG	
2337	010636	012437	010736				MOV	(R4)+,DATAHD	
2338	010642	011437	010750				MOV	(R4),DATABP	
2339	010646	105737	001312				TSTB	ERRFLG	
2340	010652	001403					BEQ	TYPMSG	
2341	010654	005737	010750				TST	DATABP	
2342	010660	001027					BNE	TYPDA'	
2343	010662	104402				TYPMSG:	TYPE		
2344	010664	011611					MTSTN		
2345	010666	104411					CNVRT		

```

2346 010670 011050 XTSTN
2347 010672 104402 TYPE
2348 010674 011677 MERRPC
2349 010676 104411 CNVRT
2350 010700 011042 ERTABO
2351 010702 104402 TYPE
2352 010704 011332 MCRLF
2353 010706 112737 177777 001312 MOVB #-1,ERRFLG
2354 010714 005737 010724 TST ERRMSG
2355 010720 001402 BEQ WRKO.FM
2356 010722 104402 TYPE
2357 010724 000000 ERRMSG: 0
2358 010726 WRKO.FM:
2359 010726 005737 010736 TST DATAHD
2360 010732 001402 BEQ TYPDAT
2361 010734 104402 TYPE
2362 010736 000000 DATAHD: 0
2363 010740 005737 010750 TYPDAT: TST DATABP
2364 010744 001402 BEQ RESREG
2365 010746 104410 CONVPT
2366 010750 000000 DATABP: 0
2367 010752 104407 RESREG: RES05
2368 010754 005777 170220 HALTS: TST @SWR
2369 010760 100005 BPL EXITER
2370 010762 010046 PUSHRO
2371 010764 016600 000002 MOV 2(SP),R0
2372 010770 000000 HALT
2373 010772 012600 POPRO
2374 010774 104414 EXITER: CKSWR
2375 010776 005237 001232 INC ERRCNT
2376 011002 032777 000400 170170 BIT #SW08,@SWR
2377 011010 001007 BNE 1$
2378 011012 032777 002000 170160 BIT #SW10,@SWR
2379 011020 001407 BEQ 2$
2380 011022 013737 001216 001214 MOV NEXT,RETURN
2381 011030 012706 001200 1$: MOV #STACK,SP
2382 011034 000177 170154 JMP @RETURN
2383 011040 000002 2$: RTI
2384 011042 000001 ERTABO: 1
2385 011044 006 002 .BYTE 6,2
2386 011046 001274 SAVPC
2387 011050 000001 XTSTN: 1
2388 011052 003 002 .BYTE 3,2
2389 011054 001226 TSTNO
2390 :ENTER HERE ON POWER FAILURE
2391
2392
2393 011056 .PFAIL:
2394 011056 012737 011070 000024 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
2395 011064 000000 HALT ;HALT ON POWER DOWN NORMAL
2396 011066 000777 BR .
2397
2398 ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2399
2400 011070 RESTAR:
2401 011070 012737 011056 000024 MOV #,PFAIL,24 ;SET UP FOR POWER FAILURE
  
```


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

2402 011076 012706 001200
2403 011102 005037 012006
2404 011106 005237 012006
2405 011112 001375
2406 011114 104402
2407 011116 011334
2408 011120 104411
2409 011122 011144
2410 011124 005037 001312
2411 011130 005037 001234
2412 011134 104412
2413 011136 104413
2414 011140 000177 170050
2415 011144 000001
2416 011146 003 002
2417 011150 001226
2418
2419
2420
2421
2422
2423 011152 000000
2424
2425
2426 011154 005737 000042
2427 011160 001042
2428 011162 022737 000176 001200
2429 011170 001036
2430 011172 105777 170006
2431 011176 100033
2432 011200 017737 170002 007612
2433 011206 042737 177600 007612
2434 011214 122737 000007 007612
2435 011222 001021
2436 011224 104402 011302
2437 011230 005137 011152
2438 011234 104402 011306
2439 011240 104411 011274
2440 011244 104403 011315
2441 011250 104405
2442 011252 000000
2443 011254 177777
2444 011256 000176
2445 011260 000 001
2446 011262 104402 011332
2447 011266 005037 011152
2448 011272 000002
2449 011274 000001
2450 011276 006 002
2451 011300 000176
2452 011302 057377 000107
2453 011306 051777 051127 020075
2454 011314 000
2455 011315 040 047040 053505
2456 011322 020075 000
2457 011326

MOV #STACK,SP
CLR TEMP
INC TEMP
BNE .-4
TYPE
MPFAIL
CNVRT
PFTAB
CLR ERRFLG
CLR LSTERR
MSTCLR
MEMCLR
JMP @RETURN
PFTAB: 1
.BYTE 3,2
TSTNO

:CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
:OF LOC.176.
:LOCATIONS USED:
RDSW: .WORD 0

.CKSWR: TST @42
BNE OUT
CMP #SWREG,SWR :SOFTWARE SWITCH REGISTER PRESENT
BNE OUT :NO, GET OUT
TSTB @TKCSR :YES, WAIT FOR
BPL OUT :READY, GET CHARACTER
MOV @TKDBR,MSG :AND STRIP OFF
BIC #177600,MSG :THE GARBAGE
CMPB #7,MSG :IS IT A <^G>
BNE OUT
TYPE,SCNTG
.CNTLU: COM @RDSW
TYPE,SMSWR
CNVRT,SWREGC
INSTR,SMNEW
PARAM
0
177777
SWREG
.BYTE 0,1
TYPE,MCRLF
OUT: CLR @RDSW
RTI
SWREGC: 1
.BYTE 6,2
SWREG
SCNTG: .ASCIZ <377>/^G/
SMSWR: .ASCIZ <377>/SWR= /
SMNEW: .ASCIZ / NEW= /
.EVEN

2514	011740	006	002		.BYTE 6,2
2515	011742	001246			TEMP2
2516				.EVEN	
2517					
2518					:BUFFERS FOR INPUT-OUTPUT
2519					
2520	011744	000000		INBUF:	0
2521		012006		.+40	
2522	012006	000000		TEMP:	0
2523		012050		.+40	
2524	012050	000000		MDATA:	0
2525		012112		.+40	

```

2526
2527 012112 000002 .MEMCLR: RTI
2528 012114 000002 .MSTCLR: RTI
2529 ;TABLE OF ERROR MESSAGES AND DATA HEADERS
2530
2531 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 CLEARFD/

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

```

2532 ;TABLE OF ERROR POINTERS
2533
2534 012760 012116 .ERRTAB:EM0
2535 012762 012716 DH1
2536 012764 013126 DT2
2537 012766 012160 EM1
2538 012770 012716 DH1
2539 012772 013126 DT2
2540 012774 012217 EM2
2541 012776 012647 DH0
2542 013000 013104 DT1
2543 013002 012264 EM3
2544 013004 012647 DH0
2545 013006 013062 DT0
2546 013010 012320 EM4
2547 013012 012647 DH0
2548 013014 013104 DT1
2549 013016 012357 EM5
2550 013020 012647 DH0
2551 013022 013104 DT1
2552 013024 012415 EM6
2553 013026 012647 DH0
2554 013030 013104 DT1
2555 013032 012451 EM7
2556 013034 012647 DH0
2557 013036 013104 DT1
2558 013040 012504 EM10
2559 013042 012716 DH1
2560 013044 013126 DT2
2561 013046 012550 EM11
2562 013050 012647 DH0
2563 013052 013104 DT1
2564 013054 012616 EM12
2565 013056 000000 0
2566 013060 000000 0
  
```

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

```

2567
2568           ;TABLE OF DATA POINTER FOR ERROR TYPEOUT
2569
2570 013062 000004      DT0: 4
2571 013064 003      007 .BYTE 3,7
2572 013066 001270      SAVR5
2573 013070 003      007 .BYTE 3,7
2574 013072 001266      SAVR4
2575 013074 006      004 .BYTE 6,4
2576 013076 001372      DQSEC
2577 013100 002      000 .BYTE 2,0
2578 013102 001264      SAVR3
2579 013104 000004      DT1: 4
2580 013106 006      004 .BYTE 6,4
2581 013110 001270      SAVR5
2582 013112 006      004 .BYTE 6,4
2583 013114 001266      SAVR4
2584 013116 006      004 .BYTE 6,4
2585 013120 001372      DQSEC
2586 013122 002      000 .BYTE 2,0
2587 013124 001262      SAVR2
2588 013126 000003      DT2: 3
2589 013130 006      004 .BYTE 6,4
2590 013132 001270      SAVR5
2591 013134 006      004 .BYTE 6,4
2592 013136 001266      SAVR4
2593 013140 006      004 .BYTE 6,4
2594 013142 001264      SAVR3
2595 000001      .END

```


.CONVR	010246	857	2245#			
.EOP	007142	1983	1993	2017#		
.ERRTA	012760	2335	2534#			
.HLT	010526	647	2315#			
.INSTE	007706	849	2148#			
.INSTG	007712	2147	2150#			
.INSTR	007570	847	2124#			
.INST1	007610	2128#	2139	2155		
.MEMCL	012112	863	2527#			
.MSG	007612	2126*	2129#	2432*	2433*	2434
.MSTCL	012114	861	2528#			
.PARAM	007740	851	2162#			
.PFAIL	011056	645	938	2393#	2401	
.RES05	010214	855	2235#			
.SAV05	010154	853	2221#			
.SCOPE	007330	841	2068#			
.SCOPI	007442	843	2092#			
.START	001512	696	936#	948		
.TRPSR	010474	649	2304#			
.TRPTA	001314	839#	2309			
.TYPE	007462	845	2100#			

. ABS. 013144 000

ERRORS DETECTED: 0

DSKZ:CZDQBD,DSKZ:CZDQBD,SEQ-DSKZ:CZDQXX,MAC,DSKZ:CZDQBD.P11

RUN-TIME: 79.7 SECONDS

RUN-TIME RATIO: 51/18 2.8

CORE USED: 20K (39 PAGES)

DOCUMENT PAGES: 59